

KOMPUTASI ANALITIKA MENGUNAKAN EXCEL & R

Wahyu Wibowo
Muhammad Alifian Nuriman



KOMPUTASI ANALITIKA MENGUNAKAN EXCEL DAN R

Wahyu Wibowo
Muhammad Alifian Nuriman



2025

Komputasi Analitika Menggunakan Excel Dan R

Penulis:

Wahyu Wibowo

Muhammad Alifian Nuriman

Kontributor:

Elina Normayanti

Kharin Octavian Ranto

Renata Amalia Larasati

251 halaman, 14,8 cm x 21 cm

ISBN

978-623-318-244-7



Diterbitkan oleh:

ITS Press, Surabaya 2025

Anggota IKAPI dan APPTI

© 2025, ITS Press, Surabaya

Barang siapa dengan sengaja dan tanpa hak melakukan perbuatan yang melanggar HAK CIPTA atas buku ini, maka akan dikenakan sanksi sesuai dengan Undang-Undang No. 28 Tahun 2014 tentang Hak Cipta.

PRAKATA

Alhamdulillah, puji syukur penulis haturkan kehadiran Allah SWT, karena atas limpahan karunia dan izin-Nya, penulis dapat menyelesaikan buku untuk mata kuliah komputasi statistika ini dengan lancar. Buku ini membahas materi pembelajaran komputasi Statistika menggunakan software utama Microsoft Excel dan R. Secara umum, buku ini menyajikan materi pembelajaran yang terdiri dari fungsi-fungsi dan manajemen data pada Microsoft Excel, manajemen data dan file pada R, operasi dasar dan fungsi dasar pada microsoft excel dan R, struktur pemrograman, aljabar linear, distribusi probabilitas, statistika inferensia, teknik optimasi dasar dan lanjut, R Project/R Markdown, visualisasi data dengan R, serta komputasi statistika lanjut.

Ucapan terima kasih penulis sampaikan kepada semua pihak yang telah membantu dalam rangka penyelesaian buku ini. Penulis menyadari bahwa buku ini tentunya masih perlu perbaikan, sehingga penulis mengharapkan kritik dan saran yang membangun dari para pembaca agar buku ini dapat menjadi lebih baik.

Surabaya, Oktober 2024

Penulis

KONTRIBUTOR

Elina Normayanti
Kharin Octavian Ranto
Renata Amalia Larasati

DAFTAR ISI

PRAKATA	iii
KONTRIBUTOR.....	iv
DAFTAR ISI.....	i
DAFTAR GAMBAR	v
DAFTAR TABEL.....	xiv
BAGIAN I	1
PENDAHULUAN	1
1.1 Komputasi Statistika	1
1.2 Pengantar Excel	1
1.3 Pengantar R	2
BAGIAN II	3
<i>MICROSOFT OFFICE EXCEL</i>	3
2.1 Mengelola <i>Microsoft Office Excel</i>	3
2.1.1 Membuat <i>Workbook</i> Baru	3
2.1.2 Menambahkan <i>Quick Acces Toolbar</i>	4
2.1.3 Validasi Data (<i>Data Validation</i>).....	5
2.1.4 Menyajikan Data dalam Bentuk Grafik	10
2.1.5 Menemukan nilai fungsi x dan Menyajikannya dalam Bentuk Grafik	13
2.1.6 Mencari Nilai Sin, Cos, Tan dan Menyajikannya dalam Bentuk Grafik	16
2.1.7 Menggunakan Fungsi <i>AND</i> dan <i>OR</i>	20
2.1.8 Menggunakan Fungsi <i>RANDBETWEEN</i> dan <i>IF</i>	26
2.1.9 Menggunakan Fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i>	33
2.1.10 Perkalian dan Invers Matriks.....	36
2.1.11 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal	39
2.1.12 Permutasi Dan Kombinasi	41
2.1.13 Menggunakan Fungsi <i>VLOOKUP</i> dan <i>HLOOKUP</i>	44
2.1.14 Menggunakan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i>	52
2.1.15 Tabel Frekuensi	58

2.1.16Pivot Tabel	64
2.2 Mengelola <i>Visual Basic</i>	69
2.2.1 Mengaktifkan <i>Visual Basic</i> pada Microsoft Excel.....	69
2.2.2 Membuat Program untuk Menginput Biodata Melalui <i>Visual Basic</i> ke dalam Bentuk <i>Microsoft Excel</i>	70
2.2.3 Membuat Mesin Hitung <i>Factorial</i> dengan Menggunakan <i>Visual Basic</i>	73
2.2.4 Membuat Mesin Hitung Kombinasi dengan Menggunakan <i>Visual Basic</i>	75
2.2.5 Menerapkan Penggunaan Sistem Looping pada <i>Visual Basic</i>	76
BAGIAN III.....	79
Pengenalan Bahasa R.....	79
3.1 Manajemen Data.....	79
3.1.1 Vektor.....	79
3.1.2 Matriks	80
3.1.3 <i>Data Frame</i>	81
3.1.4 List	81
3.1.5 Array	82
3.1.6 Factor	82
3.2 Operasi Dasar dan Fungsi Dasar R.....	83
3.2.1 Operator Aritmetika.....	83
3.2.2 Operator Relasi.....	84
3.2.3 Operator Logika.....	85
3.2.4 Operator Prioritas.....	86
3.2.5 Operator Keterkaitan.....	86
3.2.6 Fungsi Dasar Statistika di R	86
3.3 Manajemen File	89
3.3.1 <i>Import File</i>	89
3.3.2 <i>Export File</i>	92
3.3.3 Menyimpan <i>File Format R</i>	93
3.4 Struktur Pemrograman R.....	94
3.4.1 Instruksi Pemilihan dengan Fungsi <i>IF ()</i>	95
3.4.2 <i>Switch</i>	96
3.4.3 Instruksi Perulangan (<i>Loop</i>)	97
3.4.4 Function Standar di R.....	100
3.4.5 <i>R-Package</i>	101
3.5 Visualisasi Data (<i>Data Visualization</i>).....	101

3.5.1 Fungsi Plot ().....	102
3.5.2 Fungsi Hist ().....	104
3.5.3 Menambahkan Jendela Grafik	105
3.5.4 Membuat Partisi Windows	105
3.5.5 Fungsi <i>Curve</i> ().....	105
3.5.6 Fungsi <i>Barplot</i> ().....	106
3.5.7 Fungsi <i>Density</i>	107
3.5.8 Fungsi <i>Boxplot</i>	108
3.5.9 Plot Dua Dimensi dan Tiga Dimensi.....	109
3.5.10 Fungsi <i>Factor</i> untuk Data Kategorik.....	110
3.5.11 Grafik Data Kategorik.....	110
3.5.12 Fungsi <i>ggplot</i>	112
3.5.13 Mengaktifkan Data <i>Library Datasets</i>	113
3.6 Komputasi Aljabar Linear.....	114
3.6.1 Input Matriks dengan Perintah <i>cbind</i>	114
3.6.2 Input Matriks dengan Perintah <i>rbind</i>	115
3.6.3 Membuat Beberapa Jenis Matriks.....	115
3.6.4 Operasi-Operasi Pada Matriks.....	116
3.6.5 Fungsi Aljabar Linear Matriks	117
3.6.6 Fungsi Aljabar Linear Vektor.....	117
3.7 Distribusi Peluang.....	118
3.7.1 Distribusi Peluang.....	118
3.7.2 Distribusi Peluang Diskrit	120
3.7.3 Distribusi Peluang Kontinyu	122
3.8 Statistika Inferensia.....	123
3.8.1 Pengujian Hipotesis Satu Populasi	124
3.8.2 Pengujian Hipotesis Dua Populasi.....	126
3.8.3 Analysis of Variance (ANOVA) One Way	128
3.8.4 Analysis of Variance (ANOVA) Two Way	130
3.8.5 Regresi Linear Berganda.....	133
3.9 Optimasi.....	134
3.9.1 Fungsi Optimasi Univariat (<i>optimize & Optimise</i>)	135
3.9.2 Fungsi Optimasi Akar Univariat (<i>uniroot</i>).....	135
3.9.3 Metode <i>Bisection (BFzero)</i>	136
3.9.4 Fungsi Optimasi Advanced (<i>nlminb</i>).....	137
3.9.5 Fungsi Optimasi Regresi dengan <i>Least Square</i>	139
3.10 Perapihan Data (<i>Data Tidying</i>).....	140

3.10.1	Membuat Data Tibble	140
3.10.2	Merubah Struktur Data (<i>Reshape Data</i>).....	141
3.10.3	Memisahkan dan Menggabungkan <i>Cells</i>	142
3.10.4	Cek dan Handling Missing Value	143
3.11	Manipulasi Data (<i>Data Manipulation</i>).....	145
3.11.1	Manipulasi Baris/ <i>Cases</i>	146
3.11.2	Manipulasi Kolom/ <i>Variabel</i>	152
3.11.3	Membuat Variabel Baru	156
3.11.4	Menyimpulkan Baris dan Kolom.....	161
3.11.5	Operator Pipes	162
3.11.6	Mengkategorikan Baris	164
3.11.7	Memberi Nama Baris	165
3.11.8	Mengkombinasikan Tabel atau Variabel.....	166
3.11.9	Mengkombinasikan Baris/ <i>Cases/Row</i>	169
3.12	R <i>Project</i> dan R Markdown.....	176
3.12.1	R <i>Project</i>	176
3.12.2	R Markdown	178
3.13	R Shiny.....	182
3.13.1	Aplikasi R Shiny Sederhana.....	183
3.13	R Shiny.....	187
BAGIAN IV.....		189
KOMPUTASI STATISTIKA LANJUT.....		189
4.1	Metode untuk Membangkitkan Variabel Acak.....	189
4.1.1	Metode Transformasi Invers.....	189
4.1.2	Aplikasi dalam Distribusi Kontinu	190
4.1.3	Aplikasi dalam Distribusi Diskrit	190
4.1.4	Metode <i>Acceptance-Rejection</i>	191
4.1.5	Metode Transformasi	193
4.2	Integrasi Monte Carlo	195
4.2.1	Estimator Sederhana Monte Carlo	195
4.2.2	<i>Variance Reduction</i>	196
4.3	Bootstrap dan Jacknife	197
4.3.1	<i>Bootstrap</i>	197
4.3.2	<i>Jacknife</i>	200
4.3.3	Pengaplikasian <i>Cross</i>	200
TENTANG PENULIS		205

DAFTAR GAMBAR

Gambar 2.1 Membuat <i>Workbook</i> Baru (1).....	3
Gambar 2.2 Membuat <i>Workbook</i> Baru (2).....	4
Gambar 2.3 Menambahkan <i>Quick Access Toolbar</i> (1).....	4
Gambar 2.4 Menambahkan <i>Quick Access Toolbar</i> (2).....	5
Gambar 2.5 Validasi Data (1).....	5
Gambar 2.6 Validasi Data (2).....	6
Gambar 2.7 Validasi Data (3).....	6
Gambar 2.8 Validasi Data (4).....	7
Gambar 2.9 Validasi Data (5).....	7
Gambar 2.10 Validasi Data (6).....	8
Gambar 2.11 Validasi Data (7).....	8
Gambar 2.12 Validasi Data (8).....	9
Gambar 2.13 Validasi Data (9).....	9
Gambar 2.14 Validasi Data (10).....	10
Gambar 2.15 Menyajikan Data dalam Bentuk Grafik (1).....	10
Gambar 2.16 Menyajikan Data dalam Bentuk Grafik (2).....	10
Gambar 2.17 Menyajikan Data dalam Bentuk Grafik (3).....	11
Gambar 2.18 Menyajikan Data dalam Bentuk Grafik (4).....	11
Gambar 2.19 Menyajikan Data dalam Bentuk Grafik (5).....	11
Gambar 2.20 Menyajikan Data dalam Bentuk Grafik (6).....	12
Gambar 2.21 Menyajikan Data dalam Bentuk Grafik (7).....	12
Gambar 2.22 Menyajikan Data dalam Bentuk Grafik (8).....	13
Gambar 2.23 Menemukan Nilai Fungsi x (1).....	13
Gambar 2.24 Menemukan Nilai Fungsi x (2).....	14
Gambar 2.25 Menemukan Nilai Fungsi x (3).....	14
Gambar 2.26 Menemukan Nilai Fungsi x (4).....	15
Gambar 2.27 Menemukan Nilai Fungsi x (5).....	15
Gambar 2.28 Menemukan Nilai Fungsi x (6).....	16
Gambar 2.29 Mencari nilai Sin, Cos, dan Tan (1).....	16
Gambar 2.30 Mencari nilai Sin, Cos, dan Tan (2).....	17

Gambar 2.31 Mencari nilai Sin, Cos, dan Tan (3).....	17
Gambar 2.32 Mencari nilai Sin, Cos, dan Tan (4).....	18
Gambar 2.33 Mencari nilai Sin, Cos, dan Tan (5).....	19
Gambar 2.34 Mencari nilai Sin, Cos, dan Tan (6).....	19
Gambar 2.35 Mencari nilai Sin, Cos, dan Tan (7).....	20
Gambar 2.36 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (1).....	20
Gambar 2.37 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (2).....	21
Gambar 2.38 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (3).....	21
Gambar 2.39 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (4).....	22
Gambar 2.40 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (5).....	22
Gambar 2.41 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (6).....	23
Gambar 2.42 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (7).....	23
Gambar 2.43 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (8).....	23
Gambar 2.44 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (9).....	24
Gambar 2.45 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (10).....	24
Gambar 2.46 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (11).....	24
Gambar 2.47 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (12).....	25
Gambar 2.48 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (13).....	25
Gambar 2.49 Menggunakan fungsi <i>AND</i> dan <i>OR</i> (14).....	26
Gambar 2.50 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (1).....	26
Gambar 2.51 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (2).....	27
Gambar 2.52 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (3).....	27
Gambar 2.53 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (4).....	28
Gambar 2.54 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (5).....	28
Gambar 2.55 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (6).....	28
Gambar 2.56 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (7).....	29
Gambar 2.57 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (8).....	29
Gambar 2.58 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (9).....	29
Gambar 2.59 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (10).....	30
Gambar 2.60 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (11).....	30
Gambar 2.61 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (12).....	31
Gambar 2.62 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (13).....	31
Gambar 2.63 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (14).....	32

Gambar 2.64 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (15)	32
Gambar 2.65 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (16)	33
Gambar 2.66 Menggunakan fungsi <i>RANDBETWEEN</i> dan <i>IF</i> (17)	33
Gambar 2.67 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (1).....	34
Gambar 2.68 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (2).....	34
Gambar 2.69 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (3).....	34
Gambar 2.70 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (4).....	35
Gambar 2.71 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (5)	35
Gambar 2.72 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (6).....	35
Gambar 2.73 Menggunakan fungsi <i>LEN</i> , <i>LEFT</i> , <i>MID</i> , <i>RIGHT</i> (7).....	36
Gambar 2.74 Perkalian dan Invers Matriks (1)	36
Gambar 2.75 Perkalian dan Invers Matriks (2)	37
Gambar 2.76 Perkalian dan Invers Matriks (3)	37
Gambar 2.77 Perkalian dan Invers Matriks (4)	38
Gambar 2.78 Perkalian dan Invers Matriks (5)	38
Gambar 2.79 Perkalian dan Invers Matriks (6)	38
Gambar 2.80 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (1)	39
Gambar 2.81 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (2)	39
Gambar 2.82 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (3)	40
Gambar 2.83 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (4)	40
Gambar 2.84 Menggunakan Fungsi <i>ISEVEN</i> , <i>ISODD</i> , <i>MOD</i> , Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (5)	40

Gambar 2.85 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (6)	41
Gambar 2.86 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (7)	41
Gambar 2.87 Permutasi (1)	41
Gambar 2.88 Permutasi (2)	42
Gambar 2.89 Permutasi k dari n (${}_nP_k$) (1).....	42
Gambar 2.90 Permutasi k dari n (${}_nP_k$) (2).....	42
Gambar 2.91 Permutasi k dari n (${}_nP_k$) (3).....	43
Gambar 2.92 Permutasi k dari n (${}_nP_k$) (4).....	43
Gambar 2.93 Kombinasi (1).....	43
Gambar 2.94 Kombinasi (2).....	44
Gambar 2.95 Kombinasi (3).....	44
Gambar 2.96 Latihan fungsi <i>VLOOKUP</i> (1).....	44
Gambar 2.97 Latihan fungsi <i>VLOOKUP</i> (2).....	45
Gambar 2.98 Latihan fungsi <i>VLOOKUP</i> (3).....	45
Gambar 2.99 Latihan fungsi <i>VLOOKUP</i> (4).....	45
Gambar 2.100 Latihan fungsi <i>VLOOKUP</i> (5).....	46
Gambar 2.101 Latihan fungsi <i>VLOOKUP</i> (6).....	46
Gambar 2.102 Latihan fungsi <i>VLOOKUP</i> (7).....	47
Gambar 2.103 Latihan fungsi <i>VLOOKUP</i> (8).....	47
Gambar 2.104 Latihan fungsi <i>VLOOKUP</i> (9).....	47
Gambar 2.105 Latihan fungsi <i>VLOOKUP</i> (10).....	48
Gambar 2.106 Latihan fungsi <i>VLOOKUP</i> (11).....	48
Gambar 2.107 Latihan fungsi <i>VLOOKUP</i> (12).....	48
Gambar 2.108 Latihan fungsi <i>VLOOKUP</i> (13).....	49
Gambar 2.109 Latihan fungsi <i>VLOOKUP</i> (14).....	49
Gambar 2.110 Latihan fungsi <i>VLOOKUP</i> (15).....	49
Gambar 2.111 Latihan fungsi <i>HLOOKUP</i> (1)	50
Gambar 2.112 Latihan fungsi <i>HLOOKUP</i> (2)	50
Gambar 2.113 Latihan fungsi <i>HLOOKUP</i> (3)	51

Gambar 2.114 Latihan fungsi <i>HLOOKUP</i> (4)	51
Gambar 2.115 Latihan fungsi <i>HLOOKUP</i> (5)	51
Gambar 2.116 Latihan fungsi <i>HLOOKUP</i> (6).....	52
Gambar 2.117 Latihan fungsi <i>HLOOKUP</i> (7)	52
Gambar 2.118 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (1)	54
Gambar 2.119 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (2)	55
Gambar 2.120 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (3)	55
Gambar 2.121 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (4)	55
Gambar 2.122 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (5)	56
Gambar 2.123 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (6)	56
Gambar 2.124 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (7)	56
Gambar 2.125 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (8)	57
Gambar 2.126 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (9)	57
Gambar 2.127 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (10).....	57
Gambar 2.128 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (11).....	58
Gambar 2.129 Latihan fungsi <i>COUNTIFS</i> dan <i>AVERAGEIF</i> (12).....	58
Gambar 2.130 Latihan Tabel Frekuensi (1)	58
Gambar 2.131 Latihan Tabel Frekuensi (2)	59
Gambar 2.132 Latihan Tabel Frekuensi (3)	59
Gambar 2.133 Latihan Tabel Frekuensi (4)	59
Gambar 2.134 Latihan Tabel Frekuensi (5)	60
Gambar 2.135 Latihan Tabel Frekuensi (6)	60
Gambar 2.136 Latihan Tabel Frekuensi (7)	60
Gambar 2.137 Latihan Tabel Frekuensi (8)	61
Gambar 2.138 Latihan Tabel Frekuensi (9)	61
Gambar 2.139 Latihan Tabel Frekuensi (10)	61
Gambar 2.140 Latihan Tabel Frekuensi (11)	62
Gambar 2.141 Latihan Tabel Frekuensi (12)	62
Gambar 2.142 Latihan Tabel Frekuensi (13)	62
Gambar 2.143 Latihan Tabel Frekuensi (14)	63
Gambar 2.144 Latihan Tabel Frekuensi (15)	63
Gambar 2.145 Latihan Tabel Frekuensi (16)	63
Gambar 2.146 Latihan Pivot Tabel (1)	66

Gambar 2.147 Latihan Pivot Tabel (2)	66
Gambar 2.148 Latihan Pivot Tabel (3)	66
Gambar 2.149 Latihan Pivot Tabel (4)	67
Gambar 2.150 Latihan Pivot Tabel (5)	67
Gambar 2.151 Latihan Pivot Tabel (6)	68
Gambar 2.152 Latihan Pivot Tabel (7)	68
Gambar 2.153 Latihan Pivot Tabel (8)	69
Gambar 2.154 Latihan Pivot Tabel (9)	69
Gambar 2.155 Ms. Excel.....	69
Gambar 2.156 Option Ms. Excel	70
Gambar 2.157 Visual Basic.....	70
Gambar 2.158 Halaman <i>Visual Basic</i>	71
Gambar 2.159 <i>Tool Box</i>	71
Gambar 2.160 <i>View Code</i>	71
Gambar 2.161 Fungsi <i>Visual Basic</i>	72
Gambar 2.162 Menjalankan <i>Visual Basic</i>	73
Gambar 2.163 Tampilan <i>Input Biodata</i>	73
Gambar 2.164 Menampilkan <i>Tool Box</i>	73
Gambar 2.165 Menuliskan Fungsi Faktorial.....	74
Gambar 2.166 Menjalankan <i>Visual Basic</i>	74
Gambar 2.167 Tampilan Faktorial.....	74
Gambar 2.168 Membuat <i>Work Sheet</i>	75
Gambar 2.169 Menuliskan Fungsi Kombinasi.....	75
Gambar 2.170 Menjalankan <i>Visual Basic</i>	75
Gambar 2.171 Tampilan Kombinasi.....	76
Gambar 2.172 Menambahkan <i>Command Button</i>	77
Gambar 2.173 Menuliskan Fungsi.....	77
Gambar 2.174 Menjalankan <i>Visual Basic</i>	77
Gambar 3.1 Tampilan <i>File Text</i> Menggunakan Notepad.....	89
Gambar 3.2 Tampilan <i>File Excel</i>	90
Gambar 3.3 Tampilan Untuk Menyimpan <i>File</i> Dalam Bentuk <i>Text</i>	90
Gambar 3.4 Tampilan Untuk Menyimpan <i>File</i> Dalam Bentuk <i>CSV</i>	91
Gambar 3.5 Tampilan <i>File</i> Dalam Notepad	91

Gambar 3.6 Langkah Membuka <i>File Text</i> dengan Pemisah Spasi	91
Gambar 3.7 Langkah Awal Menyimpan <i>File Format R</i>	93
Gambar 3.8 Tampilan <i>Save Workspace As</i>	93
Gambar 3.9 Langkah Awal Membaca File yang Telah Dibuat	94
Gambar 3.10 Tampilan <i>Load Workspace</i>	94
Gambar 3.11 Grafik Fungsi Kuadratik	103
Gambar 3.12 Histogram Distribusi Normal (0,1)	105
Gambar 3.13 <i>Bar chart</i> Dataset <i>VADeaths</i>	106
Gambar 3.14 <i>Bar chart</i> Dataset <i>VADeaths</i> Kustomisasi	107
Gambar 3.15 <i>Stacked Bar Chart</i> Dataset <i>VADeaths</i>	107
Gambar 3.16 Grafik <i>Density</i> Dataset <i>Tree</i>	108
Gambar 3.17 <i>Boxplot</i> Dataset <i>Iris</i>	109
Gambar 3.18 Grafik Data Kategorik	111
Gambar 3.19 <i>Pie Chart</i> Kota	112
Gambar 3.20 Mengaktifkan Data <i>Library Datasets</i>	113
Gambar 3.20 Mengaktifkan Data <i>Library Datasets</i>	113
Gambar 3.21 <i>Datasets</i> yang Tersedia di R	114
Gambar 3.22 Melihat Distribusi di R	119
Gambar 3.23 Melihat Distribusi di R (Lanjutan)	119
Gambar 3.24 Penjelasan Plot Distribusi Normal	120
Gambar 3.25 Penjelasan Plot Distribusi Normal (Lanjutan)	120
Gambar 3.26 <i>Output Spread</i>	142
Gambar 3.27 <i>Output Separate</i>	142
Gambar 3.28 <i>Output Separate_rows</i>	143
Gambar 3.29 <i>Output Unite</i>	143
Gambar 3.30 <i>Output Cek Missing Value (1)</i>	144
Gambar 3.31 <i>Output Cek Missing Value (2)</i>	144
Gambar 3.32 <i>Output Handling Missing Value (1)</i>	145
Gambar 3.33 <i>Output Handling Missing Value (2)</i>	145
Gambar 3.34 <i>Output Handling Missing Value (3)</i>	145
Gambar 3.35 Tampilan Data “ <i>mtcars</i> ”	146
Gambar 3.36 <i>Output Filter</i>	146
Gambar 3.37 <i>Output Distinct</i>	147

Gambar 3.38 <i>Output Slice</i>	148
Gambar 3.39 <i>Output Slice</i> (1)	148
Gambar 3.40 <i>Output Slice</i> (2)	149
Gambar 3.41 <i>Output Slice</i> (3)	149
Gambar 3.42 <i>Output Slice</i> (4)	150
Gambar 3.43 <i>Output Slice</i> (5)	150
Gambar 3.44 <i>Output Arrange</i>	151
Gambar 3.45 <i>Output Arrange</i> (1)	151
Gambar 3.46 <i>Output Add_Row</i>	152
Gambar 3.47 <i>Output Pull</i>	152
Gambar 3.48 <i>Output Select</i>	153
Gambar 3.49 <i>Output Relocate</i>	156
Gambar 3.50 <i>Output Mutate</i>	156
Gambar 3.51 <i>Output Nilai Variabel Baru</i>	157
Gambar 3.52 <i>Output Transmute</i>	157
Gambar 3.53 <i>Output Rename</i>	160
Gambar 3.54 <i>Ilustrasi Pipes 1</i>	162
Gambar 3.55 <i>Ilustrasi Pipes 2</i>	163
Gambar 3.56 <i>Output Select, Filter, Arrange</i>	164
Gambar 3.57 <i>Output Group_by</i> (Sumber: Penulis)	164
Gambar 3.58 <i>Output Rowwise</i> (Sumber: Penulis)	165
Gambar 3.59 <i>Output Rownames</i> (Sumber: Penulis)	165
Gambar 3.60 <i>Output Mengkombinasikan Variabel</i>	166
Gambar 3.61 <i>Ilustrasi Penggabungan Data</i>	166
Gambar 3.62 <i>Output Left_join</i>	167
Gambar 3.63 <i>Output Right_join</i>	167
Gambar 3.64 <i>Output Inner_join</i>	168
Gambar 3.65 <i>Output Full_join Berdasarkan Variabel Bernilai Sama</i>	168
Gambar 3.66 <i>Output Full_join Berdasarkan Nama Variabel yang Berbeda</i>	169
Gambar 3.67 <i>Output Mengkombinasikan Baris</i>	170
Gambar 3.68 <i>Output Mengkombinasikan Baris</i> (1)	170

Gambar 3.69 <i>Output Filter Baris Semi_join Berdasarkan Nama Variabel yang Sama</i>	171
Gambar 3.70 <i>Output Semi_join Berdasarkan Nama Variabel yang Berbeda</i>	171
Gambar 3.71 <i>Output Filter Baris Anti_join Berdasarkan Nama Variabel yang Sama</i>	172
Gambar 3.72 <i>Output Anti_join Berdasarkan Nama Variabel yang Berbeda</i>	172
Gambar 3.73 <i>Output Filter Baris Nest_join Berdasarkan Nama Variabel yang Sama</i>	173
Gambar 3.74 <i>Output Filter Baris Nest_join Berdasarkan Nama Variabel yang Berbeda</i>	173
Gambar 3.75 <i>Ilustrasi Penggabungan Relational Data</i>	174
Gambar 3.76 <i>Output Intersect</i>	174
Gambar 3.77 <i>Output Union_all</i>	175
Gambar 3.78 <i>Output Setdiff</i>	176
Gambar 3.79 <i>Menampilkan New Project</i>	177
Gambar 3.80 <i>Memilih Jenis Project</i>	177
Gambar 3.81 <i>Tampilan New Directory</i>	178
Gambar 3.82 <i>Tampilan Create New Project</i>	178
Gambar 3.83 <i>Langkah Awal Membuka R-Markdown</i>	179
Gambar 3.84 <i>Tampilan New R-Markdown</i>	179
Gambar 3.85 <i>Tampilan R Setelah Menggunakan R-Markdown</i>	180
Gambar 3.86 <i>Penjelasan Tentang Tampilan R-Markdown</i>	180
Gambar 3.87 <i>Komponen Shiny (Sumber: Penulis)</i>	183
Gambar 3.88 <i>Langkah Awal Membuat Aplikasi R Shiny</i>	183
Gambar 3.89 <i>Membuat Folder di R Shiny</i>	184
Gambar 3.90 <i>Tampilan R Studio Setelah Menggunakan R</i>	184
Gambar 3.91 <i>Langkah Untuk Menampilkan Histogram</i>	185
Gambar 3.92 <i>Hasil Proses Coding</i>	185
Gambar 3.93 <i>Tampilan Histogram Interaktif</i>	186
Gambar 3.94 <i>Tampilan Untuk Melakukan Publikasi</i>	186
Gambar 3.95 <i>Tampilan Untuk Melakukan Publikasi (Lanjutan)</i>	187

DAFTAR TABEL

Tabel 3.1 Operator Aritmetika	83
Tabel 3.2 Operator Relasi	84
Tabel 3.3 Operator Logika	85
Tabel 3.4 Fungsi Dasar Statistika R	86
Tabel 3.5 Operator Logika untuk Filter ().....	147
Tabel 3.6 Fungsi <i>Select</i> ()	153
Tabel 3.7 Fungsi <i>Mutate</i> () dan <i>Transmute</i> ().....	158
Tabel 3.8 Fungsi <i>Summarise</i> ().....	161

BAGIAN I

PENDAHULUAN

Capaian Pembelajaran: Setelah mempelajari bagian ini mahasiswa diharapkan dapat menjelaskan ruang lingkup materi perkuliahan serta perangkat lunak yang dipergunakan.

1.1 Komputasi Statistika

Komputasi dapat diartikan sebagai cara untuk menemukan pemecahan masalah dari data *input* dengan menggunakan suatu algoritma. Secara umum, ilmu komputasi adalah bidang ilmu yang memiliki peran dalam penyusunan model matematika dan teknik penyelesaian numerik serta penggunaan komputer untuk menganalisis dan memecahkan masalah-masalah ilmu (sains). Sedangkan Statistika merupakan cabang ilmu yang mempelajari bagaimana cara merencanakan, mengumpulkan, menganalisis, hingga menginterpretasikan data. Dari kumpulan data, statistika dapat digunakan untuk menyimpulkan atau mendeskripsikan data. Akan tetapi, pada beberapa permasalahan, data yang diperoleh dilapangan tidak sedikit. Sehingga sangat sulit menganalisis data secara manual dalam jumlah yang banyak. Oleh karena itu peran komputasi sangat dibutuhkan untuk membantu mengolah data-data tersebut menggunakan *software* statistika. Sehingga komputasi statistika merupakan cabang ilmu statistika yang berkaitan dengan pengolahan dan pemecahan masalah data dengan menggunakan *software* statistika. Beberapa *software* untuk mengolah data antara lain, R, Python, SPSS, Minitab, SAS, Eviews.

1.2 Pengantar Excel

Excel adalah program spreadsheet dalam sistem microsoft office. Excel dapat digunakan untuk membuat dan memformat *workbook* untuk menganalisis data dan membuat data itu sendiri. Kita dapat menggunakan excel untuk melacak data, membangun model untuk menganalisis data, menulis rumus untuk melakukan perhitungan data, memproses data dalam berbagai cara, dan menyajikan data dalam bentuk grafik. Lembar kerja dalam Excel dalam satu *sheet* terdiri dari 256 kolom dan 65536 baris. Kolom ditampilkan dalam tanda huruf A,B,C dan berakhir pada kolom IV. Sedangkan baris dilambangkan dengan angka 1,2,4, dan berakhir pada

65536. Perpotongan antara baris dan kolom disebut sel, dan sel yang bergaris tebalandakan bahwa sel tersebut dalam keadaan aktif.

1.3 Pengantar R

R (juga dikenal sebagai GNU S) adalah bahasa pemrograman dan perangkat lunak untuk analisis statistika dan grafik. R dibuat oleh Ross Ihaka dan Robert Gentleman di Universitas Auckland, Selandia Baru, dan kini dikembangkan oleh *R Development Core Team*, di mana Chambers merupakan anggotanya. R dinamakan sebagian setelah nama dua pembuatnya (Robert Gentleman dan Ross Ihaka), dan sebagian sebagian dari permainan nama dari S.

R menyediakan berbagai teknik statistika (permodelan linier dan nonlinier, uji statistik klasik, analisis deret waktu, klasifikasi, klusterisasi, dan sebagainya) serta grafik. R dirancang sebagai bahasa komputer sebenarnya, dan mengizinkan penggunanya untuk menambah fungsi tambahan dengan mendefinisikan fungsi baru. Hal inilah yang menjadi salah satu kelebihan R dibandingkan *software* statistika lain. Karena sangat mudah untuk dikembangkan.

BAGIAN II

MICROSOFT OFFICE EXCEL

Capaian Pembelajaran: Setelah mempelajari bagian ini mahasiswa diharapkan mampu:

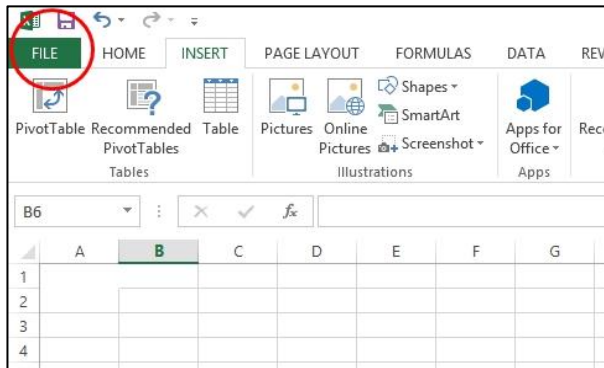
1. Memahami struktur dan tipe data, fungsi dasar hingga library yang terdapat pada excel untuk melakukan manajemen data dan analisis data.
2. Membuat program komputasi sederhana dengan Visual Basic

2.1 Mengelola Microsoft Office Excel

2.1.1 Membuat Workbook Baru

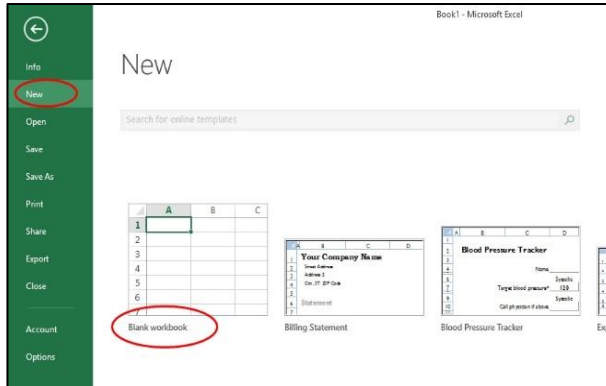
Langkah-langkah untuk membuat *workbook* baru adalah sebagai berikut.

1. Pilih *File*



Gambar 2.1 Membuat *Workbook* Baru (1) (Sumber: Penulis)

2. Pilih *New*, kemudian pilih *Blank Workbook*

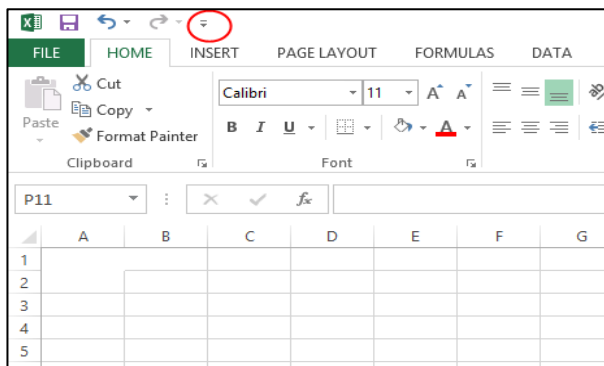


Gambar 2.2 Membuat *Workbook* Baru (2) (Sumber: Penulis)

2.1.2 Menambahkan *Quick Acces Toolbar*

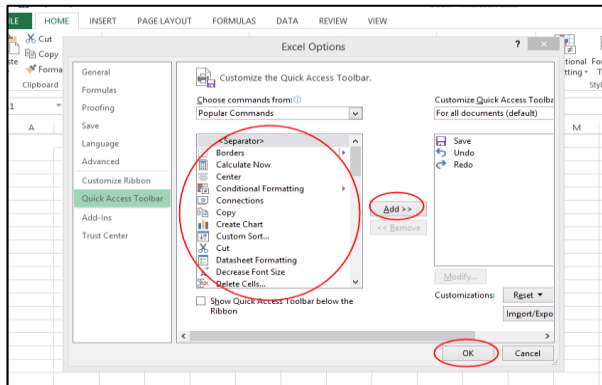
Langkah-langkah untuk menambahkan *Quick Access Toolbar* dapat dilakukan sebagai berikut.

1. Klik *dropdown* pada *Quick Access Toolbar*, kemudian pilih *More Commands*,



Gambar 2.3 Menambahkan *Quick Access Toolbar* (1) (Sumber: Penulis)

- Pilih salah satu yang akan ditambahkan pada *Quick Access Toolbar*, pilih *Add*, lalu *OK*.



Gambar 2.4 Menambahkan *Quick Access Toolbar* (2) (Sumber: Penulis)

2.1.3 Validasi Data (*Data Validation*)

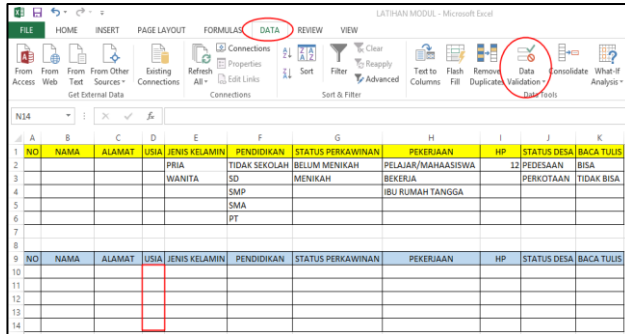
Validasi Data bertujuan untuk membatasi tipe data atau nilai yang dimasukkan oleh pengguna ke dalam sel. Latihan validasi data dapat dilakukan dengan membuat tabel sebagai berikut.

Clipboard		Font		Alignment		Number		Styles	
G11									
				</					

Gambar 2.5 Validasi Data (1) (Sumber: Penulis)

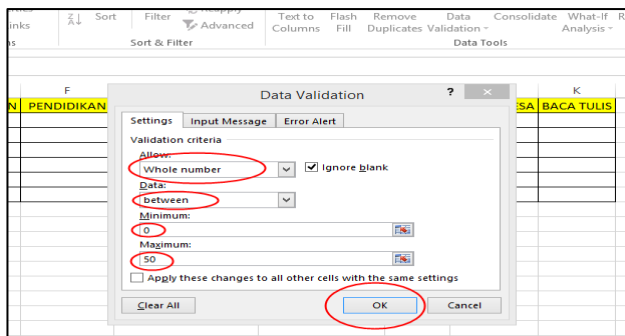
Pada kolom usia, data yang dimasukkan harus berupa angka atau bilangan. Jika data yang dimasukkan bukan angka atau bilangan maka akan keluar peringatan *error*. Sehingga data yang *diinput* menjadi benar. Langkah-langkah yang dapat dilakukan adalah sebagai berikut.

1. Blok kolom usia, kemudian pada Tab *Data*, pilih *Data Validation*.



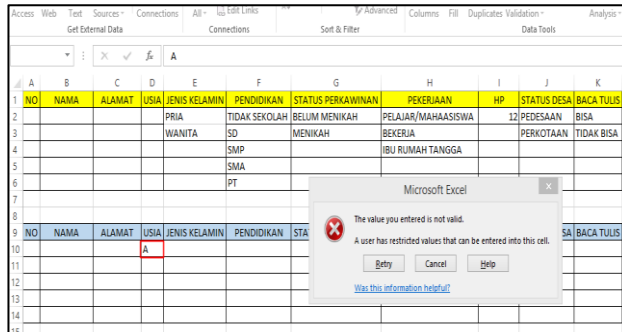
Gambar 2.6 Validasi Data (2) (Sumber: Penulis)

2. Kemudian atur *Allow* menjadi *Whole number*, atur data menjadi *Between*, dan tentukan nilai minimum dan maksimum usia yang akan diinput ke dalam tabel. Kemudian pilih *OK*.



Gambar 2.7 Validasi Data (3) (Sumber: Penulis)

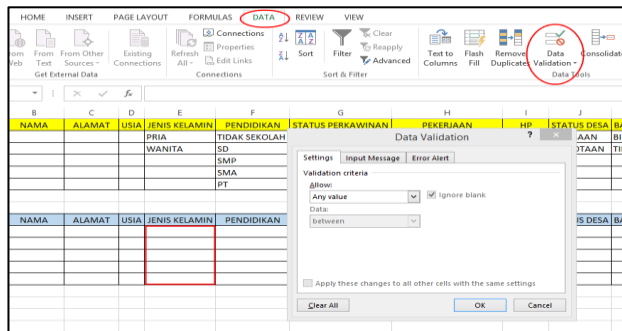
3. Jika data yang dimasukkan pada kolom usia bukan merupakan angka, maka akan muncul peringatan *error* seperti gambar di bawah ini.



Gambar 2.8 Validasi Data (4) (Sumber: Penulis)

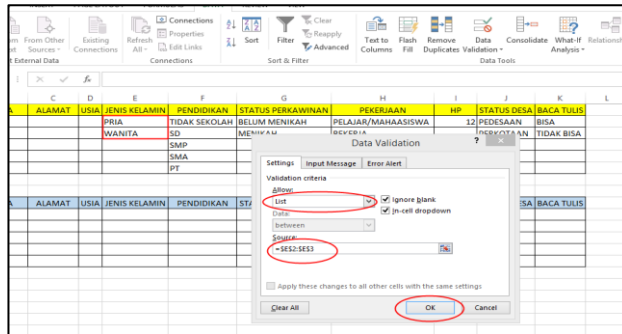
Pada kolom Jenis Kelamin, maka data yang dimasukkan hanya ada dua pilihan yaitu laki-laki dan perempuan dengan cara membuat *dropdown list*. Dengan menggunakan validasi data, dapat dilakukan langkah-langkah sebagai berikut.

1. Blok kolom jenis kelamin pada tabel yang berwarna biru, kemudian pada *Tab Data*, pilih *Data Validation*, maka akan muncul tampilan sebagai berikut.



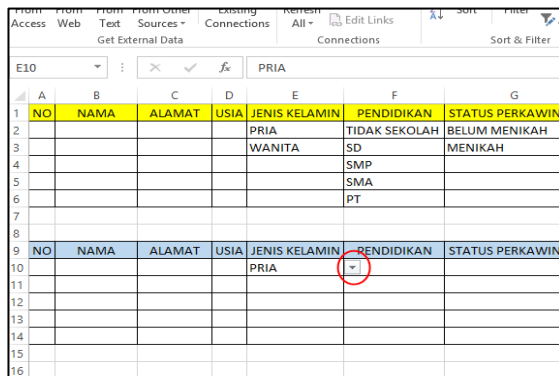
Gambar 2.9 Validasi Data (5) (Sumber: Penulis)

2. Kemudian pada tampilan *data validation*, atur *allow* menjadi *list*, atur *source* dengan cara mem-blok area sel pada kolom jenis kelamin pada tabel yang berwarna kuning, kemudian pilih OK.



Gambar 2.10 Validasi Data (6) (Sumber: Penulis)

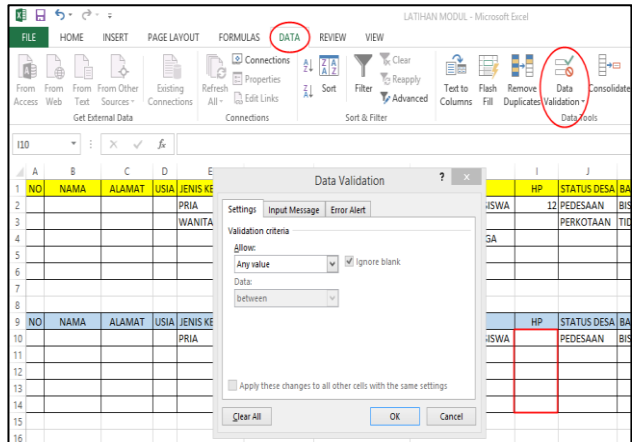
3. Kemudian pilih salah satu jenis kelamin dengan cara klik tanda segitiga di samping sel untuk melihat daftar jenis kelamin.



Gambar 2.11 Validasi Data (7) (Sumber: Penulis)

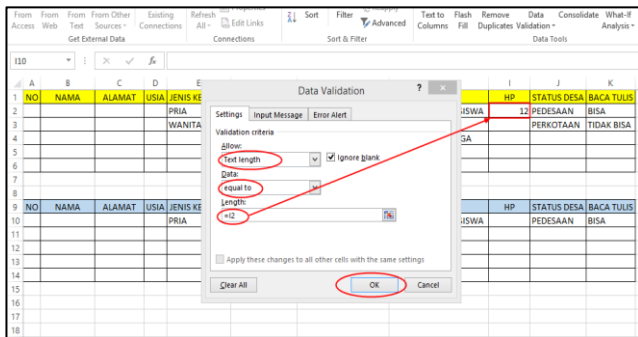
Untuk mengisi kolom pendidikan, status perkawinan, pekerjaan, status desa, dan baca tulis dapat diterapkan langkah-langkah yang sama yaitu dengan menggunakan *dropdown list*. Sedangkan untuk mengisi kolom HP hanya dibatasi untuk 12 angka, dapat diterapkan langkah-langkah sebagai berikut.

1. Blok sel yang ada pada kolom HP, kemudian pada *Tab Data* pilih *data validation*, kemudian muncul tampilan sebagai berikut.



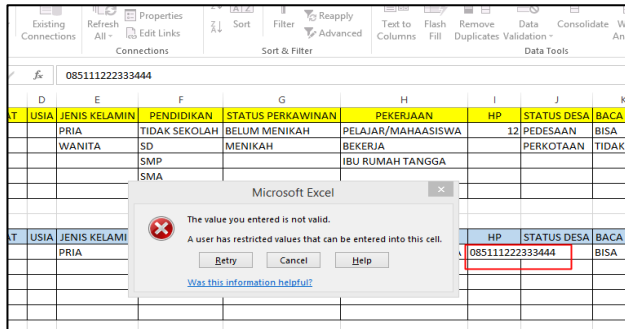
Gambar 2.12 Validasi Data (8) (Sumber: Penulis)

2. Kemudian pada tampilan *data validation*, atur *allow* menjadi *text lenght*, atur *data* menjadi *equal to*, dan atur *source* dengan cara klik sel pada kolom HP pada tabel yang berwarna kuning, kemudian pilih OK.



Gambar 2.13 Validasi Data (9) (Sumber: Penulis)

3. Kemudian jika data yang dimasukkan lebih dari 12 digit, maka akan muncul peringatan *error* seperti pada gambar di bawah ini.



Gambar 2.14 Validasi Data (10) (Sumber: Penulis)

2.1.4 Menyajikan Data dalam Bentuk Grafik

Langkah-langkah yang dapat dilakukan adalah sebagai berikut.

1. Ketik data yang akan disajikan dalam bentuk grafik. Contoh :

	A	B	C	D	E	F	G
	Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas	Persentase Penduduk	Persentase Luas
1							
2	Bali dan Kep.Nusantara	14.109	73.070				
3	Kep. Maluku	2.849	78.897				
4	Jawa	145.144	129.438				
5	Sulawesi	18.724	188.522				
6	Papua	4.021	418.708				
7	Sumatera	35.273	480.793				
8	Kalimantan	15.343	544.150				
9	Indonesia	255.463	1.913.578				

Gambar 2.15 Menyajikan Data dalam Bentuk Grafik (1) (Sumber: Penulis)

2. Untuk melengkapi kolom penduduk, dapat digunakan rumus sebagai berikut, kemudian *drag* sel ke bawah.

	A	B	C	D	E	F	G
	Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas (km ²)	Persentase Penduduk	Persentase Luas
1							
2	Bali dan Kep.Nusantara	14.109	73.070	=B2*1000			
3	Kep. Maluku	2.849	78.897	2.849.000			
4	Jawa	145.144	129.438	145.144.000			
5	Sulawesi	18.724	188.522	18.724.000			
6	Papua	4.021	418.708	4.021.000			
7	Sumatera	35.273	480.793	35.273.000			
8	Kalimantan	15.343	544.150	15.343.000			
9	Indonesia	255.463	1.913.578	255.463.000			

Gambar 2.16 Menyajikan Data dalam Bentuk Grafik (2) (Sumber: Penulis)

3. Untuk melengkapi kolom luas, dapat digunakan rumus sebagai berikut, kemudian drag sel ke bawah.

	A	B	C	D	E	F	G
	Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas (km ²)	Persentase Penduduk	Persentase Luas
1	Bali dan Kep.Nusantara	14,109	73,070	14,109,000	=C2*1000		
2	Kep. Maluku	2,849	78,897	2,849,000	78,897,000		
3	Jawa	145,144	129,438	145,144,000	129,438,000		
4	Sulawesi	18,724	188,522	18,724,000	188,522,000		
5	Papua	4,021	418,708	4,021,000	418,708,000		
6	Sumatera	55,273	480,793	55,273,000	480,793,000		
7	Kalimantan	15,343	544,150	15,343,000	544,150,000		
8	Indonesia	255,483	1,913,578	255,483,000	1,913,578,000		

Gambar 2.17 Menyajikan Data dalam Bentuk Grafik (3) (Sumber: Penulis)

4. Untuk melengkapi kolom presentase penduduk, dapat digunakan rumus sebagai berikut kemudian drag sel ke bawah. Fungsi dari tanda "\$" adalah untuk mengunci supaya jika di drag ke bawah, sel D9 tidak berubah.

	A	B	C	D	E	F	G
	Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas (km ²)	Persentase Penduduk	Persentase Luas
1	Bali dan Kep.Nusantara	14,109	73,070	14,109,000	73,070,000	=D2/D\$9	
2	Kep. Maluku	2,849	78,897	2,849,000	78,897,000	2%	
3	Jawa	145,144	129,438	145,144,000	129,438,000	57%	
4	Sulawesi	18,724	188,522	18,724,000	188,522,000	7%	
5	Papua	4,021	418,708	4,021,000	418,708,000	2%	
6	Sumatera	55,273	480,793	55,273,000	480,793,000	22%	
7	Kalimantan	15,343	544,150	15,343,000	544,150,000	6%	
8	Indonesia	255,483	1,913,578	255,483,000	1,913,578,000		

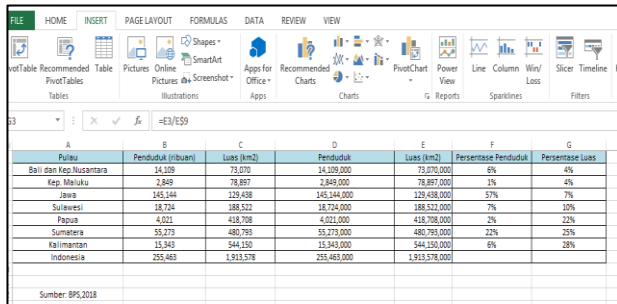
Gambar 2.18 Menyajikan Data dalam Bentuk Grafik (4) (Sumber: Penulis)

5. Untuk melengkapi kolom presentase luas, dapat digunakan rumus sebagai berikut kemudian drag sel ke bawah.

	A	B	C	D	E	F	G
	Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas (km ²)	Persentase Penduduk	Persentase Luas
1	Bali dan Kep.Nusantara	14,109	73,070	14,109,000	73,070,000	6%	=E2/E\$9
2	Kep. Maluku	2,849	78,897	2,849,000	78,897,000	2%	4%
3	Jawa	145,144	129,438	145,144,000	129,438,000	57%	7%
4	Sulawesi	18,724	188,522	18,724,000	188,522,000	7%	10%
5	Papua	4,021	418,708	4,021,000	418,708,000	2%	22%
6	Sumatera	55,273	480,793	55,273,000	480,793,000	22%	25%
7	Kalimantan	15,343	544,150	15,343,000	544,150,000	6%	28%
8	Indonesia	255,483	1,913,578	255,483,000	1,913,578,000		

Gambar 2.19 Menyajikan Data dalam Bentuk Grafik (5) (Sumber: Penulis)

6. Untuk menyajikan data pada tabel tersebut dalam bentuk grafik, blok bagian tabel yang diinginkan, misalnya Pulau, Presentase Penduduk, dan Presentase Luas, kemudian pilih *Insert*, pilih *Chart* yang diinginkan.

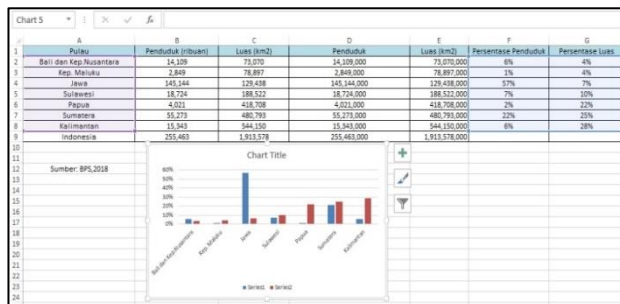


A	B	C	D	E	F	G
Pulau	Penduduk (ribuan)	Luas (km ²)	Penduduk	Luas (km ²)	Persentase Penduduk	Persentase Luas
Bali dan Kep Nusatara	14.109	73.070	14.109.000	73.070.000	6%	4%
Kep. Maluku	2.849	78.897	2.849.000	78.897.000	1%	4%
Jawa	145.144	129.458	145.144.000	129.458.000	57%	7%
Sulawesi	18.724	188.522	18.724.000	188.522.000	7%	10%
Papua	4.021	418.708	4.021.000	418.708.000	2%	22%
Sumatera	15.273	480.793	15.273.000	480.793.000	22%	25%
Kalimantan	15.343	544.150	15.343.000	544.150.000	6%	28%
Indonesia	255.463	1.913.578	255.463.000	1.913.578.000		

Sumber: BPS, 2018

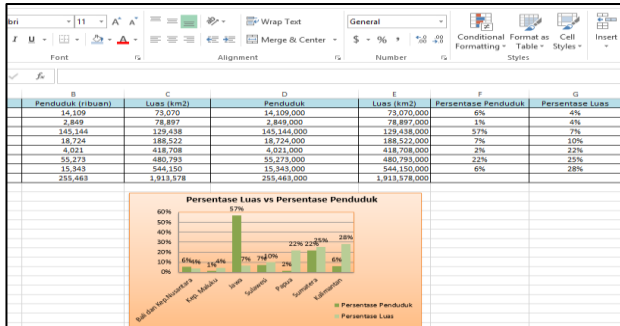
Gambar 2.20 Menyajikan Data dalam Bentuk Grafik (6) (Sumber: Penulis)

7. Untuk menambahkan judul pada grafik, dapat menambahkan tulisan pada *chart title*. Sedangkan untuk mengedit grafik dapat dilakukan dengan klik *chart styles* dan *chart filters* pada bagian kanan grafik atau melalui *Tab Design*.



Gambar 2.21 Menyajikan Data dalam Bentuk Grafik (7) (Sumber: Penulis)

Berikut merupakan tampilan grafik yang sudah diedit.



Gambar 2.22 Menyajikan Data dalam Bentuk Grafik (8) (Sumber: Penulis)

2.1.5 Menemukan nilai fungsi x dan Menyajikannya dalam Bentuk Grafik

1. Fungsi $f(x)=ax+b$

Langkah-langkah yang dapat digunakan adalah sebagai berikut.

- Tentukan nilai a dan b , masing-masing dalam sel yang berbeda. Misalnya diberikan nilai $a=3$ dan $b=4$.
- Tentukan nilai x yang akan digunakan. Buat tabel seperti gambar berikut.

J5				
	A	B	C	D
1	a=	3		
2	b=	4		
3				
4	x	f(x)=ax+b		
5	-5			
6	-4			
7	-3			
8	-2			
9	-1			
10	0			
11	1			
12	2			
13	3			
14	4			
15	5			
16	6			
17	7			
18	8			
19	9			
20	10			
21	11			
22	12			

Gambar 2.23 Menemukan Nilai Fungsi x (1) (Sumber: Penulis)

- Untuk menemukan nilai $f(x)$, dapat digunakan rumus $f(x)=ax+b$ sebagai berikut, kemudian *drag* sel ke bawah.

Clipboard			
Font			
SUM			=A5+B\$2
A	B	C	D
1 START	-4	a=	1
2 STEP	0.1	b=	1
3		c=	-4
4 x	f(x)=ax^2+bx+c		
5	-4		
6	=A5+B\$2		
7	-3.8		
8	-3.7		
9	-3.6		
10	-3.5		
11	-3.4		
12	-3.3		
13	-3.2		
14	-3.1		
15	-3		
16	-2.9		
17	-2.8		
18	-2.7		
19	-2.6		

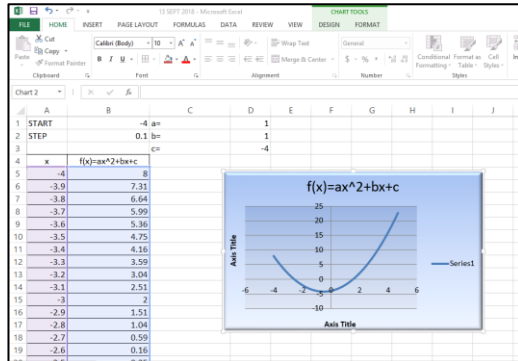
Gambar 2.26 Menemukan Nilai Fungsi x (4) (Sumber: Penulis)

- a. Untuk menemukan nilai $f(x)$, dapat digunakan rumus $f(x)=ax^2+bx+c$ sebagai berikut kemudian *drag* sel ke bawah.

Clipboard			
Font			
SUM			=D\$1*A5^2+D\$2*A5+D\$3
A	B	C	D
1 START	-4	a=	1
2 STEP	0.1	b=	1
3		c=	-4
4 x	f(x)=ax^2+bx+c		
5	-4	=D\$1*A5^2+D\$2*A5+D\$3	
6	-3.9	7.31	
7	-3.8	6.64	
8	-3.7	5.99	
9	-3.6	5.36	
10	-3.5	4.75	
11	-3.4	4.16	
12	-3.3	3.59	
13	-3.2	3.04	
14	-3.1	2.51	
15	-3	2	
16	-2.9	1.51	
17	-2.8	1.04	

Gambar 2.27 Menemukan Nilai Fungsi x (5) (Sumber: Penulis)

- b. Untuk membuat grafik dari persamaan $f(x)= ax^2+bx+c$, dapat dilakukan dengan cara klik *Insert, chart*, klik *OK*.

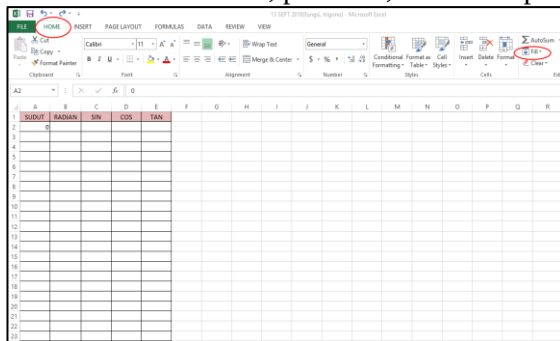


Gambar 2.28 Menemukan Nilai Fungsi x (6) (Sumber: Penulis)

2.1.6 Mencari Nilai Sin, Cos, Tan dan Menyajikannya dalam Bentuk Grafik

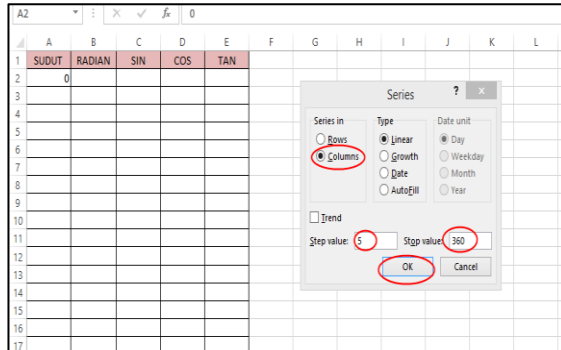
Langkah-langkah yang dapat dilakukan adalah sebagai berikut.

1. Buat tabel sudut kelipatan 5 mulai dari 0 sampai dengan 360 ke dalam sel. Untuk membuat tabel tersebut tidak perlu mengetikkan satu per satu, dapat digunakan fungsi *autofill* untuk mengisi sel dengan data yang mengikuti kelipatan atau pola tertentu.
2. Buat tabel seperti gambar berikut. Letakkan kursor pada sel A2 yang berisi angka nol. Pada *Tab Home*, pilih *Fill*, kemudian pilih *Series*



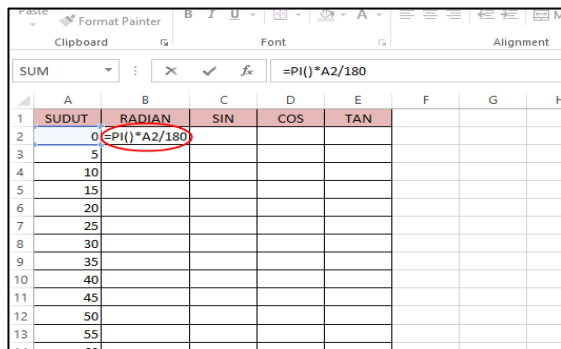
Gambar 2.29 Mencari nilai Sin, Cos, dan Tan (1) (Sumber: Penulis)

3. Pilih *Column*, lalu isi *step value* dengan 5 yang artinya bilangan yang dimunculkan adalah kelipatan 5, serta isi *stop value* dengan 360 yang artinya bilangan akan berhenti pada nilai ke 360. Sehingga bilangan yang dimunculkan adalah bilangan kelipatan 5 dari 0 sampai 360. Kemudian klik *OK*.



Gambar 2.30 Mencari nilai Sin, Cos, dan Tan (2) (Sumber: Penulis)

4. Pada *Microsoft Excel*, ketika ingin mencari nilai trigonometri dari sebuah sudut harus diubah terlebih dahulu ke bentuk radian. Maka untuk mengubah nilai sudut menjadi radian dapat dilakukan dengan mengisi rumus sebagai berikut.



Gambar 2.31 Mencari nilai Sin, Cos, dan Tan (3) (Sumber: Penulis)

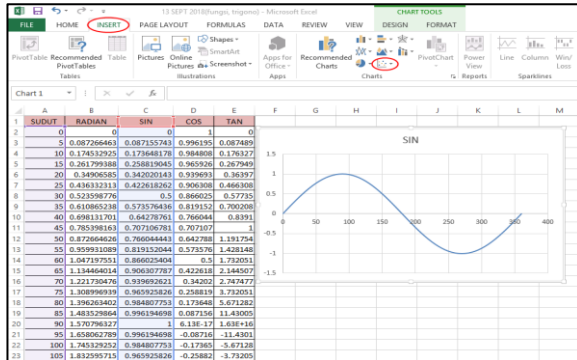
5. Untuk mencari nilai Sin, Cos, dan Tan dapat digunakan rumus sebagai berikut.

Format Painter						Format Painter					
Clipboard		Font				Clipboard		Font			
C2		=sin(B2)				D2		=cos(B2)			
A	B	C	D	E	A	B	C	D	E		
SUDUT	RADIAN	SIN	COS	TAN	SUDUT	RADIAN	SIN	COS	TAN		
0	0	=sin(B2)			0	0	=cos(B2)				
5	0.087266463				5	0.087266463	0.087155743				
10	0.174532925				10	0.174532925	0.173648178				
15	0.261799388				15	0.261799388	0.258819045				
20	0.34906585				20	0.34906585	0.342020143				
25	0.436332313				25	0.436332313	0.422618262				
30	0.523598776				30	0.523598776	0.5				
35	0.610865238				35	0.610865238	0.573576436				
40	0.698131701				40	0.698131701	0.64278761				
45	0.785398163				45	0.785398163	0.707106781				
50	0.872664626				50	0.872664626	0.766044443				
55	0.959931089				55	0.959931089	0.819152044				
60	1.047197551				60	1.047197551	0.866025404				
65	1.134464014				65	1.134464014	0.906307787				
70	1.221730476				70	1.221730476	0.939692621				
75	1.308996939				75	1.308996939	0.965925826				
80	1.396263402				80	1.396263402	0.984807753				
85	1.483529864				85	1.483529864	0.996194698				
90	1.570796327				90	1.570796327	1				
95	1.658062789				95	1.658062789	0.996194698				
100	1.745329252				100	1.745329252	0.984807753				
105	1.832595715				105	1.832595715	0.965925826				

Format Painter					
Clipboard		Font			
SUM		=TAN(B2)			
A	B	C	D	E	
SUDUT	RADIAN	SIN	COS	TAN	
0	0			=TAN(B2)	
5	0.087266463	0.087155743	0.996195		
10	0.174532925	0.173648178	0.984808		
15	0.261799388	0.258819045	0.965926		
20	0.34906585	0.342020143	0.939693		
25	0.436332313	0.422618262	0.906308		
30	0.523598776	0.5	0.866025		
35	0.610865238	0.573576436	0.819152		
40	0.698131701	0.64278761	0.766044		
45	0.785398163	0.707106781	0.707107		
50	0.872664626	0.766044443	0.642788		
55	0.959931089	0.819152044	0.573576		
60	1.047197551	0.866025404	0.5		
65	1.134464014	0.906307787	0.422618		
70	1.221730476	0.939692621	0.34202		
75	1.308996939	0.965925826	0.258819		
80	1.396263402	0.984807753	0.173648		
85	1.483529864	0.996194698	0.087156		
90	1.570796327	1	6.13E-17		
95	1.658062789	0.996194698	-0.08716		
100	1.745329252	0.984807753	-0.17365		
105	1.832595715	0.965925826	-0.25882		

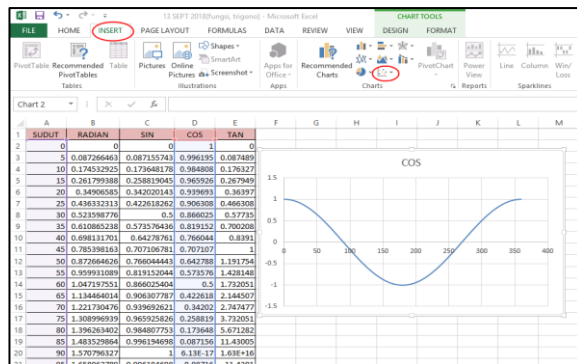
Gambar 2.32 Mencari nilai Sin, Cos, dan Tan (4) (Sumber: Penulis)

- Untuk membuat grafik dari sin dapat dilakukan dengan cara blok kolom sudut dan sin, kemudian pilih *Insert*, pada *group chart* pilih *scatter*.



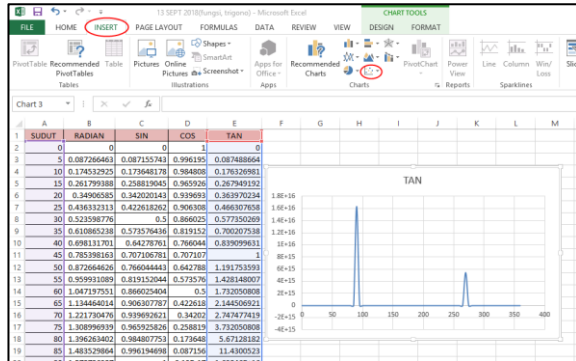
Gambar 2.33 Mencari nilai Sin, Cos, dan Tan (5) (Sumber: Penulis)

7. Untuk membuat grafik dari cos dapat dilakukan dengan cara blok kolom sudut dan cos, kemudian pilih *Insert*, pada *group chart* pilih *scatter*.



Gambar 2.34 Mencari nilai Sin, Cos, dan Tan (6) (Sumber: Penulis)

8. Untuk membuat grafik dari tan dapat dilakukan dengan cara blok kolom sudut dan tan, kemudian pilih *Insert*, pada *group chart* pilih *scatter*.



Gambar 2.35 Mencari nilai Sin, Cos, dan Tan (7) (Sumber: Penulis)

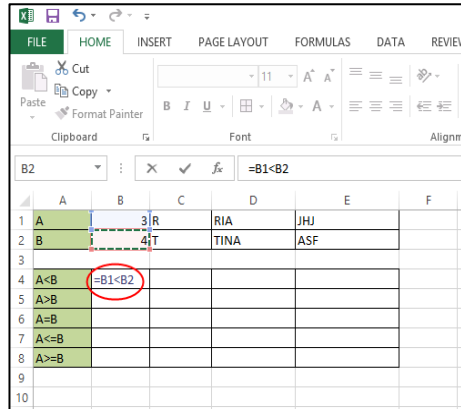
2.1.7 Menggunakan Fungsi AND dan OR

Fungsi AND dan OR adalah fungsi yang sama-sama membandingkan dua atau lebih kondisi sehingga menghasilkan nilai *TRUE* atau *FALSE*. Contohnya diberikan tabel seperti berikut.

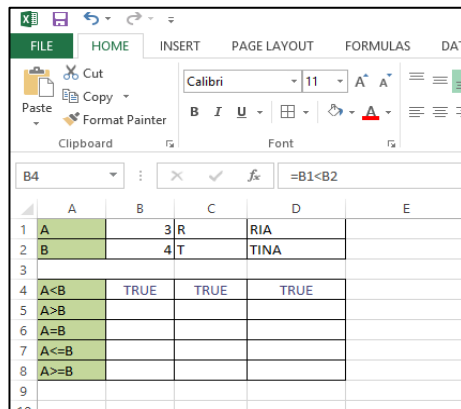
	A	B	C	D	E
1	A	3	R	RIA	
2	B	4	T	TINA	
3					
4	A<B	TRUE	TRUE	TRUE	
5	A>B				
6	A=B				
7	A<=B				
8	A>=B				
9					
10					

Gambar 2.36 Menggunakan fungsi AND dan OR (1) (Sumber: Penulis)

Untuk mengetahui $A < B$ menghasilkan nilai *TRUE* atau *FALSE*, dapat digunakan rumus sebagai berikut.



Gambar 2.37 Menggunakan fungsi AND dan OR (2) (Sumber: Penulis)



Gambar 2.38 Menggunakan fungsi AND dan OR (3) (Sumber: Penulis)

Berdasarkan gambar di atas, dapat dilihat bahwa pada kondisi A<B menghasilkan nilai *TRUE*, untuk mengisi sel yang lain dapat digunakan rumus-rumus yang sesuai dengan kondisi pada kolom A5, A6, A7, dan A8 sehingga dapat diperoleh hasil sebagai berikut.

	A	B	C	D	E
1	3	4	R	RIA	
2	4	T	T	TINA	
3					
4	A<B	TRUE	TRUE	TRUE	
5	A>B	FALSE	FALSE	FALSE	
6	A=B	FALSE	FALSE	FALSE	
7	A<=B	TRUE	TRUE	TRUE	
8	A>=B	FALSE	FALSE	FALSE	
9					

Gambar 2.39 Menggunakan fungsi AND dan OR (4) (Sumber: Penulis)

Berikutnya, untuk menggunakan fungsi AND dan OR diberikan contoh sebagai berikut.

	A	B	C	D	E	F	G	H	I	J
1	P	Q	R							
2	TRUE	TRUE	TRUE							
3	TRUE	FALSE	TRUE							
4	FALSE	TRUE	TRUE							
5	FALSE	FALSE	TRUE							
6	TRUE	TRUE	FALSE							
7	TRUE	FALSE	FALSE							
8	FALSE	TRUE	FALSE							
9	FALSE	FALSE	FALSE							
10										
11	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)	
12										
13										
14										
15										
16										
17										
18										
19										
20										

Gambar 2.40 Menggunakan fungsi AND dan OR (5) (Sumber: Penulis)

Untuk mengisi sel-sel yang kosong, dapat digunakan rumus yang sesuai dengan kondisi. Pada kondisi P AND Q dapat digunakan rumus sebagai berikut.

Clipboard		Font		Alignment		Number		Style	
SUM									
A	B	C	D	E	F	G	H	I	
P	Q	R							
TRUE	TRUE	TRUE							
TRUE	FALSE	TRUE							
FALSE	TRUE	TRUE							
FALSE	FALSE	TRUE							
TRUE	TRUE	FALSE							
TRUE	FALSE	FALSE							
FALSE	TRUE	FALSE							
FALSE	FALSE	FALSE							
P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)	
=AND(A2,B2)									

Gambar 2.41 Menggunakan fungsi AND dan OR (6) (Sumber: Penulis)

Untuk mengisi kolom P OR Q dapat digunakan rumus sebagai berikut.

Clipboard		Font		Alignment		Number		Style	
SUM									
A	B	C	D	E	F	G	H	I	
P	Q	R							
TRUE	TRUE	TRUE							
TRUE	FALSE	TRUE							
FALSE	TRUE	TRUE							
FALSE	FALSE	TRUE							
TRUE	TRUE	FALSE							
TRUE	FALSE	FALSE							
FALSE	TRUE	FALSE							
FALSE	FALSE	FALSE							
P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)	
TRUE	=OR(A2,B2)								
FALSE									
FALSE									
TRUE									
FALSE									

Gambar 2.42 Menggunakan fungsi AND dan OR (7) (Sumber: Penulis)

Untuk mengisi kolom Not P dapat digunakan rumus sebagai berikut.

Clipboard		Font		Alignment		Number		Style	
SUM									
A	B	C	D	E	F	G	H	I	
P	Q	R							
TRUE	TRUE	TRUE							
TRUE	FALSE	TRUE							
FALSE	TRUE	TRUE							
FALSE	FALSE	TRUE							
TRUE	TRUE	FALSE							
TRUE	FALSE	FALSE							
FALSE	TRUE	FALSE							
FALSE	FALSE	FALSE							
P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)	
TRUE	TRUE	=NOT(A2)							
FALSE	TRUE								
FALSE	TRUE								
FALSE	FALSE								

Gambar 2.43 Menggunakan fungsi AND dan OR (8) (Sumber: Penulis)

Untuk mengisi kolom Not Q dapat digunakan rumus sebagai berikut.

SUM									
	A	B	C	D	E	F	G	H	I
1	P	Q	R						
2	TRUE	TRUE	TRUE						
3	TRUE	FALSE	TRUE						
4	FALSE	TRUE	TRUE						
5	FALSE	FALSE	TRUE						
6	TRUE	TRUE	FALSE						
7	TRUE	FALSE	FALSE						
8	FALSE	TRUE	FALSE						
9	FALSE	FALSE	FALSE						
10									
11	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)
12	TRUE	TRUE	FALSE	TRUE					
13	FALSE	TRUE	FALSE						
14	FALSE	TRUE	TRUE						

Gambar 2.44 Menggunakan fungsi AND dan OR (9) (Sumber: Penulis)

Untuk mengisi kolom P AND Q AND R yaitu gabungan dari kolom P, Q, dan R dapat digunakan rumus sebagai berikut.

E12									
	A	B	C	D	E	F	G	H	I
1	P	Q	R						
2	TRUE	TRUE	TRUE						
3	TRUE	FALSE	TRUE						
4	FALSE	TRUE	TRUE						
5	FALSE	FALSE	TRUE						
6	TRUE	TRUE	FALSE						
7	TRUE	FALSE	FALSE						
8	FALSE	TRUE	FALSE						
9	FALSE	FALSE	FALSE						
10									
11	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)
12	TRUE	TRUE	FALSE	FALSE	=AND(A2,B2,C2)				
13	FALSE	TRUE	FALSE	TRUE					
14	FALSE	TRUE	TRUE	FALSE					
15	FALSE	FALSE	TRUE	TRUE					
16	TRUE	TRUE	FALSE	FALSE					
17	FALSE	TRUE	FALSE	TRUE					
18	FALSE	TRUE	TRUE	FALSE					
19	FALSE	FALSE	TRUE	TRUE					
20									

Gambar 2.45 Menggunakan fungsi AND dan OR (10) (Sumber: Penulis)

Untuk mengisi kolom P OR Q OR R dapat digunakan rumus sebagai berikut.

SUM									
	A	B	C	D	E	F	G	H	I
1	P	Q	R						
2	TRUE	TRUE	TRUE						
3	TRUE	FALSE	TRUE						
4	FALSE	TRUE	TRUE						
5	FALSE	FALSE	TRUE						
6	TRUE	TRUE	FALSE						
7	TRUE	FALSE	FALSE						
8	FALSE	TRUE	FALSE						
9	FALSE	FALSE	FALSE						
10									
11	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)
12	TRUE	TRUE	FALSE	FALSE	FALSE	=OR(A2,B2,C2)			
13	FALSE	TRUE	FALSE	TRUE	FALSE				
14	FALSE	TRUE	TRUE	FALSE	FALSE				
15	FALSE	FALSE	TRUE	TRUE	FALSE				
16	TRUE	TRUE	FALSE	FALSE	FALSE				
17	FALSE	TRUE	FALSE	TRUE	FALSE				
18	FALSE	TRUE	TRUE	FALSE	FALSE				
19	FALSE	FALSE	TRUE	TRUE	FALSE				
20									

Gambar 2.46 Menggunakan fungsi AND dan OR (11) (Sumber: Penulis)

Untuk mengisi $P \text{ AND } (Q \text{ OR } R)$ dapat digunakan rumus sebagai berikut.

SUM										
	A	B	C	D	E	F	G	H	I	
1	P	Q	R							
2	TRUE	TRUE	TRUE							
3	TRUE	FALSE	TRUE							
4	FALSE	TRUE	TRUE							
5	FALSE	FALSE	TRUE							
6	TRUE	TRUE	FALSE							
7	TRUE	FALSE	FALSE							
8	FALSE	TRUE	FALSE							
9	FALSE	FALSE	FALSE							
10										
11	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)	
12	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	=AND(A2,OR(B2,C2))		
13	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE				
14	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE				
15	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE				
16	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE				
17	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE				
18	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE				
19	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE				
20										

Gambar 2.47 Menggunakan fungsi AND dan OR (12) (Sumber: Penulis)

Untuk mengisi $(P \text{ AND } Q) \text{ OR } R$ dapat digunakan rumus sebagai berikut.

SUM									
	A	B	C	D	E	F	G	H	I
1	P	Q	R						
2	TRUE	TRUE	TRUE						
3	TRUE	FALSE	TRUE						
4	FALSE	TRUE	TRUE						
5	FALSE	FALSE	TRUE						
6	TRUE	TRUE	FALSE						
7	TRUE	FALSE	FALSE						
8	FALSE	TRUE	FALSE						
9	FALSE	FALSE	FALSE						
10	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P) AND (Q OR R)
12	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	=OR(AND(A2,B2),C2)	
13	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE		
14	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE		
15	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE		
16	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE		
17	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE		
18	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE		
19	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE		

Gambar 2.48 Menggunakan fungsi AND dan OR (13) (Sumber: Penulis)

Untuk mengisi kolom $(\text{NOT } P) \text{ AND } (Q \text{ OR } R)$ dapat digunakan rumus sebagai berikut.

	A	B	C	D	E	F	G	H	I
1	P	Q	R						
2	TRUE	TRUE	TRUE						
3	TRUE	FALSE	TRUE						
4	FALSE	TRUE	TRUE						
5	FALSE	FALSE	TRUE						
6	TRUE	TRUE	FALSE						
7	TRUE	FALSE	FALSE						
8	FALSE	TRUE	FALSE						
9	FALSE	FALSE	FALSE						
10	P AND Q	P OR Q	NOT P	NOT Q	P AND Q AND R	P OR Q OR R	P AND (Q OR R)	(P AND Q) OR R	(NOT P AND Q) OR R
12	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	=AND(NOT(A2),OR(B2,C2))
13	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	
14	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	
15	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	
16	TRUE	TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	
17	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	FALSE	FALSE	
18	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	
19	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	

Gambar 2.49 Menggunakan fungsi AND dan OR (14) (Sumber: Penulis)

2.1.8 Menggunakan Fungsi RANDBETWEEN dan IF

Fungsi **RANDBETWEEN** berfungsi untuk membangkitkan data acak dengan menentukan nilai batas bawah dan batas atas suatu bilangan. Contoh penerapan fungsi **IF** dan **RANDBETWEEN** adalah pada tabel sebagai berikut.

	A	B	C	D	E	F	G	H	I
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=60	KETERANGAN	KETERANGAN
2	1								
3	2								
4	3								
5	4								
6	5								
7	6								
8	7								
9	8								
10	9								
11	10								
12	11								
13	12								
14	13								

Gambar 2.50 Menggunakan fungsi RANDBETWEEN dan IF (1)
(Sumber: Penulis)

Langkah-langkah untuk mengisi kolom sel yang kosong adalah sebagai berikut.

1. Untuk mengisi nilai menggunakan fungsi **RANDBETWEEN** dengan ketentuan nilai 10 sampai 100, kemudian *drag* ke bawah.

	A	B	C	D	E	F	G	H	I
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=80	KETERANGAN	KETERANGAN
2	1	=RANDBETWEEN(10,100)							
3	2								
4	3								
5	4								
6	5								
7	6								
8	7								
9	8								
10	9								
11	10								
12	11								

Gambar 2.51 Menggunakan fungsi RANDBETWEEN dan IF (2)
(Sumber: Penulis)

- Untuk mengisi sel C2 pada tabel tersebut, jika nilai pada kolom B bernilai lebih besar sama dengan 60, maka akan muncul nilai *TRUE*, jika sebaliknya, maka akan muncul nilai *FALSE*.

	A	B	C	D	E	F	G	H	I
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=80	KETERANGAN	KETERANGAN
2	1	28	=B2>=60						
3	2	44							
4	3	29							
5	4	32							
6	5	75							
7	6	30							
8	7	35							
9	8	89							
10	9	56							
11	10	29							
12	11	98							
13	12	11							
14	13	49							

Gambar 2.52 Menggunakan fungsi RANDBETWEEN dan IF (3)
(Sumber: Penulis)

- Nilai yang dimunculkan oleh fungsi *RANDBETWEEN* akan selalu berubah-ubah. Supaya nilai tidak berubah-ubah langkah yang dilakukan adalah dengan meng-copy pada sel lain, kemudian pilih *paste value*.
- Kolom keterangan bisa didapat melalui dua cara, yakni melalui kolom B dan kolom C. Untuk mengisi keterangan pada kolom D dan E dapat digunakan rumus fungsi *IF*. Untuk kolom D, jika sel C2 bernilai *TRUE*, maka keterangan yang dihasilkan yaitu lulus, jika sebaliknya maka keterangan tidak lulus. Rumus yang digunakan adalah sebagai berikut.

Clipboard		Font	Alignment	Number	Styles	Cells			
SUM	X	✓	f _x	=IF(C2=TRUE,"LULUS","TIDAK LULUS")					
	A	B	C	D	E	F	G	H	I
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=80	KETERANGAN	KETERANGAN
2	1	62	TRUE	=IF(C2=TRUE,"LULUS","TIDAK LULUS")					
3	2	60	TRUE						
4	3	47	FALSE						
5	4	53	FALSE						
6	5	68	TRUE						
7	6	64	TRUE						
8	7	68	TRUE						
9	8	81	TRUE						
10	9	66	TRUE						

Gambar 2.53 Menggunakan fungsi RANDBETWEEN dan IF (4)
(Sumber: Penulis)

- Untuk mengisi keterangan pada kolom E, menggunakan acuan pada kolom B, jika nilai dari sel B2 lebih besar sama dengan 60, maka keterangan yang dihasilkan adalah lulus, jika sebaliknya maka tidak lulus.

Clipboard		Font	Alignment	Number	Formatting	Font Styles	Cells		
SUM		X	✓	f _x	=IF(B2>=60,"LULUS","TIDAK LULUS")				
	A	B	C	D	E	F	G	H	I
	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=80	KETERANGAN	KETERANGAN
1	1	20	FALSE	TIDAK LULUS	=IF(B2>=60,"LULUS","TIDAK LULUS")				
2	2	45	FALSE	TIDAK LULUS					
3	3	71	TRUE	LULUS					
4	4	61	TRUE	LULUS					
5	5	68	TRUE	LULUS					
6	6	12	FALSE	TIDAK LULUS					
7	7	55	FALSE	TIDAK LULUS					
8	8	20	FALSE	TIDAK LULUS					
9	9	61	TRUE	LULUS					
10	10	81	TRUE	LULUS					
11	11	54	FALSE	TIDAK LULUS					
12	12	35	FALSE	TIDAK LULUS					

Gambar 2.54 Menggunakan fungsi RANDBETWEEN dan IF (5)
(Sumber: Penulis)

- Untuk mengisi kolom F sama dengan kolom C.
- Untuk mengisi kolom G, dapat digunakan rumus sebagai berikut.

Clipboard Font Alignment Number Styles Cells									
SUM X ✓ f_x =B2<=80									
1	A	B	C	D	E	F	G	H	I
	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=60	<=80	KETERANGAN	KETERANGAN
2	1	15	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	=B2<=80		
3	2	22	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE			
4	3	32	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE			
5	4	92	TRUE	LULUS	LULUS	TRUE			
6	5	32	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE			
7	6	82	TRUE	LULUS	LULUS	TRUE			
8	7	52	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE			
9	8	69	TRUE	LULUS	LULUS	TRUE			
10	9	62	TRUE	LULUS	LULUS	TRUE			

Gambar 2.55 Menggunakan fungsi RANDBETWEEN dan IF (6)
(Sumber: Penulis)

- Untuk mengisi keterangan, jika nilai lebih besar dari 60 dan kurang dari 80 maka keterangan yang dihasilkan adalah baik, selain itu keterangan yang dihasilkan adalah kurang baik. Keterangan dapat diisi melalui dua

cara, yaitu pada kolom H menggunakan acuan pada kolom F dan G, sedangkan keterangan pada kolom I menggunakan acuan pada kolom B. untuk mengisi kolom H dapat digunakan rumus sebagai berikut.

Clipboard Font Alignment Number Styles Cells								
SUM X ✓ fx =IF(AND(F2,G2),"BAIK","KURANG BAIK")								
	A	B	C	D	E	F	G	H
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=80	<=80	KETERANGAN
2	1	57	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	=IF(AND(F2,G2),"BAIK","KURANG BAIK")
3	2	45	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	
4	3	96	TRUE	LULUS	LULUS	TRUE	FALSE	
5	4	54	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	
6	5	51	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	
7	6	10	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	
8	7	67	TRUE	LULUS	LULUS	TRUE	TRUE	

Gambar 2.56 Menggunakan fungsi RANDBETWEEN dan IF (7)
(Sumber: Penulis)

9. Untuk mengisi kolom I dapat digunakan rumus sebagai berikut.

Clipboard Font Alignment Number Styles Cells								
SUM X ✓ fx =IF(AND(B2>=60,B2<=80),"BAIK","KURANG BAIK")								
	A	B	C	D	E	F	G	H
1	NO	NILAI	NILAI>=60	KETERANGAN	KETERANGAN	>=80	<=80	KETERANGAN
2	1	91	TRUE	LULUS	LULUS	TRUE	FALSE	=IF(AND(B2>=60,B2<=80),"BAIK","KURANG BAIK")
3	2	38	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	KURANG BAIK
4	3	10	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	KURANG BAIK
5	4	26	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	KURANG BAIK
6	5	100	TRUE	LULUS	LULUS	TRUE	FALSE	KURANG BAIK
7	6	87	TRUE	LULUS	LULUS	TRUE	FALSE	KURANG BAIK
8	7	51	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	KURANG BAIK
9	8	22	FALSE	TIDAK LULUS	TIDAK LULUS	FALSE	TRUE	KURANG BAIK

Gambar 2.57 Menggunakan fungsi RANDBETWEEN dan IF (8)
(Sumber: Penulis)

Contoh penerapan dari fungsi IF juga dapat diterapkan pada kriteria nilai IPK seperti pada tabel berikut.

F6 X ✓ fx				
	A	B	C	D
1	Nama	IPK	Predikat	
2	Adi	2.6		
3	Budi	3.2		
4	Arsa	2.8		
5	Nina	3.8		
6	Rara	2.4		
7	Ani	4		
8	Kiki	2.2		
9	Ida	3.5		
10	Andi	2.7		
11	Riri	3.7		
12	Dede	2.5		
13				

Gambar 2.58 Menggunakan fungsi RANDBETWEEN dan IF (9)
(Sumber: Penulis)

Contohnya untuk mengisi predikat, kriteria-kriteria yang harus dipenuhi adalah sebagai berikut.

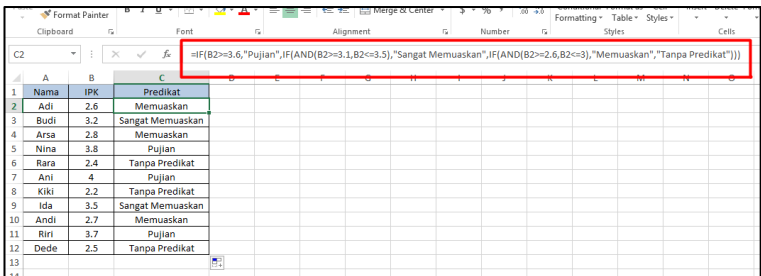
Predikat untuk $IPK \geq 3,6$ adalah pujian,

Predikat untuk $3,1 \leq IPK \leq 3,5$ adalah sangat memuaskan,

Predikat untuk $2,6 \leq IPK \leq 3$ adalah memuaskan,

Predikat untuk $IPK < 3,6$ adalah tanpa predikat.

Berikut merupakan rumus yang dapat digunakan untuk mengisi kolom predikat.

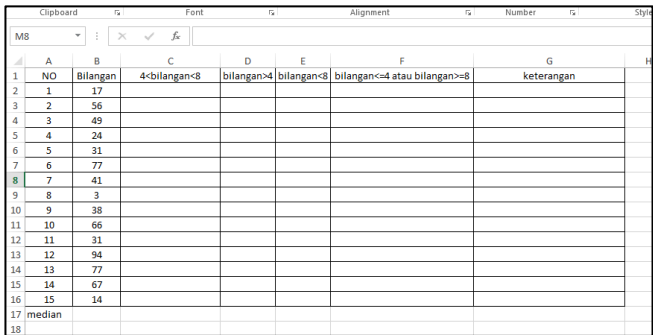


A	B	C
Nama	IPK	Predikat
Adi	2.6	Memuaskan
Budi	3.2	Sangat Memuaskan
Arsa	2.8	Memuaskan
Nina	3.8	Pujian
Rara	2.4	Tanpa Predikat
Ani	4	Pujian
Kiki	2.2	Tanpa Predikat
Ida	3.5	Sangat Memuaskan
Andi	2.7	Memuaskan
Riri	3.7	Pujian
Dede	2.5	Tanpa Predikat

Gambar 2.59 Menggunakan fungsi *RANDBETWEEN* dan *IF* (10)
(Sumber: Penulis)

Berikut merupakan contoh lain penerapan fungsi *IF*.

1. Langkah pertama yaitu memunculkan angka acak dari 1 sampai 100 dengan menggunakan fungsi *RANDBETWEEN* kemudian *copy* dan *paste value* agar nilai tidak berubah ubah seperti pada gambar berikut.



A	B	C	D	E	F	G
NO	Bilangan	<bilangan<8	bilangan>4	bilangan<8	bilangan<=4 atau >=8	keterangan
1	17					
2	56					
3	49					
4	24					
5	31					
6	77					
7	41					
8	3					
9	38					
10	66					
11	31					
12	94					
13	77					
14	67					
15	14					
median						

Gambar 2.60 Menggunakan fungsi *RANDBETWEEN* dan *IF* (11)
(Sumber: Penulis)

2. Mengisi median dengan menggunakan rumus sebagai berikut.

Format Painter							
Clipboard		Font		Alignment		Number	
SUM							
	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan>4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17					
3	2	56					
4	3	49					
5	4	24					
6	5	31					
7	6	77					
8	7	41					
9	8	3					
10	9	38					
11	10	66					
12	11	31					
13	12	94					
14	13	77					
15	14	67					
16	15	14					
17	median	=MEDIAN(B2:B16)					
18							

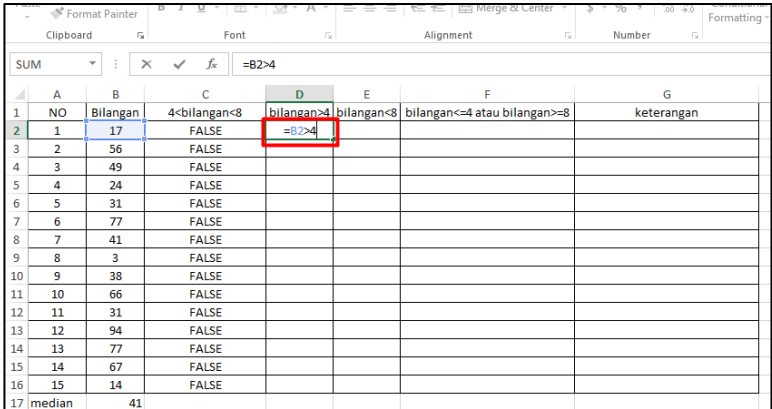
Gambar 2.61 Menggunakan fungsi *RANDBETWEEN* dan *IF* (12)
(Sumber: Penulis)

3. Mengisi kolom C dengan menggunakan rumus sebagai berikut.

Format Painter							
Clipboard		Font		Alignment		Number	
SUM							
	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan>4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17	=AND(B2>4,B2<8)				
3	2	56					
4	3	49					
5	4	24					
6	5	31					
7	6	77					
8	7	41					
9	8	3					
10	9	38					
11	10	66					
12	11	31					
13	12	94					
14	13	77					
15	14	67					
16	15	14					
17	median	41					
18							

Gambar 2.62 Menggunakan fungsi *RANDBETWEEN* dan *IF* (13)
(Sumber: Penulis)

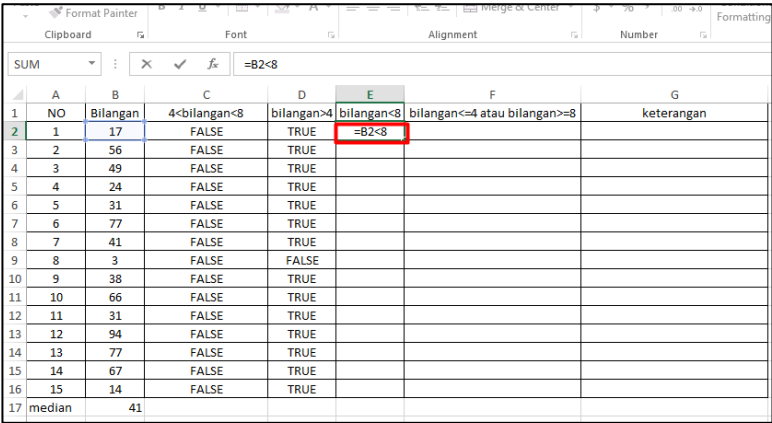
4. Mengisi kolom D dengan menggunakan rumus sebagai berikut.



	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan>4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17	FALSE	=B2>4			
3	2	56	FALSE				
4	3	49	FALSE				
5	4	24	FALSE				
6	5	31	FALSE				
7	6	77	FALSE				
8	7	41	FALSE				
9	8	3	FALSE				
10	9	38	FALSE				
11	10	66	FALSE				
12	11	31	FALSE				
13	12	94	FALSE				
14	13	77	FALSE				
15	14	67	FALSE				
16	15	14	FALSE				
17	median	41					

Gambar 2.63 Menggunakan fungsi *RANDBETWEEN* dan *IF* (14)
(Sumber: Penulis)

5. Mengisi kolom E dengan menggunakan rumus sebagai berikut.



	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan>4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17	FALSE	TRUE	=B2<8		
3	2	56	FALSE	TRUE			
4	3	49	FALSE	TRUE			
5	4	24	FALSE	TRUE			
6	5	31	FALSE	TRUE			
7	6	77	FALSE	TRUE			
8	7	41	FALSE	TRUE			
9	8	3	FALSE	FALSE			
10	9	38	FALSE	TRUE			
11	10	66	FALSE	TRUE			
12	11	31	FALSE	TRUE			
13	12	94	FALSE	TRUE			
14	13	77	FALSE	TRUE			
15	14	67	FALSE	TRUE			
16	15	14	FALSE	TRUE			
17	median	41					

Gambar 2.64 Menggunakan fungsi *RANDBETWEEN* dan *IF* (15)
(Sumber: Penulis)

6. Mengisi kolom F dengan menggunakan rumus sebagai berikut.

	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan<4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17	FALSE	TRUE	FALSE	=OR(B2<=4,B2>=8)	
3	2	56	FALSE	TRUE	FALSE		
4	3	49	FALSE	TRUE	FALSE		
5	4	24	FALSE	TRUE	FALSE		
6	5	31	FALSE	TRUE	FALSE		
7	6	77	FALSE	TRUE	FALSE		
8	7	41	FALSE	TRUE	FALSE		
9	8	3	FALSE	FALSE	TRUE		
10	9	38	FALSE	TRUE	FALSE		
11	10	66	FALSE	TRUE	FALSE		
12	11	31	FALSE	TRUE	FALSE		
13	12	94	FALSE	TRUE	FALSE		
14	13	77	FALSE	TRUE	FALSE		
15	14	67	FALSE	TRUE	FALSE		
16	15	14	FALSE	TRUE	FALSE		
17	median	41					

Gambar 2.65 Menggunakan fungsi *RANDBETWEEN* dan *IF* (16)
(Sumber: Penulis)

7. Megisi keterangan dengan kriteria kurang dari median, sama dengan median, atau lebih dari sama dengan median.

	A	B	C	D	E	F	G
1	NO	Bilangan	4<bilangan<8	bilangan<4	bilangan<8	bilangan<=4 atau bilangan>=8	keterangan
2	1	17	FALSE	TRUE	FALSE	TRUE	Kurang Dari Median
3	2	56	FALSE	TRUE	FALSE	TRUE	
4	3	49	FALSE	TRUE	FALSE	TRUE	
5	4	24	FALSE	TRUE	FALSE	TRUE	
6	5	31	FALSE	TRUE	FALSE	TRUE	
7	6	77	FALSE	TRUE	FALSE	TRUE	
8	7	41	FALSE	TRUE	FALSE	TRUE	
9	8	3	FALSE	FALSE	TRUE	TRUE	
10	9	38	FALSE	TRUE	FALSE	TRUE	
11	10	66	FALSE	TRUE	FALSE	TRUE	
12	11	31	FALSE	TRUE	FALSE	TRUE	
13	12	94	FALSE	TRUE	FALSE	TRUE	
14	13	77	FALSE	TRUE	FALSE	TRUE	
15	14	67	FALSE	TRUE	FALSE	TRUE	
16	15	14	FALSE	TRUE	FALSE	TRUE	
17	median	41					

Gambar 2.66 Menggunakan fungsi *RANDBETWEEN* dan *IF* (17)
(Sumber: Penulis)

2.1.9 Menggunakan Fungsi *LEN*, *LEFT*, *MID*, *RIGHT*

LEN berfungsi untuk menghitung jumlah huruf yang ada pada sel. Penerapan fungsi *LEN* dapat dilihat pada gambar berikut.

	A	B	C	D	E	F	G	H
1	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	66						
3	2	17						
4	3	74						
5	4	53						
6	5	40						
7	6	80						
8	7	38						
9	8	27						
10	9	25						
11	10	95						
12								
13								

Gambar 2.67 Menggunakan fungsi LEN, LEFT, MID, RIGHT (1)
(Sumber: Penulis)

- Langkah pertama yaitu memunculkan angka acak dari 10 sampai 100.
- Mengisi keterangan dengan kriteria sebagai berikut.
Jika bilangan <60 keterangannya adalah kurang
Jika bilangan ≥ 60 dan ≤ 80 keterangannya adalah baik
Jika bilangan ≥ 80 keterangannya adalah istimewa

	A	B	C	D	E	F	G	H
1	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	60	baik					
3	2	36						
4	3	15						
5	4	31						
6	5	21						
7	6	72						
8	7	43						
9	8	91						
10	9	62						
11	10	33						
12								

Gambar 2.68 Menggunakan fungsi LEN, LEFT, MID, RIGHT (2)
(Sumber: Penulis)

- Mengisi len dengan rumus sebagai berikut.

	A	B	C	D	E	F	G	H
1	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	84	istimewa	=LEN(C2)				
3	2	37	kurang					
4	3	44	kurang					
5	4	17	kurang					
6	5	10	kurang					
7	6	69	baik					
8	7	31	kurang					
9	8	61	baik					
10	9	31	kurang					
11	10	16	kurang					
12								

Gambar 2.69 Menggunakan fungsi LEN, LEFT, MID, RIGHT (3)
(Sumber: Penulis)

4. *LEFT* berfungsi untuk mengambil huruf paling kiri dari sebuah sel. Rumus yang digunakan untuk mengambil satu huruf sebelah kiri adalah sebagai berikut.

	A	B	C	D	E	F	G	H
	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	74	baik	4	=LEFT(C2,1)			
3	2	63	baik	4				
4	3	68	baik	4				
5	4	37	kurang	6				
6	5	84	istimewa	8				
7	6	78	baik	4				
8	7	56	kurang	6				
9	8	71	baik	4				
10	9	96	istimewa	8				
11	10	79	baik	4				

Gambar 2.70 Menggunakan fungsi *LEN*, *LEFT*, *MID*, *RIGHT* (4)
(Sumber: Penulis)

5. Fungsi *LEN* tidak hanya digunakan untuk mengambil satu huruf saja. Berikut merupakan rumus yang digunakan untuk mengambil dua huruf dari sebelah kiri.

	A	B	C	D	E	F	G	H
	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	86	istimewa	8	i	=LEFT(C2,2)		
3	2	41	kurang	6	k			
4	3	87	istimewa	8	i			
5	4	18	kurang	6	k			
6	5	55	kurang	6	k			
7	6	87	istimewa	8	i			
8	7	29	kurang	6	k			
9	8	82	istimewa	8	i			
10	9	60	baik	4	b			
11	10	95	istimewa	8	i			

Gambar 2.71 Menggunakan fungsi *LEN*, *LEFT*, *MID*, *RIGHT* (5)
(Sumber: Penulis)

6. *RIGHT* berfungsi untuk mengambil huruf dari sebelah kanan dari sebuah sel. Mengisi kolom *right* dapat dilakukan dengan rumus sebagai berikut.

	A	B	C	D	E	F	G	H
	no	Bilangan	Keterangan	len	left	left 2 huruf	right	mid
2	1	47	kurang	6	k	ku	=RIGHT(C2,1)	
3	2	48	kurang	6	k	ku		
4	3	26	kurang	6	k	ku		
5	4	50	kurang	6	k	ku		
6	5	27	kurang	6	k	ku		
7	6	98	istimewa	8	i	is		
8	7	97	istimewa	8	i	is		
9	8	15	kurang	6	k	ku		
10	9	91	istimewa	8	i	is		
11	10	95	istimewa	8	i	is		

Gambar 2.72 Menggunakan fungsi *LEN*, *LEFT*, *MID*, *RIGHT* (6)
(Sumber: Penulis)

7. *MID* berfungsi untuk mengambil huruf bagian tengah dari sebuah sel. Berikut adalah contoh rumus yang digunakan untuk mengambil dua huruf yang dimulai pada huruf kedua dari sebuah sel.

	A	B	C	D	E	F	G	H
1	no	Bilangan	keterangan	len	left	left 2 huruf	right	mid
2	1	11	kurang	6	k	ku	g	=MID(C2,2,2)
3	2	89	istimewa	8	i	is	a	st
4	3	15	kurang	6	k	ku	g	ur
5	4	61	baik	4	b	ba	k	ai
6	5	44	kurang	6	k	ku	g	ur
7	6	97	istimewa	8	i	is	a	st
8	7	22	kurang	6	k	ku	g	ur
9	8	10	kurang	6	k	ku	g	ur
10	9	42	kurang	6	k	ku	g	ur
11	10	44	kurang	6	k	ku	g	ur

Gambar 2.73 Menggunakan fungsi LEN, LEFT, MID, RIGHT (7)
(Sumber: Penulis)

2.1.10 Perkalian dan Invers Matriks

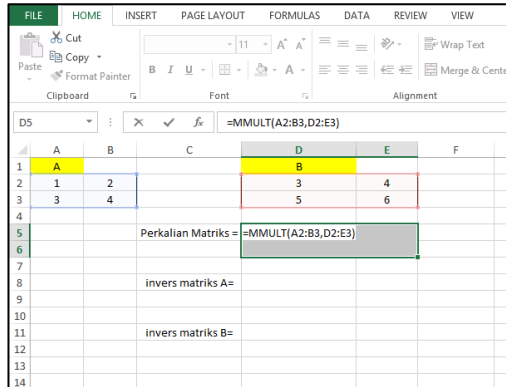
Langkah-langkah yang harus dilakukan adalah sebagai berikut.

1. Membuat matriks A dan matriks B seperti pada gambar berikut.

	A	B	C	D	E	F
1	A			B		
2	1	2		3	4	
3	3	4		5	6	
4						
5			Perkalian Matriks =			
6						
7						
8			invers matriks A=			
9						
10						
11			invers matriks B=			
12						
13						

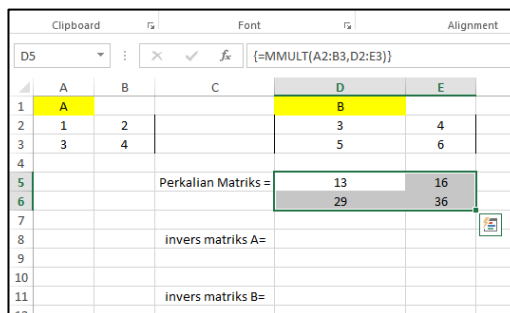
Gambar 2.74 Perkalian dan Invers Matriks (1) (Sumber: Penulis)

2. Mengalikan matriks A dan matriks B dengan menggunakan fungsi MMULT, caranya adalah dengan memblok 4 sel sebagai daerah hasil, kemudian ketikkan rumus, pilih area matriks yang akan dikalikan.



Gambar 2.75 Perkalian dan Invers Matriks (2) (Sumber: Penulis)

3. Kemudian secara bersamaan tekan tombol **CTRL**, **SHIFT** dan **ENTER** maka akan muncul hasil perkalian dari matriks A dan B.



Gambar 2.76 Perkalian dan Invers Matriks (3) (Sumber: Penulis)

4. Untuk invers matriks, dapat digunakan fungsi **MINVERSE**. Langkah yang dilakukan yang pertama yaitu dengan memblok 4 sel sebagai daerah hasil, kemudian ketikkan rumus, pilih area matriks A yang akan diinvers.

Clipboard		Font		Alignment	
D8				=MINVERSE(A2:B3)	
A	B	C	D	E	
1	A		B		
2	1	2	3	4	
3	3	4	5	6	
4					
5		Perkalian Matriks =	13	16	
6			29	36	
7					
8		invers matriks A=	=MINVERSE(A2:B3)		
9					
10					
11		invers matriks B=			
12					

Gambar 2.77 Perkalian dan Invers Matriks (4) (Sumber: Penulis)

5. Kemudian secara bersamaan tekan CTRL, SHIFT, dan ENTER.

Clipboard		Font		Alignment	
D8				{=MINVERSE(A2:B3)}	
A	B	C	D	E	
1	A		B		
2	1	2	3	4	
3	3	4	5	6	
4					
5		Perkalian Matriks =	13	16	
6			29	36	
7					
8		invers matriks A=	-2	1	
9			1.5	-0.5	
10					
11		invers matriks B=			
12					

Gambar 2.78 Perkalian dan Invers Matriks (5) (Sumber: Penulis)

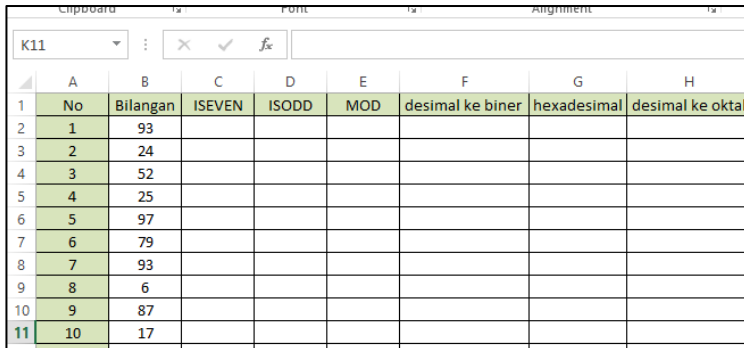
6. Dengan rumus dan langkah-langkah yang sama, invers matriks B dapat diperoleh sebagai berikut.

Clipboard		Font		Alignment	
D11				{=MINVERSE(D2:E3)}	
A	B	C	D	E	
1	A		B		
2	1	2	3	4	
3	3	4	5	6	
4					
5		Perkalian Matriks =	13	16	
6			29	36	
7					
8		invers matriks A=	-2	1	
9			1.5	-0.5	
10					
11		invers matriks B=	-3	2	
12			2.5	-1.5	
13					

Gambar 2.79 Perkalian dan Invers Matriks (6) (Sumber: Penulis)

2.1.11 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal

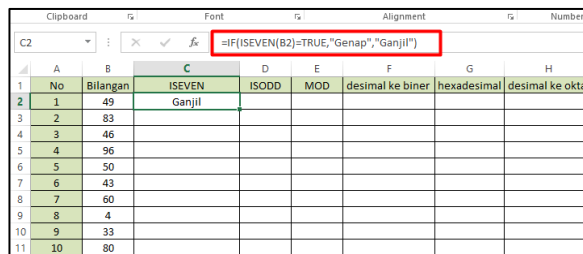
1. ISEVEN, ISODD, dan MOD berguna untuk mengetahui apakah sel yang diuji berisi angka genap atau ganjil. Langkah yang pertama yaitu memunculkan angka acak dari 1 sampai 100 sebanyak yang diinginkan.



	A	B	C	D	E	F	G	H
1	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	93						
3	2	24						
4	3	52						
5	4	25						
6	5	97						
7	6	79						
8	7	93						
9	8	6						
10	9	87						
11	10	17						

Gambar 2.80 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (1) (Sumber: Penulis)

2. Kemudian untuk menguji apakah sel B1 ganjil atau genap dapat digunakan fungsi ISEVEN sebagai berikut.



	A	B	C	D	E	F	G	H
1	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	49	Ganjil					
3	2	83						
4	3	46						
5	4	96						
6	5	50						
7	6	43						
8	7	60						
9	8	4						
10	9	33						
11	10	80						

Gambar 2.81 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (2) (Sumber: Penulis)

3. Selain itu dapat juga menggunakan fungsi ISODD sebagai berikut.

	A	B	C	D	E	F	G	H
	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	20	Genap	Genap				
3	2	19	Ganjil					
4	3	61	Ganjil					
5	4	7	Ganjil					
6	5	15	Ganjil					
7	6	17	Ganjil					
8	7	37	Ganjil					
9	8	91	Ganjil					
10	9	53	Ganjil					
11	10	69	Ganjil					

Gambar 2.82 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (3) (Sumber: Penulis)

4. Untuk menentukan angka mana yang merupakan bilangan ganjil atau genap dapat dilakukan dengan cara mengetahui nilai sisa setelah angka tersebut dibagi dengan angka 2. Angka ganjil akan menghasilkan nilai 1 dan angka genap akan menghasilkan nilai 0. Fungsi *excel* yang digunakan adalah MOD.

	A	B	C	D	E	F	G	H
	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	80	Genap	Ganjil	genap			
3	2	93	Ganjil	Genap				
4	3	63	Ganjil	Genap				
5	4	6	Genap	Ganjil				
6	5	89	Ganjil	Genap				
7	6	12	Genap	Ganjil				
8	7	69	Ganjil	Genap				
9	8	50	Genap	Ganjil				
10	9	19	Ganjil	Genap				
11	10	50	Genap	Ganjil				

Gambar 2.83 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (4) (Sumber: Penulis)

5. Mengkonversi bilangan desimal ke biner dapat digunakan fungsi DEC2BIN sebagai berikut.

	A	B	C	D	E	F	G	H
	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	22	Genap	Ganjil	genap	=DEC2BIN(B2)		
3	2	68	Genap	Ganjil	genap			
4	3	58	Genap	Ganjil	genap			
5	4	71	Ganjil	Genap	ganjil			
6	5	8	Genap	Ganjil	genap			
7	6	86	Genap	Ganjil	genap			
8	7	75	Ganjil	Genap	ganjil			
9	8	7	Ganjil	Genap	ganjil			
10	9	94	Genap	Ganjil	genap			
11	10	59	Ganjil	Genap	ganjil			

Gambar 2.84 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (5) (Sumber: Penulis)

6. Mengkonversi bilangan desimal ke hexadesimal dapat digunakan fungsi DEC2HEX sebagai berikut.

	A	B	C	D	E	F	G	H
1	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	51	Ganjil	Genap	ganjil	110011	=DEC2HEX(B2)	
3	2	63	Ganjil	Genap	ganjil	111111		
4	3	59	Ganjil	Genap	ganjil	111011		
5	4	39	Ganjil	Genap	ganjil	100111		
6	5	62	Genap	Ganjil	genap	111110		
7	6	61	Ganjil	Genap	ganjil	111101		
8	7	68	Genap	Ganjil	genap	1000100		
9	8	68	Genap	Ganjil	genap	1000100		
10	9	92	Genap	Ganjil	genap	1011100		
11	10	18	Genap	Ganjil	genap	10010		

Gambar 2.85 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (6) (Sumber: Penulis)

7. Mengkonversi bilangan desimal ke oktal dapat digunakan fungsi DEC2OCT sebagai berikut.

	A	B	C	D	E	F	G	H
1	No	Bilangan	ISEVEN	ISODD	MOD	desimal ke biner	hexadesimal	desimal ke oktal
2	1	54	Genap	Ganjil	genap	110110	36	=DEC2OCT(B2)
3	2	98	Genap	Ganjil	genap	1100010	62	
4	3	18	Genap	Ganjil	genap	10010	12	
5	4	42	Genap	Ganjil	genap	101010	2A	
6	5	67	Ganjil	Genap	ganjil	1000011	43	
7	6	42	Genap	Ganjil	genap	101010	2A	

Gambar 2.86 Menggunakan Fungsi ISEVEN, ISODD, MOD, Mengkonversi Bilangan Desimal Ke Biner, Hexadesimal, dan Oktal (7) (Sumber: Penulis)

2.1.12 Permutasi Dan Kombinasi

1. Permutasi

Langkah-langkah yang dapat dilakukan untuk menghitung permutasi n dari n (${}_nP_n$) atau $n!$ adalah sebagai berikut.

1. Buat tabel seperti berikut.

	A	B	C	D
1		permut(n,n)		
2	n	n!		
3	0			
4	1			
5	2			
6	3			
7	4			
8	5			
9	6			

Gambar 2.87 Permutasi (1) (Sumber: Penulis)

2. Kemudian ketikkan rumus seperti pada gambar berikut.

	A	B
1		permut(n,n)
2	n	n!
3	0	=PERMUT(A3,A3)
4	1	
5	2	
6	3	
7	4	
8	5	
9	6	

Gambar 2.88 Permutasi (2) (Sumber: Penulis)

Sedangkan untuk permutasi k dari n (${}_nP_k$) dapat dituliskan dengan rumus PERMUT(n,k) dengan langkah-langkah sebagai berikut.

1. Buat tabel seperti gambar berikut.

	A	B	C	D	E	F	G	H
1								
2	n	0	1	2	3	4	5	6
3	0							
4	1							
5	2							
6	3							
7	4							
8	5							
9	6							

Gambar 2.89 Permutasi k dari n (${}_nP_k$) (1) (Sumber: Penulis)

2. Pada sel permutasi 0 dari 0 dapat digunakan rumus sebagai berikut.

	A	B	C	D	E	F	G	H
1								
2	n	0	1	2	3	4	5	6
3	0	=PERMUT(\$A\$3,B2)						
4	1							
5	2							
6	3							
7	4							
8	5							
9	6							

Gambar 2.90 Permutasi k dari n (${}_nP_k$) (2) (Sumber: Penulis)

Permutasi 0 dari 1 dapat digunakan rumus sebagai berikut, kemudian *drag* ke kanan.

Clipboard		Font		Alignment				
SUM	:				=PERMUT(\$A\$4,B2)			
	A	B	C	D	E	F	G	H
1	n	k						
2	0	0	1	2	3	4	5	6
3	0	1						
4	1	=PERMUT(\$A\$4,B2)						
5	2							
6	3							
7	4							
8	5							
9	6							
10								
11								

Gambar 2.91 Permutasi k dari n (nP_k) (3) (Sumber: Penulis)

3. Ulangi rumus tersebut seterusnya sampai permutasi 0 dari 6.

2	n	0	1	2	3	4	5	6
3	0	1						
4	1	1	1					
5	2	1	2	2				
6	3	1	3	6	6			
7	4	1	4	12	24	24		
8	5	1	5	20	60	120	120	
9	6	1	6	30	120	360	720	720

Gambar 2.92 Permutasi k dari n (nP_k) (4) (Sumber: Penulis)

2. Kombinasi

Langkah-langkah yang dapat dilakukan untuk menghitung kombinasi adalah sebagai berikut.

1. Buat tabel seperti gambar berikut.

	A	B	C	D	E	F	G	H
1	n	kombinasi						
2		0	1	2	3	4	5	6
3	0							
4	1							
5	2							
6	3							
7	4							
8	5							
9	6							

Gambar 2. 93 Kombinasi (1) (Sumber: Penulis)

2. Mengisi sel kombinasi 0 dari 0 dengan menggunakan fungsi COMBIN(n,k), kemudian *drag* ke kanan.

The screenshot shows the Excel formula bar with the formula `=COMBIN($A$3,B2)`. Below it is a table with columns A through H. Row 1 is a header with 'n' in A1 and 'kombinasi' in B1. Row 2 has values 0, 1, 2, 3, 4, 5, 6 in columns A through G. Row 3 has the formula `=COMBIN($A$3,B2)` in cell B3, which is highlighted with a red border. The table continues with rows 4 through 9, each with a value in column A and empty cells in columns B through G.

A	B	C	D	E	F	G	H
n	kombinasi						
0	0	1	2	3	4	5	6
0	=COMBIN(\$A\$3,B2)						
1							
2							
3							
4							
5							
6							

Gambar 2. 94 Kombinasi (2) (Sumber: Penulis)

Lakukan dengan cara yang sama untuk mengisi sel selanjutnya.

The screenshot shows the completed table from the previous image. The formula bar is empty. The table now contains numerical values in cells B3 through B9, representing the combinations of 3 items taken 0 through 6 at a time.

A	B	C	D	E	F	G	H
n	kombinasi						
0	0	1	2	3	4	5	6
0	1						
1	1	1					
2	1	2	1				
3	1	3	3	1			
4	1	4	6	4	1		
5	1	5	10	10	5	1	
6	1	6	15	20	15	6	1

Gambar 2. 95 Kombinasi (3) (Sumber: Penulis)

2.1.13 Menggunakan Fungsi *VLOOKUP* dan *HLOOKUP*

1. *VLOOKUP*

Latihan fungsi *VLOOKUP* dapat dilakukan dengan langkah-langkah sebagai berikut.

- Buat tabel seperti gambar berikut.

The screenshot shows a table with columns A through M. The table contains data for various subjects, including Bahasa Inggris, Desain Eksperimen, Ekonomi Teknik, Komputasi Statistika, Manajemen Operasi, Matematika 1, Metode Regresi, Pendidikan Agama, Pengantar Ilmu Komputer, Pengantar Ilmu Ekonomi, Pengantar Metode Statistika, Teknik Pengukuran Kerja, and Teknik Sampling. The table is organized into sections for 'Nilai huruf', 'Nilai angka', and 'IPS'. The 'IPS' section includes a formula `=SUM(SKS)` and a table for 'Nama', 'NRP', 'IPS', and 'Ket.'.

A	B	C	D	E	F	G	H	I	J	K	L	M
Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka	IPS =					
Bahasa Inggris	3	A	ANI		A	4						
Desain Eksperimen	3	B	BUDI		AB	3,5						
Ekonomi Teknik	2	A	RIZKA		B	3						
Komputasi Statistika	3	B	AYU		BC	2,5						
Manajemen Operasi	3	B	RONI		C	2						
Matematika 1	3	A	ANI		D	1						
Metode Regresi	3	B	BUDI		E	0						
Pendidikan Agama	2	A	RIZKA									
Pengantar Ilmu Komputer	3	B	AYU		No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS
Pengantar Ilmu Ekonomi	3	B	RONI		1							
Pengantar Metode Statistika	4	A	RIZKA		2							
Teknik Pengukuran Kerja	3	B	AYU		3							
Teknik Sampling	3	B	RONI		4							
					5							
					6							
					7							

Gambar 2. 96 Latihan fungsi *VLOOKUP* (1) (Sumber: Penulis)

- Mengisi nama dan NRP sesuai yang diinginkan, kemudian mengisi sel IPS dengan menggunakan rumus yang tersedia yang diperoleh dari tabel di bawahnya.

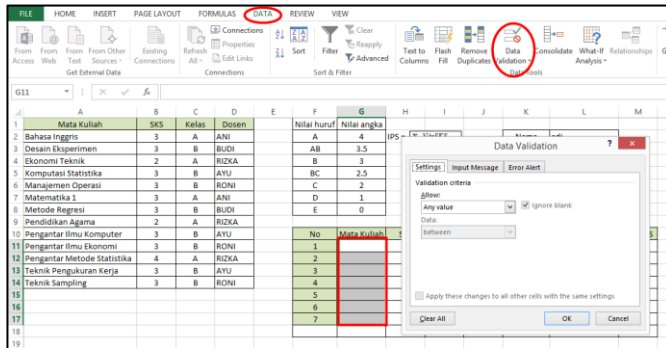
The screenshot shows an Excel spreadsheet with two tables. The first table (A1:D14) lists courses, SKS, classes, and lecturers. The second table (F1:H5) shows student data with columns for 'Nilai huruf', 'Nilai angka', and 'IPS'. A formula bar shows the formula for cell H5: `=VLOOKUP(SK5, $A$1:$D$14, 2, FALSE)`. A third table (I1:M7) shows a list of student IDs (NRP) and their corresponding IPS values.

Mata Kuliah	SKS	Kelas	Dosen
Bahasa Inggris	3	A	ANI
Desain Eksperimen	3	B	BUDI
Ekonomi Teknik	2	A	RIZKA
Komputasi Statistika	3	B	AYU
Manajemen Operasi	3	B	RONI
Matematika 1	3	A	ANI
Metode Regresi	3	B	BUDI
Pendidikan Agama	2	A	RIZKA
Pengantar Ilmu Komputer	3	B	AYU
Pengantar Ilmu Ekonomi	3	B	RONI
Pengantar Metode Statistika	4	A	RIZKA
Teknik Pengukuran Kerja	3	B	AYU
Teknik Sampling	3	B	RONI

No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka	IPS
1							
2							
3							
4							
5							
6							
7							

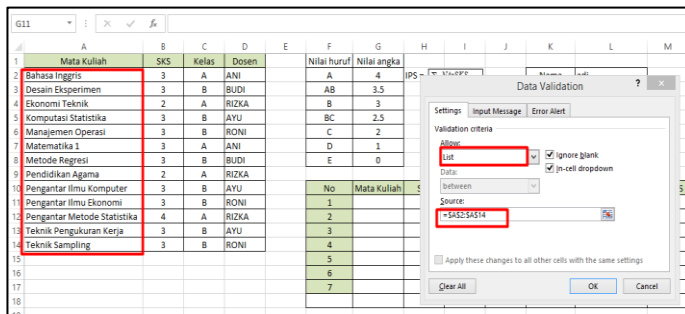
Gambar 2. 97 Latihan fungsi VLOOKUP(2) (Sumber: Penulis)

- c. Mengisi kolom mata kuliah menggunakan *Data Validation*, dengan cara pilih *Data Validation* pada tab *Data*.



Gambar 2. 98 Latihan fungsi VLOOKUP(3) (Sumber: Penulis)

- d. Mengatur *Allow* menjadi *List*, mengatur *source* dengan cara blok pada kolom mata kuliah, kemudian klik *ok*.



Gambar 2. 99 Latihan fungsi VLOOKUP(4) (Sumber: Penulis)

- e. Kemudian mengisi sel mata kuliah dengan cara klik segitiga di samping sel untuk memilih mata kuliah yang diinginkan.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka							
2	Bahasa Inggris	3	A	ANI		A	4	$IPS = \frac{\sum \text{Nilai SKS}}{\sum \text{SKS}}$			Nama			
3	Desain Eksperimen	3	B	BUDI		AB	3,5				NRP			
4	Ekonomi Teknik	2	A	RIZKA		B	3				IPS			
5	Komputasi Statistika	3	B	AYU		BC	2,5				Ket.			
6	Manajemen Operasi	3	B	RONI										
7	Matematika 1	3	A	ANI		D	1							
8	Metode Regresi	3	B	BUDI		E	0							
9	Pendidikan Agama	2	A	RIZKA										
10	Pengantar Ilmu Komputer	3	B	AYU										
11	Pengantar Ilmu Ekonomi	3	B	RONI		No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS	
12	Pengantar Metode Statistika	4	A	RIZKA		1	Bahasa Inggris							
13	Teknik Pengukuran Kerja	3	B	AYU		2	Komputasi Statistika							
14	Teknik Sampling	3	B	RONI		3	Matematika 1							
15						4	Metode Regresi							
16						5	Teknik Sampling							
17						6	Manajemen Operasi							
18						7	Desain Eksperimen							
19														

Gambar 2.100 Latihan fungsi VLOOKUP (5) (Sumber: Penulis)

- f. Mengisi kolom SKS menggunakan fungsi VLOOKUP, dengan rumus sebagai berikut. (G11 menunjukkan mata kuliah yang akan dimunculkan SKS nya, A2 sampai D14 menunjukkan daerah tabel yang dipilih, dan 2 merupakan nomor kolom SKS yaitu pada kolom kedua).

SUM														
=VLOOKUP(G11,\$A\$2:\$D\$14,2)														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka							
2	Bahasa Inggris	3	A	ANI		A	4	$IPS = \frac{\sum \text{Nilai SKS}}{\sum \text{SKS}}$			Nama			
3	Desain Eksperimen	3	B	BUDI		AB	3,5				NRP			
4	Ekonomi Teknik	2	A	RIZKA		B	3				IPS			
5	Komputasi Statistika	3	B	AYU		BC	2,5				Ket.			
6	Manajemen Operasi	3	B	RONI										
7	Matematika 1	3	A	ANI		D	1							
8	Metode Regresi	3	B	BUDI		E	0							
9	Pendidikan Agama	2	A	RIZKA										
10	Pengantar Ilmu Komputer	3	B	AYU										
11	Pengantar Ilmu Ekonomi	3	B	RONI		No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS	
12	Pengantar Metode Statistika	4	A	RIZKA		1	Bahasa Inggris	=VLOOKUP(G11,\$A\$2:\$D\$14,2)						
13	Teknik Pengukuran Kerja	3	B	AYU		2	Komputasi Statistika							
14	Teknik Sampling	3	B	RONI		3	Matematika 1							
						4	Metode Regresi							

Gambar 2.101 Latihan fungsi VLOOKUP (6) (Sumber: Penulis)

- g. Mengisi kolom kelas dengan menggunakan cara yang sama seperti kolom SKS, hanya mengubah nomor kolom yaitu kolom ketiga.

SUM										
	A	B	C	D	E	F	G	H	I	J
1	Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka			
2	Bahasa Inggris	3	A	ANI		A	4	IPS = $\frac{\sum N \times S_{SKS}}{\sum SKS}$		Nama
3	Desain Eksperimen	3	B	BUDI		AB	3.5			NRP
4	Ekonomi Teknik	2	A	RIZKA		B	3			IPS
5	Komputasi Statistika	3	B	AYU		BC	2.5			Ket.
6	Manajemen Operasi	3	B	RONI		C	2			
7	Matematika 1	3	A	ANI		D	1			
8	Metode Regresi	3	B	BUDI		E	0			
9	Pendidikan Agama	2	A	RIZKA						
10	Pengantar Ilmu Komputer	3	B	AYU		No	Mata Kuliah	SKS	Kelas	Dosen
11	Pengantar Ilmu Ekonomi	3	B	RONI		1	Bahasa Inggris	3		
12	Pengantar Metode Statistika	4	A	RIZKA		2	Komputasi Statistika	3		
13	Teknik Pengukuran Kerja	3	B	AYU		3	Matematika 1	3		
14	Teknik Sampling	3	B	RONI		4	Metode Regresi	3		
15						5	Teknik Sampling	3		

Gambar 2.102 Latihan fungsi *VLOOKUP* (7)

- h. Mengisi kolom dosen dengan menggunakan cara yang sama seperti kolom SKS, hanya mengubah nomor kolom yaitu kolom keempat.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka					
2	Bahasa Inggris	3	A	ANI		A	4	IPS = $\frac{\sum N \times S_{SKS}}{\sum SKS}$			Nama	
3	Desain Eksperimen	3	B	BUDI		AB	3.5				NRP	
4	Ekonomi Teknik	2	A	RIZKA		B	3				IPS	
5	Komputasi Statistika	3	B	AYU		BC	2.5				Ket.	
6	Manajemen Operasi	3	B	RONI		C	2					
7	Matematika 1	3	A	ANI		D	1					
8	Metode Regresi	3	B	BUDI		E	0					
9	Pendidikan Agama	2	A	RIZKA								
10	Pengantar Ilmu Komputer	3	B	AYU		No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka
11	Pengantar Ilmu Ekonomi	3	B	RONI		1	Bahasa Inggris	3	A			
12	Pengantar Metode Statistika	4	A	RIZKA		2	Komputasi Statistika	3	B			
13	Teknik Pengukuran Kerja	3	B	AYU		3	Matematika 1	3	A			
14	Teknik Sampling	3	B	RONI		4	Metode Regresi	3	B			
15						5	Teknik Sampling	3	A			

Gambar 2.103 Latihan fungsi *VLOOKUP* (8) (Sumber: Penulis)

- i. Nilai huruf diisi sesuai yang diinginkan. Kemudian mengisi nilai angka dengan menggunakan fungsi *VLOOKUP* dengan menggunakan tabel F2 sampai G8 kolom ke 2.

	F	G	H	I	J	K	L	M
	Nilai huruf	Nilai angka						
	A	4	IPS = $\frac{\sum N \times S_{SKS}}{\sum SKS}$			Nama		
	AB	3.5				NRP		
	B	3				IPS		
	BC	2.5				Ket.		
	C	2						
	D	1						
	E	0						
	No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka (NA)	NA*SKS
1	Bahasa Inggris	3	A	ANI	A			
2	Komputasi Statistika	3	B	AYU	AB			
3	Matematika 1	3	A	ANI	A			
4	Metode Regresi	3	B	BUDI	C			
5	Teknik Sampling	3	B	RONI	B			
6	Manajemen Operasi	3	B	RONI	BC			
7	Desain Eksperimen	3	B	BUDI	B			

Gambar 2.104 Latihan fungsi *VLOOKUP* (9) (Sumber: Penulis)

- j. Mengisi NA*SKS dengan mengalikan nilai angka dengan SKS dengan cara sebagai berikut.

G	H	I	J	K	L	M
Nilai angka	$IPS = \frac{\sum NdxSKS}{\sum SKS}$			Nama		
4				NRP		
3.5				IPS		
3				Ket.		
2.5						
2						
1						
0						
Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS
Bahasa Inggris	3	A	ANI	A	4	=11*I11
Komputasi Statistika	3	B	AYU	AB	3.5	
Matematika 1	3	A	ANI	A	4	

Gambar 2.105 Latihan fungsi *VLOOKUP* (10) (Sumber: Penulis)

- k. Menjumlahkan seluruh SKS, Nilai Angka, dan NA*SKS dengan menggunakan fungsi *SUM*.

SUM									
	A	B	C	D	E	F	G	H	I
	Mata Kuliah	SKS	Kelas	Dosen		Nilai huruf	Nilai angka	IPS	
1	Bahasa Inggris	3	A	ANI		A	4	$IPS = \frac{\sum NdxSKS}{\sum SKS}$	
2	Desain Eksperimen	3	B	BUDI		AB	3.5		
3	Desain Eksperimen	3	B	BUDI		AB	3		
4	Komputasi Statistika	3	B	AYU		BC	2.5		
5	Manajemen Operasi	3	B	RONI		C	2		
6	Metode Regresi	3	B	BUDI		D	1		
7	Matematika 1	3	A	ANI		E	0		
8	Metode Regresi	3	B	BUDI					
9	Pendidikan Agama	2	A	RIZKA					
10	Pengantar Ilmu Komputer	3	B	AYU					
11	Pengantar Ilmu Ekonomi	3	B	RONI					
12	Pengantar Metode Statistika	4	A	RIZKA					
13	Teknik Pengukuran Kerja	3	B	AYU					
14	Teknik Sampling	3	B	RONI					
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38									
39									
40									
41									
42									
43									
44									
45									
46									
47									
48									
49									
50									
51									
52									
53									
54									
55									
56									
57									
58									
59									
60									
61									
62									
63									
64									
65									
66									
67									
68									
69									
70									
71									
72									
73									
74									
75									
76									
77									
78									
79									
80									
81									
82									
83									
84									
85									
86									
87									
88									
89									
90									
91									
92									
93									
94									
95									
96									
97									
98									
99									
100									

Gambar 2.106 Latihan fungsi *VLOOKUP* (11) (Sumber: Penulis)

- l. Mengisi sel IPS dengan menggunakan rumus sebagai berikut.

F	G	H	I	J	K	L	M
Nilai huruf	Nilai angka	$IPS = \frac{\sum NdxSKS}{\sum SKS}$			Nama		
A	4				NRP		
AB	3.5				IPS	=M18/H18	
B	3				Ket.		
BC	2.5						
C	2						
D	1						
E	0						
No	Mata Kuliah	SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS
1	Bahasa Inggris	3	A	ANI	A	4	12
2	Komputasi Statistika	3	B	AYU	AB	3.5	10.5
3	Matematika 1	3	A	ANI	A	4	12
4	Metode Regresi	3	B	BUDI	C	2	6
5	Teknik Sampling	3	B	RONI	B	3	9
6	Manajemen Operasi	3	B	RONI	BC	2.5	7.5
7	Desain Eksperimen	3	B	BUDI	B	3	9
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							
32							
33							
34							
35							
36							
37							
38							
39							
40							
41							
42							
43							
44							
45							
46							
47							
48							
49							
50							
51							
52							
53							
54							
55							
56							
57							
58							
59							
60							
61							
62							
63							
64							
65							
66							
67							
68							
69							
70							
71							
72							
73							
74							
75							
76							
77							
78							
79							
80							
81							
82							
83							
84							
85							
86							
87							
88							
89							
90							
91							
92							
93							
94							
95							
96							
97							
98							
99							
100							

Gambar 2.107 Latihan fungsi *VLOOKUP* (12) (Sumber: Penulis)

- m. Membulatkan nilai IPS menjadi desimal 2 angka di belakang koma.

H	I	J	K	L	M
$S = \frac{\sum N_i \times SKS_i}{\sum SKS_i}$			Nama		
			NRP		
			IPS	=ROUND(M18/H18,2)	
			Ket.		
SKS	Kelas	Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS
	A	ANI	A	4	12

Gambar 2.108 Latihan fungsi VLOOKUP (13) (Sumber: Penulis)

- n. Untuk mengisi sel keterangan, dapat digunakan fungsi IF dengan syarat sebagai berikut.

Jika $IPS > 3$ maka keterangannya adalah istimewa.

Jika $2,5 \leq IPS \leq 3$ maka keterangannya adalah baik.

Jika $2 \leq IPS \leq 2,5$ maka keterangannya adalah cukup.

Jika $IPS \leq 2$ maka keterangannya adalah kurang.

J	K	L	M	N	O	P
	Nama					
	NRP					
	IPS	3.14				
	Ket.	=IF(L4>3,"ISTIMEWA",IF(L4<=2,"KURANG",IF(2<L4<=2.5,"CUKUP","BAIK")))				
Dosen	Nilai Huruf	Nilai Angka(NA)	NA*SKS			
ANI	A	4	12			
AYU	AB	3.5	10.5			
ANI	A	4	12			

Gambar 2.109 Latihan fungsi VLOOKUP (14) (Sumber: Penulis)

- o. Setelah diinput maka dapat diperoleh hasil sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Mata Kuliah	SKS	Kelas	Dosen	Nilai huruf	Nilai angka							
2	Bahasa Inggris	3	A	ANI	A	4					Nama		
3	Desain Eksperimen	3	B	BUDI	AB	3.5					NRP		
4	Ekonomi Teknik	2	A	RIZKA	B	3					IPS	3.14	
5	Komputasi Statistika	3	B	AYU	BC	2.5					Ket.	ISTIMEWA	
6	Manajemen Operasi	3	B	RONI	C	2							
7	Matematika 1	3	A	ANI	D	1							
8	Metode Regresi	3	B	BUDI	E	0							
9	Pendidikan Agama	2	A	RIZKA									
10	Pengantar Ilmu Komputer	3	B	AYU									
11	Pengantar Ilmu Ekonomi	3	B	RONI									
12	Pengantar Metode Statistika	4	A	RIZKA									
13	Teknik Pengukuran Kerja	3	B	AYU									
14	Teknik Sampling	3	B	RONI									
15													
16													
17													
18													
19													

Gambar 2.110 Latihan fungsi VLOOKUP (15) (Sumber: Penulis)

2. HLOOKUP

Latihan fungsi *HLOOKUP* dapat dilakukan dengan langkah-langkah sebagai berikut.

- Buat contoh soal seperti gambar berikut. Pembeli akan mendapatkan diskon apabila membeli lebih dari 1 barang.

	A	B	C	D	E	F	G	H	I	J
1	Kode	Nama Kasir		Harga			Diskon			
2	10A	Riri		Buku Tulis	Rp36,000		Buku Tulis	Tas	Pensil	Kertas
3	10B	Andi		Tas	Rp150,000		2%	4%	1%	3%
4	10C	Roy		Pensil	Rp12,000					
5				Kertas	Rp40,000					
6										
7	Nama Barang	Harga	Diskon	Jumlah	Kode	Nama Kasir	Total Harga			
8	Buku Tulis			3	10A					
9	Tas			4	10C					
10	Pensil			5	10A					
11	Kertas			1	10B					

Gambar 2.111 Latihan fungsi *HLOOKUP* (1) (Sumber: Penulis)

- Untuk mengisi kolom harga masih digunakan fungsi *VLOOKUP*. Fungsi *VLOOKUP* digunakan untuk mencari nilai pada tabel secara vertikal sedangkan *HLOOKUP* digunakan untuk mencari nilai pada tabel secara horizontal.

SUM : ✕ ✓ fx =VLOOKUP(A8,\$D\$2:\$E\$5,2,FALSE)										
	A	B	C	D	E	F	G	H	I	J
1	Kode	Nama Kasir		Harga			Diskon			
2	10A	Riri		Buku Tulis	Rp36,000		Buku Tulis	Tas	Pensil	Kertas
3	10B	Andi		Tas	Rp150,000		2%	4%	1%	3%
4	10C	Roy		Pensil	Rp12,000					
5				Kertas	Rp40,000					
6										
7	Nama Barang	Harga	Diskon	Jumlah	Kode	Nama Kasir	Total Harga			
8	Buku Tulis	=VLOOKUP(A8,\$D\$2:\$E\$5,2,FALSE)								
9	Tas			4	10C					
10	Pensil			5	10A					
11	Kertas			1	10B					

Gambar 2.112 Latihan fungsi *HLOOKUP* (2) (Sumber: Penulis)

- Untuk mengisi kolom diskon, dapat digunakan fungsi *HLOOKUP* karena data dalam tabel berbentuk horizontal dengan rumus sebagai berikut.

- f. Total harga adalah harga barang dikurangi diskon dan dikali jumlah barang yang dibeli.

SUM : ✕ ✓ f _x =(B8-C8)*D8										
	A	B	C	D	E	F	G	H	I	J
1	Kode	Nama Kasir		Harga			Diskon			
2	10A	Riri		Buku Tulis	Rp36,000		Buku Tulis	Tas	Pensil	Kertas
3	10B	Andi		Tas	Rp150,000		2%	4%	1%	3%
4	10C	Roy		Pensil	Rp12,000					
5				Kertas	Rp40,000					
6										
7	Nama Barang	Harga	Diskon	Jumlah	Kode	Nama Kasir	Total Harga			
8	Buku Tulis	36000	720	3	10A	Riri	=(B8-C8)*D8			
9	Tas	150000	6000	4	10C	Roy				
10	Pensil	12000	120	5	10A	Riri				
11	Kertas	40000	0	1	10B	Andi				
12										

Gambar 2.116 Latihan fungsi *HLOOKUP* (6) (Sumber: Penulis)

- g. Jika semua tabel sudah terisi maka tampilan akan menjadi seperti berikut.

	A	B	C	D	E	F	G	H	I	J
1	Kode	Nama Kasir		Harga			Diskon			
2	10A	Riri		Buku Tulis	Rp36,000		Buku Tulis	Tas	Pensil	Kertas
3	10B	Andi		Tas	Rp150,000		2%	4%	1%	3%
4	10C	Roy		Pensil	Rp12,000					
5				Kertas	Rp40,000					
6										
7	Nama Barang	Harga	Diskon	Jumlah	Kode	Nama Kasir	Total Harga			
8	Buku Tulis	36000	720	3	10A	Riri	105840			
9	Tas	150000	6000	4	10C	Roy	576000			
10	Pensil	12000	120	5	10A	Riri	59400			
11	Kertas	40000	0	1	10B	Andi	40000			
12										

Gambar 2.117 Latihan fungsi *HLOOKUP* (7) (Sumber: Penulis)

2.1.14 Menggunakan fungsi *COUNTIFS* dan *AVERAGEIF*

Latihan fungsi *COUNTIFS* dan *AVERAGEIF* dapat dilakukan dengan contoh soal berikut.

1. Disajikan 100 data seperti pada gambar berikut.

	A	B	C	D	E	F	G	H	I	J
1	No	village	sex	age	mobile	compute	internet	working	education	
2	1	Urban	Female	60	No	No	No	No	Primary	
3	2	Urban	Male	42	Yes	No	No	Yes	Junior	
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary	
5	4	Urban	Male	44	No	No	No	Yes	Primary	
6	5	Urban	Female	38	No	No	No	Yes	Primary	
7	6	Urban	Female	19	Yes	No	Yes	No	Senior	
8	7	Urban	Female	13	Yes	No	Yes	No	Junior	
9	8	Urban	Male	29	Yes	No	No	Yes	Primary	
10	9	Urban	Female	25	Yes	No	No	No	Junior	
11	10	Urban	Male	62	No	No	No	Yes	Primary	
12	11	Urban	Female	52	No	No	No	Yes	Primary	
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior	
14	13	Urban	Male	37	Yes	No	No	Yes	Junior	
15	14	Urban	Female	38	Yes	No	No	No	Junior	
16	15	Urban	Female	73	No	No	No	Yes	No-Education	
17	16	Urban	Male	33	Yes	No	No	Yes	Junior	
18	17	Urban	Female	30	Yes	No	No	Yes	Senior	
19	18	Urban	Female	8	No	No	No	No	Primary	
20	19	Urban	Male	34	No	No	No	Yes	Primary	
21	20	Urban	Female	27	Yes	No	No	No	Junior	
22	21	Urban	Male	34	Yes	No	No	Yes	Primary	
23	22	Urban	Female	32	Yes	No	No	Yes	Primary	
24	23	Urban	Female	11	No	No	No	No	Primary	
25	24	Urban	Male	6	No	No	No	No	Primary	
26	25	Urban	Male	53	No	No	No	Yes	Primary	
27	26	Urban	Female	52	No	No	No	Yes	Primary	
28	27	Urban	Male	26	Yes	No	Yes	Yes	Primary	
29	28	Urban	Female	71	No	No	No	No	Primary	
30	29	Urban	Male	53	Yes	Yes	Yes	Yes	Junior	
31	30	Urban	Female	49	Yes	No	No	Yes	Senior	
32	31	Urban	Male	31	Yes	Yes	Yes	Yes	Higher	
33	32	Urban	Female	24	Yes	Yes	Yes	Yes	Primary	
34	33	Urban	Male	43	Yes	No	No	Yes	Senior	
35	34	Urban	Female	51	Yes	No	No	Yes	Junior	
36	35	Urban	Male	18	Yes	Yes	Yes	No	Senior	
37	36	Urban	Male	97	Yes	No	No	No	Primary	
38	37	Urban	Female	70	Yes	No	No	No	Primary	
39	38	Urban	Male	40	No	No	No	No	Junior	
40	39	Urban	Male	57	Yes	No	No	Yes	Senior	
41	40	Urban	Female	51	Yes	No	No	Yes	Senior	
42	41	Urban	Female	15	Yes	Yes	Yes	No	Junior	
43	42	Urban	Male	12	Yes	Yes	Yes	No	Primary	
44	43	Urban	Male	62	Yes	No	No	Yes	Higher	
45	44	Urban	Female	48	Yes	No	No	No	Junior	
46	45	Urban	Male	15	Yes	Yes	Yes	No	Junior	

47	46	Urban	Male	60	Yes	No	No	Yes	Senior
48	47	Urban	Female	47	Yes	No	No	Yes	Primary
49	48	Urban	Male	15	Yes	No	No	No	Primary
50	49	Urban	Male	28	Yes	No	No	Yes	Senior
51	50	Urban	Female	27	Yes	No	No	Yes	Senior
52	51	Urban	Female	5	Yes	No	No	No	No-Education
53	52	Urban	Male	50	Yes	No	No	Yes	Junior
54	53	Urban	Male	56	Yes	Yes	Yes	Yes	Higher
55	54	Urban	Female	53	Yes	No	Yes	Yes	Senior
56	55	Urban	Male	38	Yes	Yes	Yes	Yes	Senior
57	56	Urban	Female	31	Yes	Yes	Yes	Yes	Higher
58	57	Urban	Male	10	Yes	Yes	Yes	No	Primary
59	58	Urban	Male	16	Yes	No	Yes	No	Senior
60	59	Urban	Male	18	Yes	No	Yes	No	Higher
61	60	Urban	Male	56	Yes	No	No	Yes	Senior
62	61	Urban	Female	43	Yes	No	No	No	Junior
63	62	Urban	Male	31	Yes	Yes	Yes	Yes	Higher
64	63	Urban	Female	28	Yes	Yes	Yes	Yes	Higher
65	64	Urban	Male	57	Yes	No	No	Yes	Primary
66	65	Urban	Female	48	Yes	No	No	Yes	Primary
67	66	Urban	Male	17	Yes	No	Yes	No	Junior
68	67	Urban	Male	34	Yes	Yes	Yes	Yes	Higher
69	68	Urban	Female	33	Yes	Yes	Yes	Yes	Higher
70	69	Urban	Male	61	Yes	No	No	No	Senior
71	70	Urban	Female	56	Yes	No	No	Yes	Primary
72	71	Urban	Female	37	Yes	Yes	Yes	Yes	Higher
73	72	Urban	Female	7	Yes	No	No	No	Primary
74	73	Urban	Female	51	Yes	Yes	Yes	Yes	Primary
75	74	Urban	Female	16	Yes	Yes	Yes	No	Senior
76	75	Urban	Male	37	Yes	No	No	Yes	Senior
77	76	Urban	Female	30	Yes	No	Yes	Yes	Senior
78	77	Urban	Female	7	Yes	No	No	No	Primary
79	78	Urban	Male	54	Yes	Yes	Yes	Yes	Senior
80	79	Urban	Female	57	Yes	Yes	Yes	Yes	Primary
81	80	Urban	Female	28	Yes	Yes	Yes	Yes	Higher
82	81	Urban	Male	84	No	No	No	Yes	No-Education
83	82	Urban	Female	75	No	No	No	No	No-Education
84	83	Urban	Male	46	Yes	No	No	Yes	Senior
85	84	Urban	Female	42	Yes	No	No	Yes	Primary
86	85	Urban	Female	12	Yes	Yes	Yes	No	Primary
87	86	Urban	Female	11	Yes	Yes	Yes	No	Primary
88	87	Urban	Male	51	Yes	No	No	Yes	Senior
89	88	Urban	Female	45	Yes	No	No	Yes	Senior
90	89	Urban	Female	26	Yes	Yes	Yes	Yes	Higher
91	90	Urban	Male	9	Yes	No	No	No	Primary
92	91	Urban	Female	65	Yes	No	No	No	Higher
93	92	Urban	Female	29	Yes	Yes	Yes	Yes	Higher
94	93	Urban	Male	30	Yes	Yes	Yes	Yes	Higher
95	94	Urban	Male	43	Yes	No	No	Yes	Primary
96	95	Urban	Female	43	Yes	No	No	No	Junior
97	96	Urban	Female	71	No	No	No	No	No-Education
98	97	Urban	Male	76	Yes	No	No	No	Primary
99	98	Urban	Female	71	Yes	No	No	No	Primary
100	99	Urban	Male	47	Yes	No	No	Yes	Primary
101	100	Urban	Female	43	Yes	No	No	No	Primary
102									
103									

Gambar 2.118 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (1)
(Sumber: Penulis)

- Membuat tabel jumlah dan rata-rata umur *Female* dan *Male*, serta pengguna *Mobile* atau tidak seperti pada tabel berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	No	village	sex	age	mobile	computer	internet	working	education			jumlah	rata-rata umur	
2	1	Urban	Female	60	No	No	No	No	Primary					
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female			
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male			
5	4	Urban	Male	44	No	No	No	Yes	Primary					
6	5	Urban	Female	38	No	No	No	Yes	Primary					
7	6	Urban	Female	19	Yes	No	Yes	No	Senior					
8	7	Urban	Female	13	Yes	No	Yes	No	Junior					
9	8	Urban	Male	29	Yes	No	No	Yes	Primary					
10	9	Urban	Female	25	Yes	No	No	No	Junior					
11	10	Urban	Male	62	No	No	No	Yes	Primary					
12	11	Urban	Female	52	No	No	No	Yes	Primary					
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior					
14	13	Urban	Male	37	Yes	No	No	Yes	Junior					

Gambar 2.119 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (2)
(Sumber: Penulis)

- Untuk menghitung jumlah *Female* yang ada pada kolom C, maka dapat digunakan fungsi *COUNTIFS* dengan rumus sebagai berikut.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	No	village	sex	age	mobile	computer	internet	working	education			jumlah	rata-rata umur	
2	1	Urban	Female	60	No	No	No	No	Primary					
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female	=COUNTIFS(C2:C101,"female")		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male			
5	4	Urban	Male	44	No	No	No	Yes	Primary					
6	5	Urban	Female	38	No	No	No	Yes	Primary					
7	6	Urban	Female	19	Yes	No	Yes	No	Senior					
8	7	Urban	Female	13	Yes	No	Yes	No	Junior					
9	8	Urban	Male	29	Yes	No	No	Yes	Primary					
10	9	Urban	Female	25	Yes	No	No	No	Junior					
11	10	Urban	Male	62	No	No	No	Yes	Primary					
12	11	Urban	Female	52	No	No	No	Yes	Primary					
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior					
14	13	Urban	Male	37	Yes	No	No	Yes	Junior					
15	14	Urban	Female	38	Yes	No	No	No	Junior					

Gambar 2.120 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (3)
(Sumber: Penulis)

- Menghitung jumlah *Male* dengan menggunakan langkah yang sama seperti sebelumnya.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	No	village	sex	age	mobile	computer	internet	working	education			jumlah	rata-rata umur	
2	1	Urban	Female	60	No	No	No	No	Primary					
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female	52		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male	=COUNTIFS(C2:C101,"male")		
5	4	Urban	Male	44	No	No	No	Yes	Primary					
6	5	Urban	Female	38	No	No	No	Yes	Primary					
7	6	Urban	Female	19	Yes	No	Yes	No	Senior					
8	7	Urban	Female	13	Yes	No	Yes	No	Junior					
9	8	Urban	Male	29	Yes	No	No	Yes	Primary					
10	9	Urban	Female	25	Yes	No	No	No	Junior					
11	10	Urban	Male	62	No	No	No	Yes	Primary					
12	11	Urban	Female	52	No	No	No	Yes	Primary					

Gambar 2.121 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (4)
(Sumber: Penulis)

5. Menghitung rata-rata umur *Female* dengan menggunakan fungsi *AVERAGEIF* dengan rumus sebagai berikut.

SUM													=AVERAGEIF(\$C\$2:\$C\$101,"female",\$D\$2:\$D\$101)			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No	village	sex	age	mobile	computer	internet	working	education							
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur			
3	2	Urban	Male	42	Yes	No	No	Yes	Junior			Female	52	=AVERAGEIF(\$C\$2:\$C\$101,"female",\$D\$2:\$D\$101)		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary			Male	48			
5	4	Urban	Male	44	No	No	No	Yes	Primary							
6	5	Urban	Female	38	No	No	No	Yes	Primary							
7	6	Urban	Female	19	Yes	No	Yes	No	Senior							
8	7	Urban	Female	13	Yes	No	Yes	No	Junior							
9	8	Urban	Male	29	Yes	No	No	Yes	Primary							
10	9	Urban	Female	25	Yes	No	No	No	Junior							
11	10	Urban	Male	62	No	No	No	Yes	Primary							
12	11	Urban	Female	52	No	No	No	Yes	Primary							
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior							

Gambar 2.122 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (5)
(Sumber: Penulis)

6. Menghitung rata-rata umur *Male* menggunakan fungsi *AVERAGEIF* dengan rumus sebagai berikut.

SUM													=AVERAGEIF(\$C\$2:\$C\$101,"male",\$D\$2:\$D\$101)			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No	village	sex	age	mobile	computer	internet	working	education							
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur			
3	2	Urban	Male	42	Yes	No	No	Yes	Junior			Female	52	38.3		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary			Male	48	=AVERAGEIF(\$C\$2:\$C\$101,"male",\$D\$2:\$D\$101)		
5	4	Urban	Male	44	No	No	No	Yes	Primary							
6	5	Urban	Female	38	No	No	No	Yes	Primary							
7	6	Urban	Female	19	Yes	No	Yes	No	Senior							
8	7	Urban	Female	13	Yes	No	Yes	No	Junior							
9	8	Urban	Male	29	Yes	No	No	Yes	Primary							
10	9	Urban	Female	25	Yes	No	No	No	Junior							
11	10	Urban	Male	62	No	No	No	Yes	Primary							
12	11	Urban	Female	52	No	No	No	Yes	Primary							

Gambar 2.123 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (6)
(Sumber: Penulis)

7. Untuk mengisi sel L8, yaitu jumlah orang perempuan yang menggunakan ponsel dapat digunakan rumus sebagai berikut.

SUM													=COUNTIFS(\$C\$2:\$C\$101,"female",\$E\$2:\$E\$101,"yes")			
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	No	village	sex	age	mobile	computer	internet	working	education							
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur			
3	2	Urban	Male	42	Yes	No	No	Yes	Junior			Female	52	38.3		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary			Male	48	39.25		
5	4	Urban	Male	44	No	No	No	Yes	Primary							
6	5	Urban	Female	38	No	No	No	Yes	Primary							
7	6	Urban	Female	19	Yes	No	Yes	No	Senior							
8	7	Urban	Female	13	Yes	No	Yes	No	Junior							
9	8	Urban	Male	29	Yes	No	No	Yes	Primary							
10	9	Urban	Female	25	Yes	No	No	No	Junior							
11	10	Urban	Male	62	No	No	No	Yes	Primary							
12	11	Urban	Female	52	No	No	No	Yes	Primary							

Gambar 2.124 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (7)
(Sumber: Penulis)

8. Untuk mengisi sel L9, yaitu jumlah orang laki-laki yang menggunakan ponsel dapat digunakan rumus sebagai berikut.

SUM : X ✓ ✕ =COUNTIFS(\$C\$2:\$C\$101,"male",\$E\$2:\$E\$101,"yes")															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	No	village	sex	age	mobile	computer	internet	working	education						
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur		
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female	52	38.3		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male	48	39.25		
5	4	Urban	Male	44	No	No	No	Yes	Primary						
6	5	Urban	Female	38	No	No	No	Yes	Primary						
7	6	Urban	Female	19	Yes	No	Yes	No	Senior						
8	7	Urban	Female	13	Yes	No	Yes	No	Junior		Female	42			
9	8	Urban	Male	29	Yes	No	No	Yes	Primary		Male	=COUNTIFS(\$C\$2:\$C\$101,"male",\$E\$2:\$E\$101,"yes")			
10	9	Urban	Female	25	Yes	No	No	No	Junior		Total				
11	10	Urban	Male	62	No	No	No	Yes	Primary						

Gambar 2.125 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (8)
(Sumber: Penulis)

9. Untuk mengisi sel M8, yaitu jumlah orang perempuan yang tidak menggunakan ponsel dapat digunakan rumus sebagai berikut.

SUM : X ✓ ✕ =COUNTIFS(\$C\$2:\$C\$101,"female",\$E\$2:\$E\$101,"no")															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	No	village	sex	age	mobile	computer	internet	working	education						
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur		
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female	52	38.3		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male	48	39.25		
5	4	Urban	Male	44	No	No	No	Yes	Primary						
6	5	Urban	Female	38	No	No	No	Yes	Primary						
7	6	Urban	Female	19	Yes	No	Yes	No	Senior						
8	7	Urban	Female	13	Yes	No	Yes	No	Junior		Female	42	=COUNTIFS(\$C\$2:\$C\$101,"female",\$E\$2:\$E\$101,"no")		
9	8	Urban	Male	29	Yes	No	No	Yes	Primary		Male	41			
10	9	Urban	Female	25	Yes	No	No	No	Junior		Total				
11	10	Urban	Male	62	No	No	No	Yes	Primary						

Gambar 2.126 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (9)
(Sumber: Penulis)

10. Untuk mengisi sel M9, yaitu jumlah orang laki-laki yang tidak menggunakan ponsel dapat digunakan rumus sebagai berikut.

SUM : X ✓ ✕ =COUNTIFS(\$C\$2:\$C\$101,"male",\$E\$2:\$E\$101,"no")															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	No	village	sex	age	mobile	computer	internet	working	education						
2	1	Urban	Female	60	No	No	No	No	Primary			jumlah	rata-rata umur		
3	2	Urban	Male	42	Yes	No	No	Yes	Junior		Female	52	38.3		
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary		Male	48	39.25		
5	4	Urban	Male	44	No	No	No	Yes	Primary						
6	5	Urban	Female	38	No	No	No	Yes	Primary						
7	6	Urban	Female	19	Yes	No	Yes	No	Senior						
8	7	Urban	Female	13	Yes	No	Yes	No	Junior		Female	42	10		
9	8	Urban	Male	29	Yes	No	No	Yes	Primary		Male	41	=COUNTIFS(\$C\$2:\$C\$101,"male",\$E\$2:\$E\$101,"no")		
10	9	Urban	Female	25	Yes	No	No	No	Junior		Total				

Gambar 2.127 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (10)
(Sumber: Penulis)

11. Menghitung total dengan menggunakan fungsi SUM.

	J	K	L	M	N
			jumlah	rata-rata umur	
Female		52		38.3	
Male		48		39.25	

	Sex	yes	no	Total
Female		42	10	
Male		41	7	
Total		=SUM(B8:B9)		

Gambar 2.128 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (11)
(Sumber: Penulis)

12. Setelah semua sel terisi, tampilan akan menjadi seperti berikut.

Lapboard		Fast		Keyboard		Number		Upper		Cells				
P9														
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	No	village	sex	age	mobile	computer	internet	working	education					
2	1	Urban	Female	60	No	No	No	No	Primary					
3	2	Urban	Male	42	No	No	No	Yes	Junior					
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary					
5	4	Urban	Male	44	No	No	No	Yes	Primary					
6	5	Urban	Female	38	No	No	No	Yes	Primary					
7	6	Urban	Female	19	Yes	No	Yes	No	Senior					
8	7	Urban	Female	13	Yes	No	Yes	No	Junior					
9	8	Urban	Male	29	Yes	No	No	Yes	Primary					
10	9	Urban	Female	25	Yes	No	No	No	Junior					
11	10	Urban	Male	62	No	No	No	Yes	Primary					
12	11	Urban	Female	52	No	No	No	Yes	Primary					
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior					
14	13	Urban	Male	37	Yes	No	No	Yes	Junior					

jumla				rata-rata umur	
Female	52			38.3	
Male	48			39.25	

Mobile			
Sex	yes	no	Total
Female	42	10	52
Male	41	7	48
Total	83	17	100

 CC BY-SA

Gambar 2.129 Latihan fungsi *COUNTIFS* dan *AVERAGEIF* (12)
(Sumber: Penulis)

2.1.15 Tabel Frekuensi

Latihan membuat tabel frekuensi dapat dilakukan dengan langkah-langkah sebagai berikut.

1. Memunculkan angka acak dari 1 sampai 100 sebanyak 100 angka dengan menggunakan fungsi `RANDBETWEEN` kemudian *copy* dan *paste value* sehingga angka tidak berubah-ubah, membuat tabel seperti berikut.

[illegible]

Gambar 2.130 Latihan Tabel Frekuensi (1) (Sumber: Penulis)

- Mengisi sel D4 dengan menggunakan rumus banyak kelas(k) yaitu $k=1+3,3*\log n$. Fungsi *ROUND* digunakan untuk membulatkan bilangan.

Clipboard		Font		Alignment		
SUM				=ROUND(1+3.3*LOG10(100),0)		
	A	B	C	D	E	F
1	x		Banyak Kelas (k)	=ROUND(1+3.3*LOG10(100),0)		
2	74		MIN			
3	90		MAX			
4	60		Panjang Kelas(i)			
5	97					
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI

Gambar 2.131 Latihan Tabel Frekuensi (2) (Sumber: Penulis)

- Mengisi nilai minimum dengan menggunakan fungsi *MIN*.

Clipboard		Font		Alignment		
SUM				=MIN(A2:A101)		
	A	B	C	D	E	F
1	x		Banyak Kelas (k)	8		
2	74		MIN	=MIN(A2:A101)		
3	90		MAX			
4	60		Panjang Kelas(i)			
5	97					
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI
7	18					
8	77					

Gambar 2.132 Latihan Tabel Frekuensi (3) (Sumber: Penulis)

- Mengisi nilai maksimum dengan menggunakan fungsi *MAX*.

Clipboard		Font		Alignment			
D3		X	✓	<i>f_x</i>	=MAX(A2:A101)		
	A	B	C	D	E	F	G
1	x		Banyak Kelas (k)	8			
2	74		MIN	1			
3	90		MAX	=MAX(A2:A101)			
4	60		Panjang Kelas(i)				
5	97						
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI
7	18						
8	77						
9	26						

Gambar 2.133 Latihan Tabel Frekuensi (4) (Sumber: Penulis)

5. Mengisi panjang kelas dengan menggunakan rumus $i = \frac{\max - \min}{k}$, kemudian dibulatkan ke atas dengan menggunakan fungsi *ROUNDUP*.

The screenshot shows an Excel spreadsheet with the following data and formula:

	A	B	C	D	E	F
1	x		Banyak Kelas (k)	8		
2	74		MIN	1		
3	90		MAX	99		
4	60		Panjang Kelas(i)	<code>=ROUNDUP((D3-D2)/D1,0)</code>		
5	97					
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI
7	18					
8	77					
9	26					
10	77					
11	48					
12	62					

Gambar 2.134 Latihan Tabel Frekuensi (5) (Sumber: Penulis)

6. Mengisi batas bawah dengan nilai minimum yaitu 1, kemudian melanjutkannya dengan rumus sebagai berikut.

The screenshot shows an Excel spreadsheet with the following data and formula:

	A	B	C	D	E	F
1	x		Banyak Kelas (k)	8		
2	74		MIN	1		
3	90		MAX	99		
4	60		Panjang Kelas(i)	13		
5	97					
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI
7	18		1	1		
8	77		2	<code>=D7+\$D\$4</code>		
9	26		3			
10	77		4			

Gambar 2.135 Latihan Tabel Frekuensi (6) (Sumber: Penulis)

7. Mengisi kolom batas atas dengan rumus sebagai berikut.

The screenshot shows an Excel spreadsheet with the following data and formula:

	A	B	C	D	E	F
1	x		Banyak Kelas (k)	8		
2	74		MIN	1		
3	90		MAX	99		
4	60		Panjang Kelas(i)	13		
5	97					
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI
7	18		1	1	<code>=D7+\$D\$4-1</code>	
8	77		2		14	
9	26		3		27	
10	77		4		40	

Gambar 2.136 Latihan Tabel Frekuensi (7) (Sumber: Penulis)

8. Mengisi frekuensi dengan 2 cara, yang pertama yaitu pada kolom F7 dengan menggunakan fungsi COUNTIFS.

SUM		=COUNTIFS(\$A\$2:\$A\$101,">="&D7,\$A\$2:\$A\$101,"<="&E7)							
	A	B	C	D	E	F	G	H	I
1	x		Banyak Kelas (k)	8					
2	74		MIN	1					
3	90		MAX	99					
4	60		Panjang Kelas (i)	13					
5	97								
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi
7	18		1	1	13	=COUNTIFS(\$A\$2:\$A\$101,">="&D7,\$A\$2:\$A\$101,"<="&E7)			
8	77		2	14	26				
9	26		3	27	39				
10	77		4	40	52				

Gambar 2.137 Latihan Tabel Frekuensi (8) (Sumber: Penulis)

9. Mengisi kolom frekuensi dengan cara yang kedua yaitu menggunakan fungsi *FREQUENCY*. Langkah pertama yaitu memblok area kolom frekuensi yaitu G7 sampai G14, lalu ketikkan rumus sebagai berikut.

G7		=FREQUENCY(A2:A101,E7:E14)							
	A	B	C	D	E	F	G	H	
1	x		Banyak Kelas (k)	8					
2	74		MIN	1					
3	90		MAX	99					
4	60		Panjang Kelas (i)	13					
5	97								
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi
7	18		1	1	13	8	=FREQUENCY(A2:A101,E7:E14)		
8	77		2	14	26	15			
9	26		3	27	39	14			
10	77		4	40	52	10			
11	48		5	53	65	12			
12	53		6	66	78	21			
13	92		7	79	91	9			
14	71		8	92	104	11			
15	73								

Gambar 2.138 Latihan Tabel Frekuensi (9) (Sumber: Penulis)

10. Letakkan kursor pada *formula bar*, kemudian tekan CTRL+SHIFT+ENTER secara bersama-sama.

G7		=FREQUENCY(A2:A101,E7:E14)							
	A	B	C	D	E	F	G	H	I
1	x		Banyak Kelas (k)	8					
2	74		MIN	1					
3	90		MAX	99					
4	60		Panjang Kelas (i)	13					
5	97								
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi
7	18		1	1	13	8	=FREQUENCY(A2:A101,E7:E14)		
8	77		2	14	26	15			
9	26		3	27	39	14			
10	77		4	40	52	10			
11	48		5	53	65	12			
12	53		6	66	78	21			
13	92		7	79	91	9			
14	71		8	92	104	11			
15	73								

Gambar 2.139 Latihan Tabel Frekuensi (10) (Sumber: Penulis)

11. Menjumlahkan kolom frekuensi menggunakan fungsi SUM.

The screenshot shows an Excel spreadsheet with a frequency table. The formula bar at the top displays $=SUM(G7:G14)$. The table has columns A through H. Row 15 (G15) contains the formula $=SUM(G7:G14)$ in the G column, which is highlighted with a red box.

	A	B	C	D	E	F	G	H
1	x		Banyak Kelas (k)	8				
2	74		MIN	1				
3	90		MAX	99				
4	60		Panjang Kelas(i)	13				
5	97							
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF
7	18		1	1	13	8	8	
8	77		2	14	26	15	15	
9	26		3	27	39	14	14	
10	77		4	40	52	10	10	
11	48		5	53	65	12	12	
12	53		6	66	78	21	21	
13	92		7	79	91	9	9	
14	71		8	92	104	11	11	
15	73						$=SUM(G7:G14)$	
16	87							

Gambar 2.140 Latihan Tabel Frekuensi (11) (Sumber: Penulis)

12. Mengisi kolom frekuensi relatif dengan rumus sebagai berikut.

The screenshot shows an Excel spreadsheet with a frequency table. The formula bar at the top displays $=G7/SG$15$. The table has columns F through I. Row 8 (H8) contains the formula $=G7/SG$15$ in the H column, which is highlighted with a red box.

	F	G	H	I
7	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	X_i
8	8	8	$=G7/SG$15$	
9	15	15		
10	14	14		
11	10	10		
12	12	12		
13	21	21		
14	9	9		
15	11	11		
16		100		

Gambar 2.141 Latihan Tabel Frekuensi (12) (Sumber: Penulis)

13. Mengisi kolom xi dengan rumus $x_i = (\text{batas bawah} + \text{batas atas})/2$.

The screenshot shows an Excel spreadsheet with a frequency table. The formula bar at the top displays $= (D7+E7)/2$. The table has columns A through J. Row 8 (I8) contains the formula $= (D7+E7)/2$ in the I column, which is highlighted with a red box.

	A	B	C	D	E	F	G	H	I	J
1	x		Banyak Kelas (k)	8						
2	74		MIN	1						
3	90		MAX	99						
4	60		Panjang Kelas(i)	13						
5	97									
6	93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	x_i	$f_i \cdot x_i$
7	18		1	1	13	8	8	0.08	$= (D7+E7)/2$	
8	77		2	14	26	15	15	0.15		
9	26		3	27	39	14	14	0.14		
10	77		4	40	52	10	10	0.1		
11	48		5	53	65	12	12	0.12		

Gambar 2.142 Latihan Tabel Frekuensi (13) (Sumber: Penulis)

14. Mengalikan xi dengan frekuensi sebagai berikut.

=G7*17										
A	B	C	D	E	F	G	H	I	J	
x		Banyak Kelas (k)	8							
74		MIN	1							
90		MAX	99							
60		Panjang Kelas(l)	13							
97										
93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi	Fi.Xi	M
18		1	1	13	8	8	0.08	7	=G7*17	
77		2	14	26	15	15	0.15	20		
26		3	27	39	14	14	0.14	33		
77		4	40	52	10	10	0.1	46		
48		5	53	65	12	12	0.12	59		
53		6	66	78	21	21	0.21	72		
92		7	79	91	9	9	0.09	85		
71		8	92	104	11	11	0.11	98		

Gambar 2.143 Latihan Tabel Frekuensi (14) (Sumber: Penulis)

15. Menjumlahkan hasil kali xi dan frekuensi menggunakan fungsi SUM.

=SUM(J7:J14)										
A	B	C	D	E	F	G	H	I	J	
x		Banyak Kelas (k)	8							
74		MIN	1							
90		MAX	99							
60		Panjang Kelas(l)	13							
97										
93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi	Fi.Xi	
18		1	1	13	8	8	0.08	7	56	
77		2	14	26	15	15	0.15	20	300	
26		3	27	39	14	14	0.14	33	462	
77		4	40	52	10	10	0.1	46	460	
48		5	53	65	12	12	0.12	59	708	
53		6	66	78	21	21	0.21	72	1512	
92		7	79	91	9	9	0.09	85	765	
71		8	92	104	11	11	0.11	98	1078	
73						100			=SUM(J7:J14)	
87										

Gambar 2.144 Latihan Tabel Frekuensi (15) (Sumber: Penulis)

16. Menghitung mean dengan rumus $mean=\frac{\sum f_i.x_i}{\sum f_i}$.

=J15/G15										
A	B	C	D	E	F	G	H	I	J	K
x		Banyak Kelas (k)	8							
74		MIN	1							
90		MAX	99							
60		Panjang Kelas(l)	13							
97										
93		INTERVAL	BATAS BAWAH	BATAS ATAS	FREKUENSI	FREKUENSI	FREKUENSI RELATIF	Xi	Fi.Xi	MEAN
18		1	1	13	8	8	0.08	7	56	=J15/G15
77		2	14	26	15	15	0.15	20	300	
26		3	27	39	14	14	0.14	33	462	
77		4	40	52	10	10	0.1	46	460	
48		5	53	65	12	12	0.12	59	708	
53		6	66	78	21	21	0.21	72	1512	
92		7	79	91	9	9	0.09	85	765	
71		8	92	104	11	11	0.11	98	1078	
73					100					
87									5341	

Gambar 2.145 Latihan Tabel Frekuensi (16) (Sumber: Penulis)

2.1.16 Pivot Tabel

Latihan membuat pivot tabel dapat dilakukan dengan langkah-langkah sebagai berikut.

1. Disajikan data sebagai berikut.

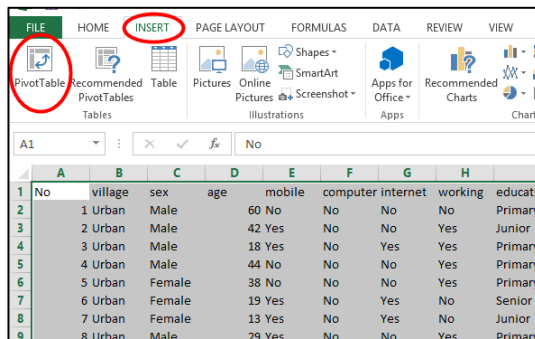
	A	B	C	D	E	F	G	H	I	J
1	No	village	sex	age	mobile	compute	internet	working	education	
2	1	Urban	Female	60	No	No	No	No	Primary	
3	2	Urban	Male	42	Yes	No	No	Yes	Junior	
4	3	Urban	Male	18	Yes	No	Yes	Yes	Primary	
5	4	Urban	Male	44	No	No	No	Yes	Primary	
6	5	Urban	Female	38	No	No	No	Yes	Primary	
7	6	Urban	Female	19	Yes	No	Yes	No	Senior	
8	7	Urban	Female	13	Yes	No	Yes	No	Junior	
9	8	Urban	Male	29	Yes	No	No	Yes	Primary	
10	9	Urban	Female	25	Yes	No	No	No	Junior	
11	10	Urban	Male	62	No	No	No	Yes	Primary	
12	11	Urban	Female	52	No	No	No	Yes	Primary	
13	12	Urban	Male	14	Yes	Yes	Yes	No	Junior	
14	13	Urban	Male	37	Yes	No	No	Yes	Junior	
15	14	Urban	Female	38	Yes	No	No	No	Junior	
16	15	Urban	Female	73	No	No	No	Yes	No-Education	
17	16	Urban	Male	33	Yes	No	No	Yes	Junior	
18	17	Urban	Female	30	Yes	No	No	Yes	Senior	
19	18	Urban	Female	8	No	No	No	No	Primary	
20	19	Urban	Male	34	No	No	No	Yes	Primary	
21	20	Urban	Female	27	Yes	No	No	No	Junior	
22	21	Urban	Male	34	Yes	No	No	Yes	Primary	
23	22	Urban	Female	32	Yes	No	No	Yes	Primary	
24	23	Urban	Female	11	No	No	No	No	Primary	
25	24	Urban	Male	6	No	No	No	No	Primary	
26	25	Urban	Male	53	No	No	No	Yes	Primary	
27	26	Urban	Female	52	No	No	No	Yes	Primary	
28	27	Urban	Male	26	Yes	No	Yes	Yes	Primary	
29	28	Urban	Female	71	No	No	No	No	Primary	
30	29	Urban	Male	53	Yes	Yes	Yes	Yes	Junior	
31	30	Urban	Female	49	Yes	No	No	Yes	Senior	
32	31	Urban	Male	31	Yes	Yes	Yes	Yes	Higher	
33	32	Urban	Female	24	Yes	Yes	Yes	Yes	Primary	
34	33	Urban	Male	43	Yes	No	No	Yes	Senior	
35	34	Urban	Female	51	Yes	No	No	Yes	Junior	
36	35	Urban	Male	18	Yes	Yes	Yes	No	Senior	
37	36	Urban	Male	97	Yes	No	No	No	Primary	
38	37	Urban	Female	70	Yes	No	No	No	Primary	
39	38	Urban	Male	40	No	No	No	No	Junior	
40	39	Urban	Male	57	Yes	No	No	Yes	Senior	
41	40	Urban	Female	51	Yes	No	No	Yes	Senior	
42	41	Urban	Female	15	Yes	Yes	Yes	No	Junior	
43	42	Urban	Male	12	Yes	Yes	Yes	No	Primary	
44	43	Urban	Male	62	Yes	No	No	Yes	Higher	
45	44	Urban	Female	48	Yes	No	No	No	Junior	
46	45	Urban	Male	15	Yes	Yes	Yes	No	Junior	

47	46	Urban	Male	60	Yes	No	No	Yes	Senior
48	47	Urban	Female	47	Yes	No	No	Yes	Primary
49	48	Urban	Male	15	Yes	No	No	No	Primary
50	49	Urban	Male	28	Yes	No	No	Yes	Senior
51	50	Urban	Female	27	Yes	No	No	Yes	Senior
52	51	Urban	Female	5	Yes	No	No	No	No-Education
53	52	Urban	Male	50	Yes	No	No	Yes	Junior
54	53	Urban	Male	56	Yes	Yes	Yes	Yes	Higher
55	54	Urban	Female	53	Yes	No	Yes	Yes	Senior
56	55	Urban	Male	38	Yes	Yes	Yes	Yes	Senior
57	56	Urban	Female	31	Yes	Yes	Yes	Yes	Higher
58	57	Urban	Male	10	Yes	Yes	Yes	No	Primary
59	58	Urban	Male	16	Yes	No	Yes	No	Senior
60	59	Urban	Male	18	Yes	No	Yes	No	Higher
61	60	Urban	Male	56	Yes	No	No	Yes	Senior
62	61	Urban	Female	43	Yes	No	No	No	Junior
63	62	Urban	Male	31	Yes	Yes	Yes	Yes	Higher
64	63	Urban	Female	28	Yes	Yes	Yes	Yes	Higher
65	64	Urban	Male	57	Yes	No	No	Yes	Primary
66	65	Urban	Female	48	Yes	No	No	Yes	Primary
67	66	Urban	Male	17	Yes	No	Yes	No	Junior
68	67	Urban	Male	34	Yes	Yes	Yes	Yes	Higher
69	68	Urban	Female	33	Yes	Yes	Yes	Yes	Higher
70	69	Urban	Male	61	Yes	No	No	No	Senior
71	70	Urban	Female	56	Yes	No	No	Yes	Primary
72	71	Urban	Female	37	Yes	Yes	Yes	Yes	Higher
73	72	Urban	Female	7	Yes	No	No	No	Primary
74	73	Urban	Female	51	Yes	Yes	Yes	Yes	Primary
75	74	Urban	Female	16	Yes	Yes	Yes	No	Senior
76	75	Urban	Male	37	Yes	No	No	Yes	Senior
77	76	Urban	Female	30	Yes	No	Yes	Yes	Senior
78	77	Urban	Female	7	Yes	No	No	No	Primary
79	78	Urban	Male	54	Yes	Yes	Yes	Yes	Senior
80	79	Urban	Female	57	Yes	Yes	Yes	Yes	Primary
81	80	Urban	Female	28	Yes	Yes	Yes	Yes	Higher
82	81	Urban	Male	84	No	No	No	Yes	No-Education
83	82	Urban	Female	75	No	No	No	No	No-Education
84	83	Urban	Male	46	Yes	No	No	Yes	Senior
85	84	Urban	Female	42	Yes	No	No	Yes	Primary
86	85	Urban	Female	12	Yes	Yes	Yes	No	Primary
87	86	Urban	Female	11	Yes	Yes	Yes	No	Primary
88	87	Urban	Male	51	Yes	No	No	Yes	Senior
89	88	Urban	Female	45	Yes	No	No	Yes	Senior
90	89	Urban	Female	26	Yes	Yes	Yes	Yes	Higher
91	90	Urban	Male	9	Yes	No	No	No	Primary

92	91	Urban	Female	65	Yes	No	No	No	Higher
93	92	Urban	Female	29	Yes	Yes	Yes	Yes	Higher
94	93	Urban	Male	30	Yes	Yes	Yes	Yes	Higher
95	94	Urban	Male	43	Yes	No	No	Yes	Primary
96	95	Urban	Female	43	Yes	No	No	No	Junior
97	96	Urban	Female	71	No	No	No	No	No-Education
98	97	Urban	Male	76	Yes	No	No	No	Primary
99	98	Urban	Female	71	Yes	No	No	No	Primary
100	99	Urban	Male	47	Yes	No	No	Yes	Primary
101	100	Urban	Female	43	Yes	No	No	No	Primary
102									
103									

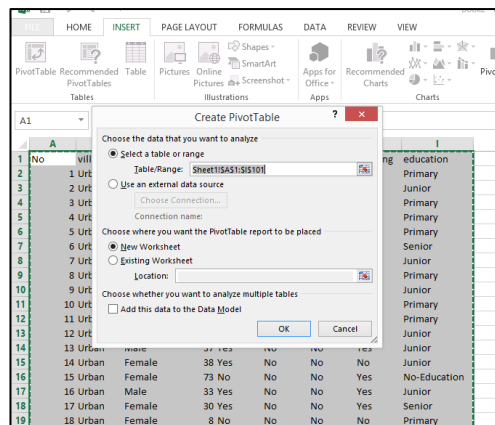
Gambar 2.146 Latihan Pivot Tabel (1) (Sumber: Penulis)

2. Blok semua sel pada tabel, pilih *insert*, kemudian pilih *pivot table*.



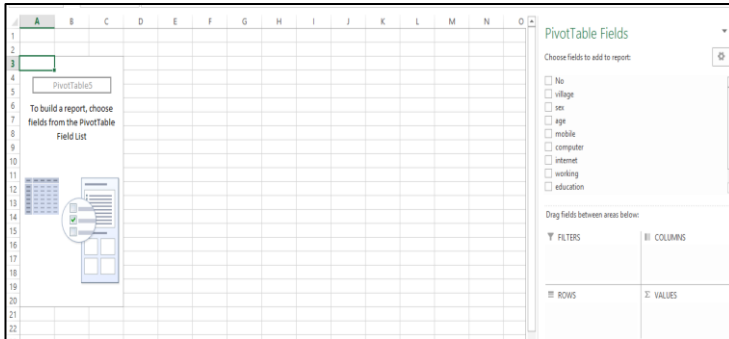
Gambar 2.147 Latihan Pivot Tabel (2) (Sumber: Penulis)

3. Kemudian akan muncul tampilan seperti berikut. Pilih *New Worksheet* untuk meletakkan pivot tabel di *worksheet* yang baru, lalu pilih *ok*.



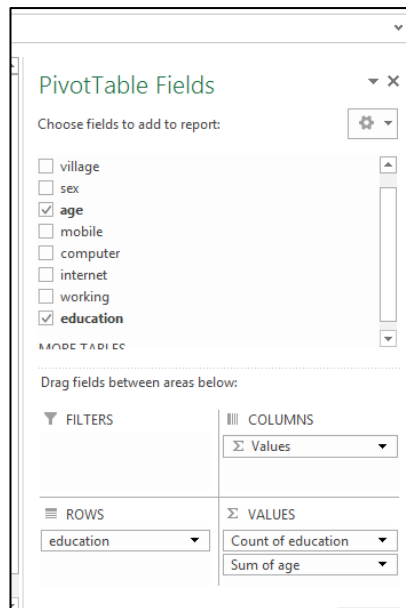
Gambar 2.148 Latihan Pivot Tabel (3) (Sumber: Penulis)

4. Maka akan muncul tampilan sebagai berikut.



Gambar 2.149 Latihan Pivot Tabel (4) (Sumber: Penulis)

5. Isi tabel dengan konten yang diinginkan dengan cara mencentang atau *drag* ke area yang diinginkan. Misalnya memunculkan jumlah dari kategori *education* dan rata-rata umur.



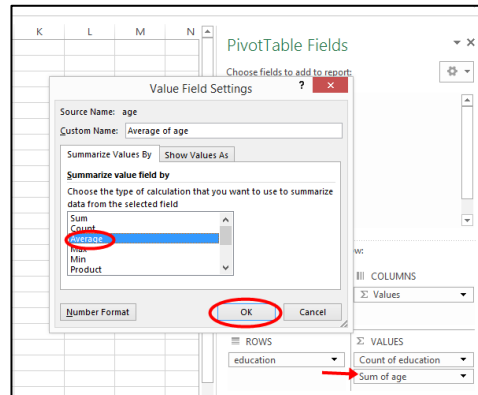
Gambar 2.150 Latihan Pivot Tabel (5) (Sumber: Penulis)

6. Maka pada sel akan muncul tampilan sebagai berikut.

	A	B	C
1			
2			
3	Row Labels	Count of education	Sum of age
4	Higher	15	539
5	Junior	18	604
6	No-Education	5	308
7	Primary	40	1542
8	Senior	22	885
9	Grand Total	100	3878
10			
11			

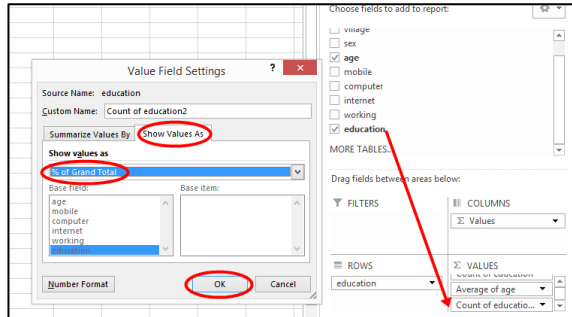
Gambar 2.151 Latihan Pivot Tabel (6) (Sumber: Penulis)

7. Untuk memunculkan rata-rata umur dapat dilakukan dengan mengklik *sum of age* pada *field values*, pilih *value field setting*, pilih *average*, kemudian pilih *ok*.



Gambar 2.152 Latihan Pivot Tabel (7) (Sumber: Penulis)

8. Selain itu, kategori-kategori dalam *education* dapat dimunculkan persentasenya yaitu dengan cara *drag education* ke *field value* kemudian klik *count of education*, pilih *value field setting*, pilih *show values as*, atur menjadi *% of grand total*, kemudian pilih *ok*.



Gambar 2.153 Latihan Pivot Tabel (8) (Sumber: Penulis)

9. Maka tampilan pivot tabel menjadi seperti berikut.

Row Labels	Count of education	Average of age	Count of education2
Higher	15	35.93333333	15.00%
Junior	18	33.55555556	18.00%
No-Education	5	61.6	5.00%
Primary	40	38.55	40.00%
Senior	22	40.22727273	22.00%
Grand Total	100	38.78	100.00%

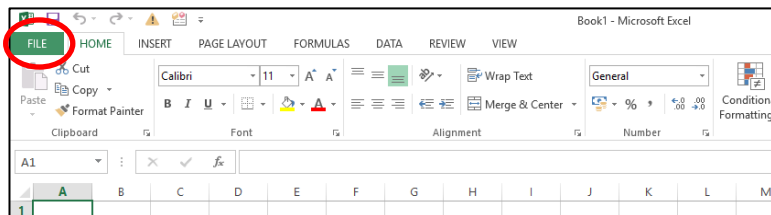
Gambar 2.154 Latihan Pivot Tabel (9) (Sumber: Penulis)

2.2 Mengelola Visual Basic

2.2.1 Mengaktifkan Visual Basic pada Microsoft Excel

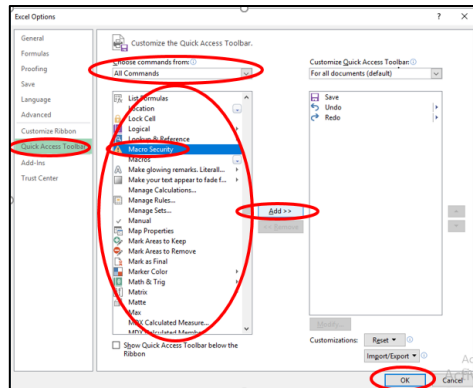
Langkah-langkah untuk mengaktifkan *visual basic* pada *Microsoft Excel* dapat dilakukan dengan cara sebagai berikut.

1. Buka *Microsoft Excel*
2. Pada menu bar klik menu *File*



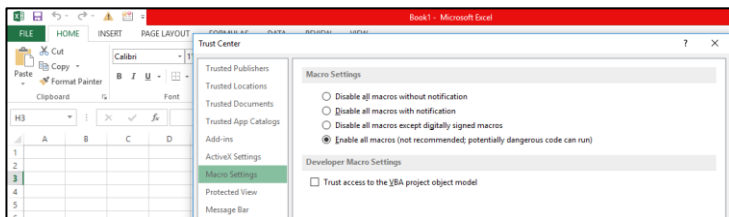
Gambar 2.155 Ms. Excel (Sumber: Penulis)

- Pilih menu *Option*, kemudian akan muncul jendela seperti dibawah ini. Pilih *Quick Access Toolbar*, pada kolom *Choose commands from* ubah menjadi *All Commands* cari *command Macro Security*, klik *Add*. Ulangi langkah tersebut untuk menambahkan *command Visual Basic*. Terakhir klik *OK*



Gambar 2.156 Option Ms. Excel (Sumber: Penulis)

- Klik *Macro Settings*, akan muncul dialog seperti dibawah ini dan klik *“Enable all macros...”*



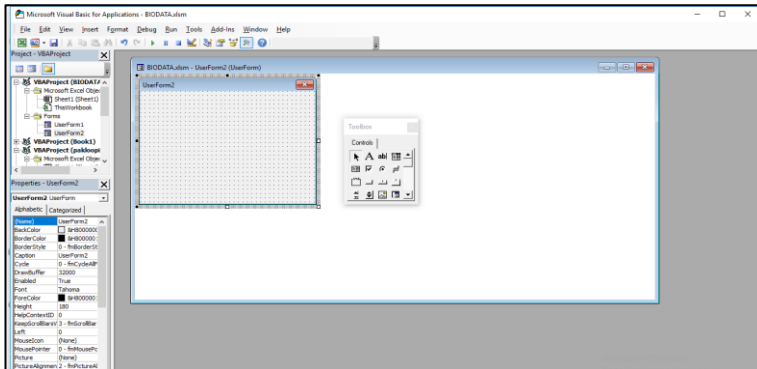
Gambar 2.157 Visual Basic (Sumber: Penulis)

Visual basic siap digunakan.

2.2.2 Membuat Program untuk Menginput Biodata Melalui *Visual Basic* ke dalam Bentuk *Microsoft Excel*

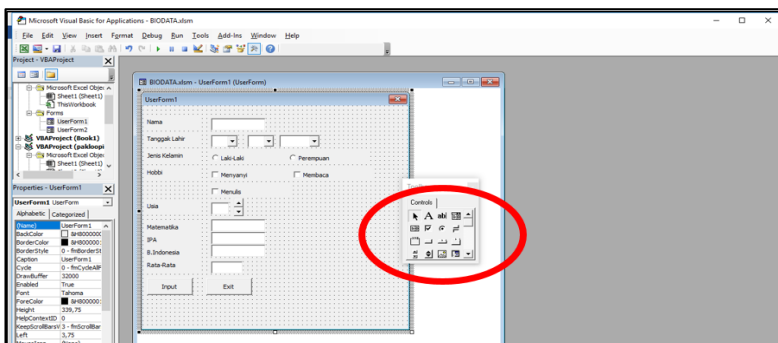
Langkah-langkah dalam membuat program untuk menginput biodata melalui *visual basic* ke dalam bentuk *Microsoft Excel*

- Buka *excel*, klik *visual basic* maka akan keluar jendela seperti dibawah ini



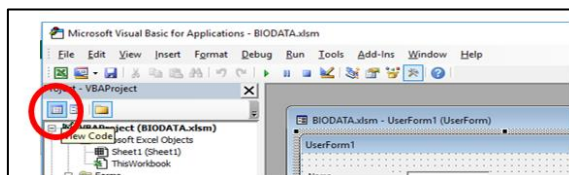
Gambar 2.158 Halaman Visual Basic (Sumber: Penulis)

2. *Input label box, text box, check box, option button, dan command button dari toolbox. Kemudian susun secara rapi seperti gambar dibawah ini*



Gambar 2.159 Tool Box (Sumber: Penulis)

3. *Pada Project-VBAPROJECT, klik view code untuk menuliskan code/fungsi pada VB*



Gambar 2.160 View Code (Sumber: Penulis)

[illegible][illegible]

Microsoft Visual Basic for Applications - WORD4.doc

File Edit View Insert Format Debug Run Tools Add-Ins Window Help

Project - VBAProject

Properties - UserForm1

UserForm1

Name: UserForm1

Target: ThisDocument

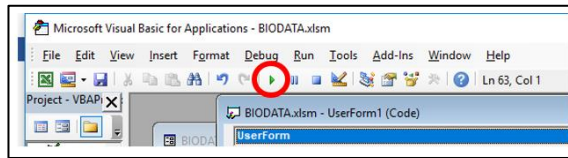
Private Sub CommandButton1_Click()
Dim Me
End Sub

Private Sub SpinButton1_Change()
TextBox8.Value = SpinButton1.Value
SpinButton1.Min = 1
SpinButton1.Max = 30
SpinButton1.SmallChange = 1
End Sub

Private Sub UserForm1_Initialize()
ComboBox1.AddItem "1"
ComboBox1.AddItem "2"
ComboBox1.AddItem "3"
ComboBox1.AddItem "4"
ComboBox1.AddItem "5"
ComboBox1.AddItem "6"
ComboBox1.AddItem "7"
ComboBox1.AddItem "8"
ComboBox1.AddItem "9"
ComboBox1.AddItem "10"
End Sub

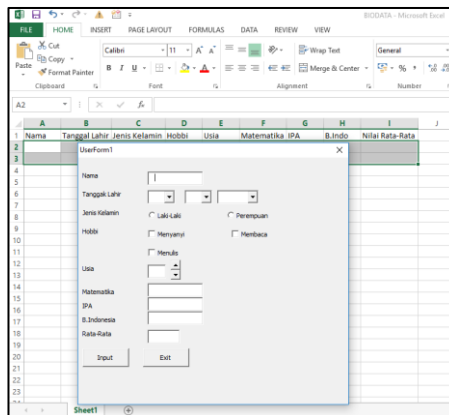
72

- Setelah selesai, klik tombol Fn+F5 atau klik tombol dibawah ini untuk menjalankan *visual basic*.



Gambar 2.162 Menjalankan *Visual Basic* (Sumber: Penulis)

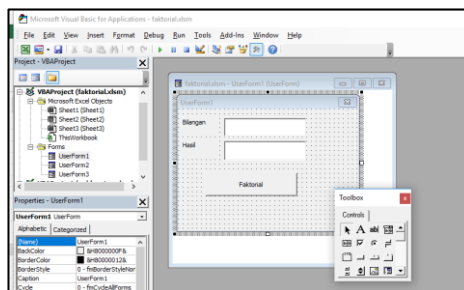
- Maka akan otomatis terbuka pada *excel* dengan tampilan dibawah ini, VB sudah dapat digunakan.



Gambar 2.163 Tampilan *Input Biodata* (Sumber: Penulis)

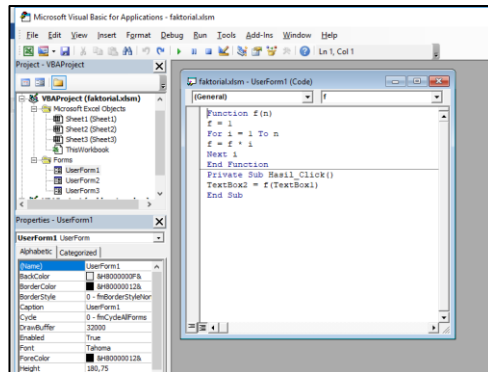
2.2.3 Membuat Mesin Hitung *Factorial* dengan Menggunakan *Visual Basic*

- Buka *excel*, kemudian buka klik *visual basic* jangan lupa pada *macros* klik “*enable all macros ...*”
- Tambahkan label, *text box* dan *command button* sehingga seperti dibawah ini



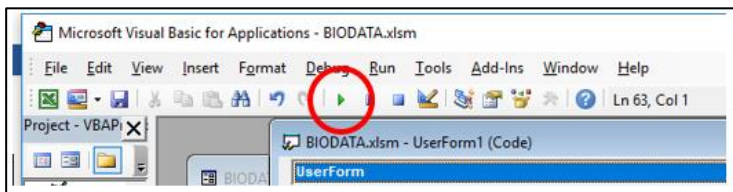
Gambar 2.164 Menampilkan *Tool Box* (Sumber: Penulis)

3. Pada view code tuliskan rumus manual factorial seperti dibawah ini



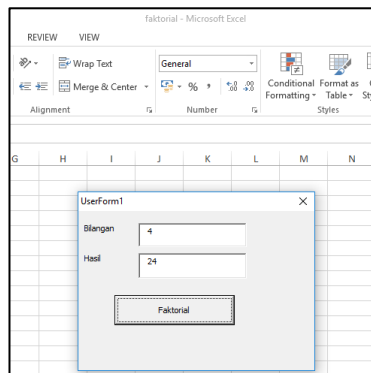
Gambar 2.165 Menuliskan Fungsi Faktorial (Sumber: Penulis)

4. Setelah selesai run visual basic menggunakan Fn+F5 atau mengeklik tombol dibawah ini.



Gambar 2.166 Menjalankan Visual Basic (Sumber: Penulis)

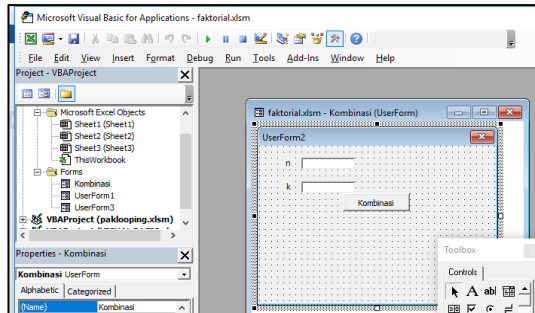
5. Maka akan muncul di excel, saat diisikan nomor factorial yang ingin dihitung di kolom bilangan kemudian klik factorial maka akan muncul hasil faktorial di kolom hasil seperti dibawah ini.



Gambar 2.167 Tampilan Faktorial (Sumber: Penulis)

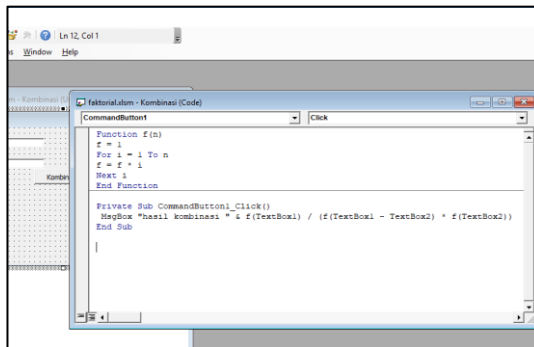
2.2.4 Membuat Mesin Hitung Kombinasi dengan Menggunakan *Visual Basic*

1. Buka *excel*, kemudian buka klik *visual basic* jangan lupa pada *macros* klik “*enable all macros ...*”
2. Tambahkan label, *text box* dan *command button* sehingga seperti dibawah ini



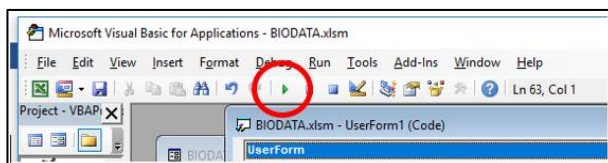
Gambar 2.168 Membuat *Work Sheet* (Sumber: Penulis)

3. Pada *view code* tuliskan rumus manual kombinasi seperti dibawah ini



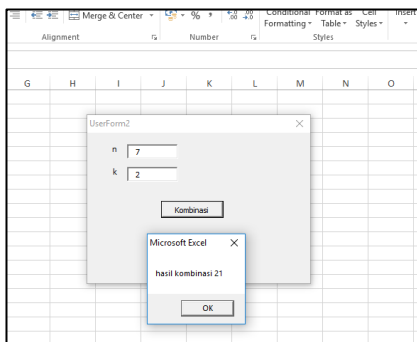
Gambar 2.169 Menuliskan Fungsi Kombinasi (Sumber: Penulis)

4. Setelah selesai *run visual basic* menggunakan *Fn+F5* atau klik tombol dibawah ini



Gambar 2.170 Menjalankan *Visual Basic* (Sumber: Penulis)

5. Maka akan muncul di *excel*, saat diisikan dua angka kombinasi pada kolom n dan k yang ingin dihitung kemudian klik kombinasi maka akan muncul hasil kombinasi di *message box* hasil seperti dibawah ini.



Gambar 2.171 Tampilan Kombinasi (Sumber: Penulis)

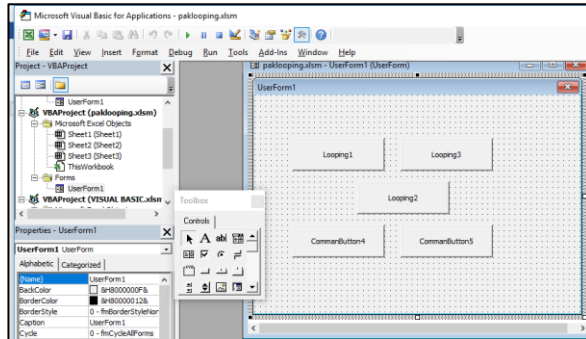
2.2.5 Menerapkan Penggunaan Sistem Looping pada Visual Basic

Sistem *looping* merupakan sistem pengulangan. Pada contoh berikut terdapat 5 contoh penerapan *looping* dalam *visual basic*, dengan diibaratkan dengan *looping 1*, *looping 2*, *looping 3*, *command button 4*, dan *command button 5*. Penjelasan fungsi dari masing-masing *looping* adalah sebagai berikut.

1. *Looping 1*: mengulang munculnya angka 18 pada kolom satu mulai dari baris 1 sampai 6
2. *Looping 2*: memunculkan angka genap antara 1 sampai 30 pada kolom 1
3. *Looping 3*: memunculkan angka ganjil antara 1 sampai 15 pada kolom satu dengan mengosongkan baris dengan nomor genap
4. *Command Button 4*: mengulang angka 15 pada kolom 2 dan baris 2 sampai 6
5. *Command Button 5*: mengulang angka 43 pada kolom 3 sampai 5 dan baris 1 sampai 6

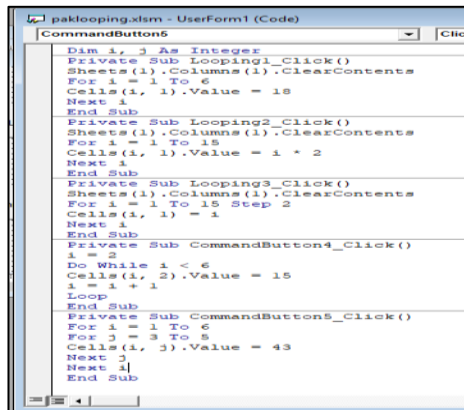
Langkah-langkah penerapan *looping* dalam *visual basic* adalah sebagai berikut.

1. Buka *excel*, kemudian buka klik *visual basic* jangan lupa pada *macros* klik “*enable all macros ...*”
2. Tambahkan 5 *command botton*, ubah sehingga seperti dibawah ini



Gambar 2.172 Menambahkan *Command Button* (Sumber: Penulis)

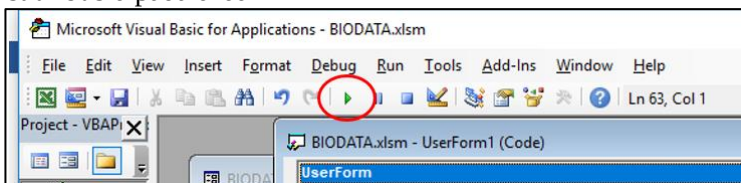
3. Pada view code tuliskan rumus fungsi sebagai berikut



Gambar 2.173 Menuliskan Fungsi (Sumber: Penulis)

Pada bagian paling atas pada code tertera “Dim i, j As Integer” yang berarti bahwa i diasumsikan sebagai baris dan j sebagai kolom, sedangkan integer dimaksudkan sebagai mendefinisikan kolom i dan j sebagai suatu angka atau bilangan.

4. Setelah selesai klik Fn+F5 atau tanda dibawah ini untuk memunculkan visual basic pada excel



Gambar 2.174 Menjalankan Visual Basic (Sumber: Penulis)

BAGIAN III

PENGENALAN BAHASA R

Capaian Pembelajaran: Setelah mempelajari bagian ini mahasiswa diharapkan mampu:

1. Menggunakan tipe data dan operasinya untuk keperluan komputasi
2. Menggunakan struktur kendali dalam pemrograman R
3. Mendeskripsikan data secara numerik dan grafik
4. Menghitung peluang dan menggunakannya dalam statistika inferensia.
5. Menerapkan metode numerik dan optimasi
6. Melakukan pra-pemrosesan data melalui perapihan dan manipulasi data.
7. Memvisualisasikan data menggunakan R
8. Melakukan cloud-computing sederhana

3.1 Manajemen Data

Setelah mempelajari Manajemen Data, diharapkan mahasiswa mempunyai kemampuan untuk menjelaskan dasar-dasar dalam memanajemen data menggunakan R. Materi yang akan dipelajari adalah membuat struktur data dalam bentuk matriks, vektor, *data frame*, *list*, *array*, *factor* dengan beberapa tipe data seperti *numeric*, *logic*, dan *character*, dan *integer*.

Materi yang akan dipelajari adalah membuat struktur data dalam bentuk matriks, vektor, *data frame*, *list*, *array*, *factor* dengan beberapa tipe data seperti *numeric*, *logic*, dan *character*.

3.1.1 Vektor

Vektor merupakan tipe sederhana dari objek data. Vektor atau *array* 2 dimensi adalah struktur data dimana semua unsur atau elemen dalam vektor memiliki tipe kelas data yang sama. Untuk mengakses vektor,

1. Jika *x* adalah objek vektor, maka akses terhadap elemen ke-*i* dilakukan melalui *x[i]*
2. *Index i* berupa bilangan integer atau nama dari *index* atau elemen *logical* dari objek.

Berikut kode untuk membuat vektor pada masing-masing tipe data.

- a. *Character* (berisi karakter atau string)

```
nama <- c("Dina", "Anton", "Andi")
```


b. *Numeric* (kelas object yang berisikan segala jenis angka)

```
tinggi <-c(160,164.5,156.8)
```

c. *Integer* (kelas *object* yang bersikan bilangan bulat)

```
jml_saudara <-c(3,4,5)
```

d. *Logical*

```
abc <-c(TRUE,FALSE,TRUE)
```

didalam vektor, dimungkinkan untuk melakukan editing data antara lain :

- 1) Menampilkan elemen. Misalkan `nama[3]`, `tinggi[1:2]`
- 2) Menampilkan elemen dengan kriteria tertentu. Misalkan `tinggi[tinggi>160]`
- 3) Menghapus elemen tertentu. Misalkan `jml_saudara[-2]`
- 4) Mengganti elemen. Misalkan `abc[1]=FALSE`
- 5) Menambah elemen. Misalkan `append(nama,"Budi")`
- 6) Mengurutkan elemen. Misalkan `sort(tinggi)`
- 7) Menggabungkan data ke bawah. Misalkan `xstack<-rbind(tinggi,tinggi)`
- 8) Menggabungkan data ke samping. Misalkan `xgab<-cbind(tinggi,tinggi)`

3.1.2 Matriks

Matriks atau *array* dua dimensi adalah struktur data yang memiliki *row* dan *column*, unsur-unsur di dalam matriks semua harus memiliki tipe kelas data yang sama. Untuk mengakses Matriks,

1. Jika `mat` adalah objek matriks, maka akses terhadap individual elemen : `mat[m,n]`, atau `mat[i]`,
2. Baris : `mat[m,]`
3. Kolom : `mat[,n]`
4. `m,n, i` adalah bilangan *integer*.

Berikut beberapa kode untuk membuat matriks,

- a. Membuat matriks berisi angka 1 sampai 12 dengan elemen 3 baris dan 4 kolom

```
matrix(1:12,3,4)
```
- b. Mengubah *object* vektor ke matriks

```
vektor<-1:12  
(vektor)<-c(6,2)
```
- c. Menggabungkan 2 matriks berdasarkan baris

`Matrix1<-rbind(matrix,matrix)`

- d. Menggabungkan 2 matriks berdasarkan kolom

`Matrix1<-cbind(matrix,matrix)`

- e. Mengakses elemen matriks

`Matrix[2,3]`

3.1.3 Data Frame

Data frame adalah struktur data dimana satu kolom harus memiliki tipe kelas yang sama, namun kolom yang berbeda dapat memiliki tipe kelas yang berbeda. *Data frame* juga merupakan tipe khusus list yang digunakan untuk menyimpan data dari berbagai macam tipe dalam bentuk matriks. Untuk mengakses *data frame*,

1. Akses terhadap *data frame* berlaku seperti matriks dan *list*

Matriks : `dataolah[m.n]`

List : `dataolah$status` atau `dataolah[[2]]`

2. Akses lain sebagai vektor

`Attach(namedataframe)`

`Detach(namadataframe)`

Misalkan :

`df1<-data.frame(nama,tinggi,jml_saudara)`

nama tinggi jml_saudara

1 Dina 160.0 3

2 Anton 164.5 4

3 Andi 156.8 5

Akses Data Frame

`df1$nama`

`df1$[2,2]`

`df1$nama=="Lily"`

3.1.4 List

List adalah objek R yang memungkinkan elemen dari berbagai tipe kelas data seperti *integer*, *character*, *logical*, dan lainnya. *List* juga dapat berisi matriks atau fungsi sebagai elemennya. Umumnya *list* digunakan untuk menyimpan hasil suatu fungsi. Perintah `unlist` akan membuat *list* menjadi vektor. Untuk mengakses *list*,

1. *By name* : menggunakan operator \$
2. *By position* : menggunakan operator [[]]
3. Sebagai vektor : menggunakan operator []

Berikut contoh kode untuk membuat *List*.

```
list1<-list("man","women",c(10,12,21),TRUE,FALSE,1 9.1)
```

```
## Mengeluarkan elemen ke 2 dan 3 dari list1
```

```
list1[2:3]
```

```
## Mengeluarkan elemen ke-1 dari elemen ke-3 dari list1
```

```
list1[[3]][1]
```

3.1.5 Array

Array dalam pemrograman R secara sederhana disebut sebagai struktur data multidimensi. *Array* dasarnya sama seperti *matrix*, namun *array* dapat mempunyai lebih dari dua dimensi.

1. Membuat *array* 3 dimensi, dimana membangun matriks dimensi 2x2 yang berisikan elemen huruf kapital dengan *function* LETTERS kemudian diulang sebanyak 3 kali

```
array1<-array(LETTERS,dim=c(2,2,3))
```

2. Mengakses *array*

```
object_array[baris,kolom,lembar,buku]
```

```
array1[1,2,3]
```

3.1.6 Factor

Factor adalah objek data yang digunakan untuk data kategori dan menyimpannya sebagai level. *Factor* dapat menyimpan *string* dan *integer*, *factor* berguna untuk suatu kolom yang memiliki nilai uni dalam jumlah terbatas.

1. Data kategori dengan skala pengukuran nominal

```
golongan_darah<-c("A","A","B","0")
```

```
factor(golongan_darah)
```

2. Data kategori dengan skala pengukuran ordinal

```
Tingkat_pendidikan<-c("SD","SMP","SMA")
```

```
Ordered(tingkat_pendidikan)
```

```
Factor_ordinal<-factor(tingkat_pendidikan,levels=tingkat_pendidikan,  
ordered=TRUE)
```

3. Mengakses *Factor*
`Factor_ordinal[1:2]`

Latihan Soal

Buatlah contoh data masing-masing dengan struktur data Vektor, Matriks, *Array*, *Factor*, *List*, dan *Data Frame*.

3.2 Operasi Dasar dan Fungsi Dasar R

Setelah mempelajari operasi dasar dan fungsi dasar R, mahasiswa diharapkan mampu mengoperasikan data dan membuat fungsi dasar untuk perhitungan menggunakan R.

Operasi dasar R terdiri dari operator aritmetika, operator relasi, operator logika, operator prioritas, dan operasi keterkaitan

3.2.1 Operator Aritmetika

Tabel 3.1 Operator Aritmetika

Simbol	Keterangan
+	<i>Addition</i> , untuk operasi penjumlahan
-	<i>Substraction</i> , untuk operasi pengurangan
*	<i>Multiplication</i> , untuk operasi perkalian
/	<i>Divison</i> , untuk operasi pembagian
^	<i>Exponentiation</i> , untuk operasi pemangkatan
%%	<i>Modulus</i> , untuk mencari sisa bagi
%%/%	<i>Integer</i> , untuk mencari bilangan bulat hasil pembagian saja, tanpa sisa bagi

```
> x<-5
> y<-16
> x+y
[1] 21
> x-y
[1] -11
```

```

> x*y
[1] 80
> y/x
[1] 3.2
> y%/3
[1] 5
> y%/x
[1] 3
> y%%x
[1] 1
> y^x
[1] 1048576

```

Fungsi-fungsi standar yang sering digunakan :

1. Fungsi konversi ke *integer* : *round*, *trunc*, *floor*, *ceiling*
2. Fungsi umum : *abs*, *sign*, *log*, *log10*, *sqrt*, *exp*, *sin*, *cos*, *tan*, *asin*, *acos*, *atan*, *sinh*, *cosh*, *tanh*
3. Fungsi jumlah dan *product* dari vektor : *sum*, *prod*, *cusum*, *cumprod*
4. Fungsi Maksimum dan Minimum : *max*, *min*, *cummax*, *cummin*
5. Fungsi *range* (x) menghitung $c(\min(x), \max(x))$
6. Fungsi untuk mengurutkan : *sort*
7. Fungsi untuk membalikan : *rev*

3.2.2 Operator Relasi

Tabel 3.2 Operator Relasi

Simbol	Keterangan	Deskripsi
=	Sama dengan	Bernilai <i>TRUE</i> jika kedua objek bernilai sama
!=	Tidak sama dengan	Bernilai <i>TRUE</i> jika kedua objek tidak bernilai sama
>	Lebih besar dari	Bernilai <i>TRUE</i> jika objek kanan lebih besar dari kiri
<	Lebih kecil dari	Bernilai <i>TRUE</i> jika nilai objek kanan lebih kecil dari kiri

<code>>=</code>	Lebih besar sama dengan	Bernilai <i>TRUE</i> jika nilai objek kanan lebih besar atau sama dengan dari nilai objek kiri
<code><=</code>	Lebih kecil sama dengan	Bernilai <i>TRUE</i> jika nilai objek kanan lebih kecil atau sama dengan dari nilai objek kiri

```

> x<-5
> y<-16
> x<y
[1] TRUE
> x>y
[1] FALSE
> x<=5
[1] TRUE
> y>=20
[1] FALSE
> y==16
[1] TRUE
> x!=5
[1] FALSE

```

3.2.3 Operator Logika

Tabel 3.3 Operator Logika

Simbol	Keterangan
<code>&&</code>	Operator logika <i>AND</i>
<code> </code>	Operator logika <i>OR</i>
<code>!</code>	Operator logika <i>NOT</i>
<code>&</code>	Operator logika <i>AND</i> elemen wise
<code> </code>	Operator logika <i>OR</i> elemen wise

```

> v<-c(TRUE, TRUE, FALSE)
> t<-c(FALSE, FALSE, FALSE)
> v&& t
[1] FALSE

```

```

> v||t
[1] TRUE
> !v
[1] FALSE FALSE TRUE
> v&t
[1] FALSE FALSE FALSE
> v|t
[1] TRUE TRUE FALSE

```

3.2.4 Operator Prioritas

Ketika beberapa operator digunakan dalam satu ekspresi, kita perlu menentukan prioritas operator dengan menggunakan tanda kurung ().

```

> 2+6*5
[1] 32
> (2+6)*5
[1] 40

```

3.2.5 Operator Keterkaitan

Dimungkinkan untuk memiliki beberapa operator dengan prioritas yang sama dalam sebuah ekspresi. Dalam kasus seperti itu, urutan eksekusi ditentukan melalui asosiativitas.

```

> 3/4/5
[1] 0.15
> 3/(4/5)
[1] 3.75

```

3.2.6 Fungsi Dasar Statistika di R

Tabel 3.4 Fungsi Dasar Statistika R

Fungsi	Kegunaan	Contoh : <code>xvec <- c(2,1,4,3,5)</code>
<i>length</i>	menghitung banyak elemen vektor	<i>length(xvec)</i>
<i>sum</i>	menjumlah data	<i>sum(xvec)</i>
<i>min</i>	nilai minimum	<i>min(xvec)</i>

<i>max</i>	nilai maksimum	<i>max(xvec)</i>
<i>mean</i>	menghitung rata-rata	<i>mean(xvec)</i>
<i>sd</i>	menghitung standard deviasi	<i>sd(xvec)</i>
<i>median</i>	Mencari median	<i>median(xvec)</i>
<i>sqrt</i>	menghitung akar bilangan	<i>sqrt(xvec)</i>
<i>exp</i>	menghitung e^x ($e = 2,718281828$)	<i>exp(0)</i>
<i>pi</i>	konstanta pi (3.14)	<i>pi</i>
<i>log</i>	menghitung nilai ln	<i>log(2.71828)</i>
<i>log10</i>	menghitung nilai logaritma basis 10	<i>log10(10)</i>
<i>factorial</i>	Menghitung nilai faktorial	<i>factorial(3)</i>
<i>rep</i>	Mengulangi bilangan	<i>rep(1,10)</i>
<i>seq</i>	Menampilkan serangkaian bilangan dengan interval tertentu	<i>seq(0,10,1)</i>
<i>options</i>	Mengatur banyak digits hasil perhitungan	<i>option(digits=4)</i>

Latihan Soal

Soal 1.

Soal 2.

$$x = 2$$

$$x = 2$$

$$y = \frac{1}{1+x^2}$$

$$y = 1/(1+x*x)$$

$$y = 0.2$$

$$y = 0.2$$

$$y = \frac{x^2}{1+x^2}$$

$$x = 2$$

$$y = 0.8$$

$$y = x*x / (1+x*x)$$

$$X \sim \text{Bernoulli}(p)$$

$$P(X = k) = p^k(1-p)^{1-k}, \quad k = 0, 1$$

Langkah-langkah menghitung nilai peluang

Input nilai p dan k

Hitung nilai $px = p^k(1-p)^{1-k}$

Tampilkan nilai px

Soal 3.

$$\text{Kombinasi}(n, k) = \binom{n}{k} = \frac{n!}{(n-k)! k!}$$

Langkah-langkah

Input n dan k

Hitung $knk = \binom{n}{k}$

Tampilkan knk

Soal 4.

Buat sembarang data !

Tampilkan nilai statistik deskriptifnya !

Simpan nilai statistik deskriptif dalam bentuk list !

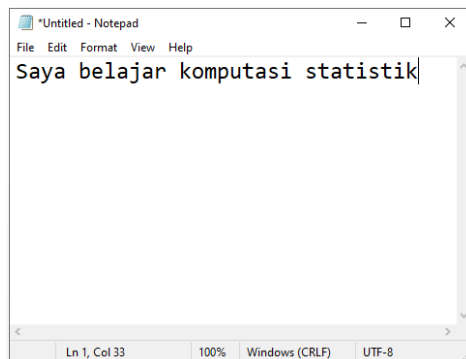
3.3 Manajemen File

Setelah mempelajari manajemen *file*, diharapkan mahasiswa mampu melakukan *import file* dengan format txt, xls, csv, mpj/mtw/mtp, sav ke R serta melakukan *export file* dari R ke txt, xlsx, csv, mpj/mtw/mtp, dan sav.

Setiap *software* selalu memiliki sistem pembuatan *file* tersendiri. Identifikasi jenis *file* bisa terlihat dari ekstensi nya. Misalkan Excel (.xls), Microsoft Word (.doc), Minitab (.mtw/.mpj/.mtp), SPSS (.sav), Database (.dbf), dan lain-lain.

3.3.1 Import File

Untuk kemudahan, tipe *file* yang bisa dibaca oleh semua *file* adalah *file text*. *File text* bisa dibuat dengan *software* Notepad, Wordpad, atau editor-editor di setiap *software*. Sebagai contoh dengan menggunakan Notepad dan disimpan dalam ekstensi .txt. Script/coding umumnya juga tersimpan dalam bentuk *file text*.



Gambar 3.1 Tampilan File Text Menggunakan Notepad (Sumber: Penulis)

Jika *file text* hanya berisi *text* atau tulisan yang tidak mengikuti struktur tertentu, maka akan mudah dibaca oleh *software* lain. Akan tetapi, jika *file text* berisi data terstruktur dalam bentuk baris dan kolom, maka perlu diperhatikan jenis pembatas (separator) apa yang dipergunakan antar kolom. Umumnya, separator yang biasa digunakan adalah koma (,), spasi (), atau semicolon (;), dan tab. Sebagai contoh, kita akan membuat *file* dalam Excel sebagai berikut,

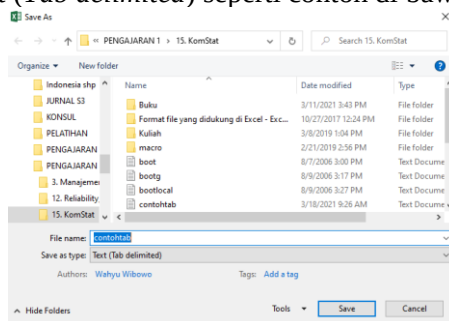
	A	B	C
1	xchar	xnum	
2	adi	6	
3	ani	2	
4	ali	3	
5	ari	4	
6	abi	5	

Gambar 3.2 Tampilan *File* Excel (Sumber: Penulis)

Kemudian disimpan dalam extensi. xls

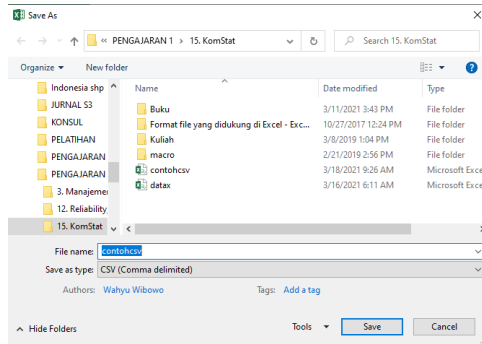
Selanjutnya, *file* akan disimpan dalam bentuk *text* menggunakan tab sebagai penanda antar kolom, maka digunakan

Save as type : Text (Tab delimited) seperti contoh di bawah.



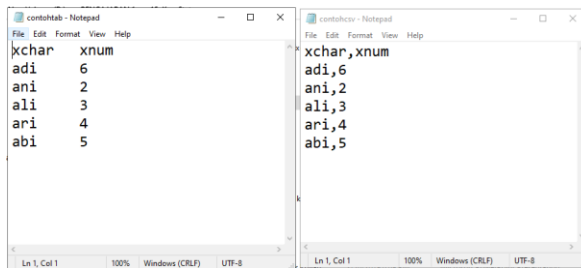
Gambar 3.3 Tampilan Untuk Menyimpan *File* Dalam Bentuk *Text* (Sumber: Penulis)

Selain itu bisa menggunakan *CSV (Comma delimited)* jika ingin menyimpan dalam bentuk *text* dengan koma (,) sebagai penanda antar kolom.



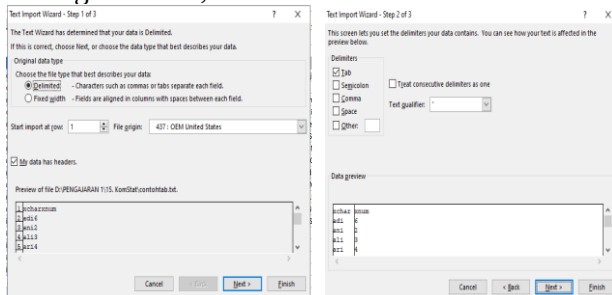
Gambar 3.4 Tampilan Untuk Menyimpan *File* Dalam Bentuk CSV (Sumber: Penulis)

Maka hasilnya adalah sebagai berikut,



Gambar 3.5 Tampilan *File* Dalam Notepad (Sumber: Penulis)

Untuk membuka *file text* dengan pemisah spasi, perlu dilakukan serangkaian *interface* sebagai berikut,



Gambar 3.6 Langkah Membuka *File Text* dengan Pemisah Spasi (Sumber: Penulis)

Berikut akan dijelaskan beberapa *syntax* untuk melakukan import file ke R

1. Dari *Text* ke R

```
datatab <- read.table("d:/contohtab.txt",header=true)
datatab
```

Opsi *header* adalah untuk menentukan apakah ada nama variabel atau tidak pada data di setiap kolomnya

2. Dari CSV ke R

```
datacsv <- read.csv("d:/contohcsv.csv")
datacsv
```

3. Dari Excel ke R

```
mydata <- read.xlsx("d:/ contoh.xlsx", sheetName = "sheet1")
mydata
```

sheetName adalah untuk mengambil data dari *sheet* tertentu atau bisa menggunakan *package* "*readxl*"

```
install.package("readxl")
library(readxl)
mydata1<-read_excel("d:/contoh.xlsx")
mydata1
```

4. Dari SPSS ke R

```
Install.packages("foreign")
library(foreign)
Mydata_spss<-read.spss("d:/contohdata.sav", to.data.frame = TRUE)
view(Mydata_spss)
```

to.data.frame digunakan untuk membuat *data frame* di R

5. Dari Minitab ke R

```
Install.packages("foreign")
library(foreign)
Mydata_mtb <-read.spss("d:/contohdata.mtp")
Mydata_mtb
```

3.3.2 *Export File*

Berikut akan dijelaskan beberapa *syntax* untuk melakukan *export file* dari R

1. Dari R ke text

```
Write.table(mydata, "d:/mydata.txt", sep=""/t")
```

2. Dari R ke CSV

```
write.csv(mydata, file = "d:/mydata.csv, row.names = FALSE)
```

3. Dari R ke excel

```
library(xlsx)  
write.xlsx(mydata, "d:/mydata.xlsx")
```

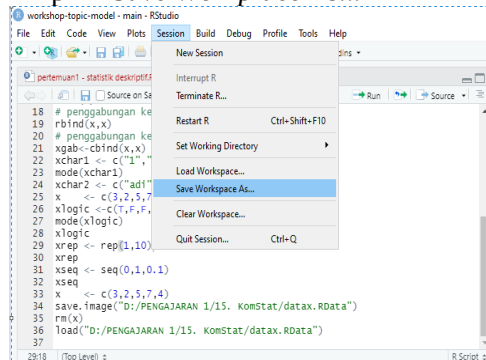
4. R ke SPSS

```
library(foreign)  
write.foreign(mydata, "d:/mydata.txt", "d:/mydata.sav",  
package="SPSS")
```

3.3.3 Menyimpan File Format R

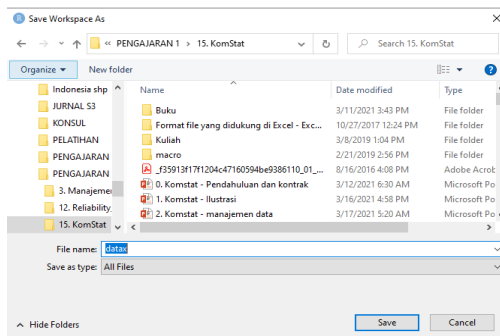
Misalkan kita membuat *project* dengan `x<-c(3,2,5,7,4)` kemudian disimpan dengan format R. Langkah yang dilakukan adalah sebagai berikut

1. Klik Session -> pilih Save Workplace As...



Gambar 3.7 Langkah Awal Menyimpan File Format R (Sumber: Penulis)

2. Pilih *directory* tempat penyimpanan, dan tulis Nama *file* misalnya “datax”

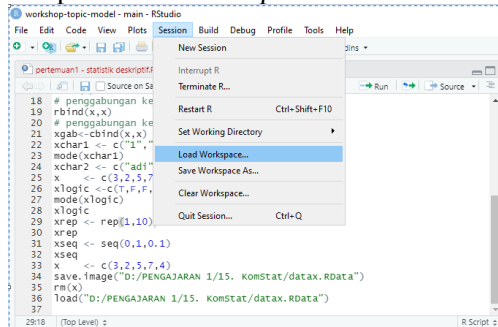


Gambar 3.8 Tampilan Save Workspace As (Sumber: Penulis)

3. Klik Save

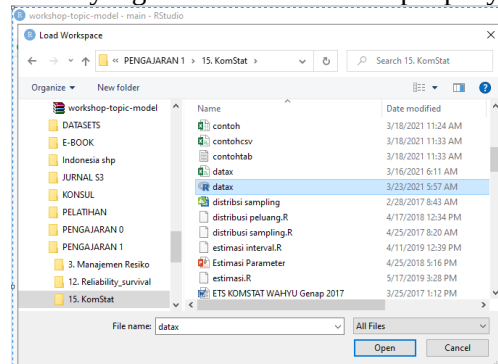
Misalkan kita ingin membaca *file* “datax” yang telah kita buat, maka

1. Klik Session -> pilih Load Workspace



Gambar 3.9 Langkah Awal Membaca File yang Telah Dibuat
(Sumber: Penulis)

2. Pilih file format R yang akan dibuka dari tempat penyimpanan,



Gambar 3.10 Tampilan Load Workspace (Sumber: Penulis)

Latihan Soal

- Buka website <https://archive.ics.uci.edu/ml/index.php>
- *Download* sebarang data, kemudian baca di Excel, Minitab dan R

3.4 Struktur Pemrograman R

Setelah mempelajari struktur pemrograman R diharapkan mahasiswa dapat menjelaskan struktur pemrograman dalam R seperti instruksi *looping*, persyaratan program dijalankan, penghentian program, dan melewati

perintah. Mahasiswa juga diharapkan mampu membuat pemrograman sederhana dengan menerapkan struktur kendali.

Secara umum perintah-perintah dalam program R dijalankan secara urut dari atas ke bawah. Untuk keperluan tersebut, setiap baris program diberi penomoran. Namun demikian, karena kasus pemrograman bermacam-macam dan kadang bersifat kompleks, dimungkinkan perintah tersebut dijalankan tidak hanya satu kali, namun perlu juga diulang, dilakukan dengan syarat, atau tidak perlu dilakukan. Oleh karena itu, perlu suatu struktur pemrograman, agar program dapat dijalankan dengan baik.

3.4.1 Instruksi Pemilihan dengan Fungsi *IF* ()

Pernyataan *if* berisi kondisi yang mengevaluasi ke keluaran logis. Ini menjalankan blok perintah jika kondisi mengevaluasi ke *TRUE*. Kemudian melewati blok perintah jika kondisi mengevaluasi ke *FALSE*.

Syntax bentuk satu kasus :

```
if (condition)
{
  #do something
}
```

Contoh : Jika x lebih besar dari 100, maka nilainya akan ditambah 5

```
if (x>100)
{
  x<-x+5
}
```

Syntax bentuk 2 kasus :

```
if (condition)
{
  #do something
} else {
  #do something else
}
```

Contoh :

```
a<-5
b<-6
if (a>b)
{
  print("a is greater than b")
} else
```



```
{
  print("b is greater than a")
}
```

Syntax bentuk bersusun (lebih dari 1 syarat)

```
nilai<-function(skor)
```

```
{
  if (skor>=80)
  {
    nilai <- "A"
  } else
  {
    if (skor>=60)
    {
      nilai <- "B"
    } else
    {
      nilai <- "C"
    }
  }
  return (nilai)
}
```

3.4.2 Switch

Switch adalah cara yang lebih mudah untuk memilih di antara beberapa alternatif daripada beberapa pernyataan *if-else*. *Switch* R mengambil satu argumen *input* dan mengeksekusi kode tertentu berdasarkan nilai *input*. Setiap nilai yang mungkin dari masukan disebut kasus.

Contoh :

```
a <- 4
switch (a,
  "1"="this is the first case in switch",
  "2"="this is the second case in switch",
  "3"="this is the third case in switch",
  "4"="this is the fourth case in switch",
  "5"="this is the fifth case in switch"
)
```

3.4.3 Instruksi Perulangan (*Loop*)

Instruksi perulangan adalah instruksi yang dapat mengulang pelaksanaan sederetan instruksi-instruksi lainnya berulang-ulang sesuai persyaratan yang ditetapkan. Ada 3 macam bentuk instruksi perulangan yang biasa digunakan dalam program yaitu *for*, *while*, dan *repeat*.

a. *for loops*

Perulangan *for* di R, diulangi urutan untuk melakukan tugas yang berulang. Perintah ini bekerja dengan variabel *iterable* untuk melewati suatu urutan.

```
for (variable in sequence){  
  code_to_repeat  
}
```

Contoh :

```
vec <- c(1:10)                x<-c(4,1,5,7,2,3)  
  
for (i in vec){               n<-length(x)  
  
  print(vec[i])              akumulasi <-0  
  
}                             for (i in 1:n)  
  
vec1 <- c(11:20)              {  
  
for(i in 1:10){               akumulasi<-akumulasi + x[i]  
  
  print(vec1[i])             }  
  
}                             rata <-akumulasi/n
```

b. *while loops*

Perulangan *while* di R mengevaluasi suatu kondisi. Jika kondisi bernilai *TRUE*, maka akan melalui blok kode, sedangkan jika kondisi bernilai *FALSE*, maka keluar dari perulangan. Perulangan *while* di R terus melakukan perulangan melalui blok kode tertutup selama kondisinya *TRUE*.

```
while (condition){  
  code_to_run  
}
```

Contoh :

```
i<-0
```

```
n<-1
```

```
while (i<10){
```

```
jumlah<-0
```

```
  print (paste("this is iteration  
no",i))
```

```
  while (jumlah <=1000)
```

```
  {
```

```
    i <-i+1
```

```
    jumlah<-jumlah+n
```

```
  }
```

```
  n<-n+1
```

```
}
```

c. Repeat Loop

Repeat loop akan menjalankan *statement/code* yang sama berulang-ulang hingga *stop condition* tercapai.

```
repeat {  
  commands  
  if(condition){  
    break  
  }  
}
```

Berikut contoh penerapan dari *repeat loop*

```
coba <- c("contoh")
```

```
counter <- 1
```

```
repeat {  
  print(coba)  
  counter <- counter + 1  
  if(counter < 5){  
    break  
  }  
}
```

d. Break Statement

Pernyataan *break* digunakan untuk keluar dari *loop*. Andaikan sebuah *loop* mencari elemen tertentu secara berurutan, perulangan harus terus berjalan hingga menemukan elemen atau hingga akhir urutan. Jika menemukan elemen lebih awal, perulangan lebih lanjut tidak diperlukan.

Dalam kasus seperti itu, pernyataan R *break* dapat memutuskan kita keluar dari *loop* lebih awal.

Contoh untuk *for loop* :

```
vec <- c(1:10)
for(i in vec){
  print(paste("this is iteration no ", i))
  if(i==7){
    print("break!!")
    break
  }
}
```

Contoh untuk *while loop* :

```
# while loop
a = 2
b = 4
while(a<7){
  print(a)
  a = a + 1
  if(b+a>10){
    break
  }
}
```

Contoh untuk *repeat loop* :

```
# repeat loop
a = 1
repeat{
  print(a)
  a = a+1
  if(a>6){
    break
  }
}
```

e. Next Statement

Pernyataan *next* di R menyebabkan *loop* melewati iterasi saat ini dan memulai yang berikutnya.

Contoh :

```
vec <- c(1:10)
for (i in vec){
```

```

if (i==5 || i==7){
  print("next!!")
  next
}
print(paste("this is iteration no",i))
}

```

3.4.4 Function Standar di R

Function adalah kode-kode yang disusun untuk melakukan tugas tertentu, seperti perhitungan matematis, pembacaan data, analisis statistik, dan lainnya. Seperti halnya fungsi matematika yang kita kenal, $f(x)$ yang akan memetakan nilai x_1 ke $f(x_1)$ sesuai dengan aturan pemetaannya. Dalam fungsi R, fungsi membutuhkan/tidak membutuhkan *input* seperti x_1 dan mengeluarkan *output* seperti $f(x_1)$.

Struktur fungsi dalam R adalah sebagai berikut,

```

function_name <- function(arg_1, arg_2, ...) {
  Function body
}

```

Keterangan :

- *Function name* : Nama fungsi yang akan dibuat
- *Argument* : Saat suatu fungsi dipanggil, kita akan memberikan nilai ke argumen
- *Function Body* : *Function body* mengandung kumpulan-kumpulan *statement* yang mendefinisikan apa yang akan dilakukan fungsi
- *Return* : Nilai kembalian suatu fungsi adalah ekspresi terakhir pada *function body* yang akan dievaluasi

Contoh.

```

1. new_function <-function(a,b,c){
  result <- a*b+c
  print (result)
}

```

```

2. deskriptif <-function(x){
  rata <- mean(x)
  stdev<- sd(x)
}

```

```
hasil<- c(rerata=rata,stdev=stdev)
return(hasil)
}
```

3.4.5 R-Package

R-package merupakan kumpulan fungsi dan set data yang dikembangkan oleh komunitas. Komunitas ini meningkatkan kegunaan R dengan meningkatkan fungsionalitas dasar R yang ada, atau menambahkan yang baru. Misalnya jika berkerja dengan basis data, terdapat *package dplyr* atau *data.table* yaitu dua paket R yang paling populer. Terdapat ratusan R-package yang dapat diinstall sesuai keperluan. Package yang tersedia dapat diaktifkan dengan klik pada *package* yang bersesuaian.

Untuk mencari informasi R-package dapat digunakan syntax

```
packageDescription("package") => packageDescription("MASS")
```

```
help(package = "package") => help(package = "MASS")
```

Package disimpan di *Repository* sehingga dapat diinstall oleh pengguna yang mengakses. Beberapa *Repository* yang tersedia adalah sebagai berikut

CRAN : <https://cran.r-project.org/>

Bioconductor : <https://www.bioconductor.org/>

Github : <https://github.com/>

Latihan Soal

1. Buatlah program untuk memberikan bilangan ganjil atau genap.
2. Buatlah program untuk menghitung berapa banyak bilangan ganjil dan genap dari suatu vektor bilangan bulat.
3. Buatlah function untuk menghitung luas segita, luas persegi, dan luas persegi panjang.

3.5 Visualisasi Data (*Data Visualization*)

Setelah mempelajari visualisasi data, diharapkan mahasiswa mampu memilih dan menyajikan data dengan paket program R. Dalam statistika, visualisasi merupakan bagian penting dalam analisis data, baik untuk *pre-processing* maupun dalam analisis utama. Kelebihan visualisasi data adalah sebagai berikut

1. Informasi lebih mudah dimengerti dan dipahami
2. Membantu identifikasi pola dan *trend*
3. Mempermudah interaksi dalam proses *story telling*
4. Mempercepat proses pengambilan keputusan

3.5.1 Fungsi Plot ()

Fungsi plot merupakan salah satu fungsi grafik yang paling sering digunakan untuk menampilkan suatu grafik pada R

Contoh :

```
x<-c(1:10)
```

```
plot(x)
```

Agar grafik lebih interaktif dan informatif, beberapa parameter dalam perintah plot dapat diatur sesuai keperluan, misalkan judul, sumbu, warna, dan lain-lain.

```
plot(x, y = NULL, type = "p", xlim = NULL, ylim = NULL,
     log = "", main = NULL, sub = NULL,
     xlab = NULL, ylab = NULL, ann = par("ann"),
     axes = TRUE, frame.plot = axes, panel.first = NULL,
     panel.last = NULL, asp = NA, ...)
```

Menambahkan judul

```
plot(x,main="Latihan Grafik")
```

Menambahkan Keterangan Sumbu

```
plot(x,main="LatihanGrafik",xla="sumbu x",ylab="sumbu y")
```

Mengubah Warna

```
plot(x,main="Latihan Grafik",xlab="sumbu x",ylab="sumbu
y",col="red")
```

Membatasi sumbu x

```
plot(x,xlim=c(3,8)) # membatasi sumbu x
```

Mengubah bentuk tanda titik

```
plot(x,main="Latihan Grafik",xlab="sumbu x",ylab="sumbu
y",col="red",type="l")
```

Contoh :

Buatlah grafik fungsi kuadrat $f(x) = x^2 + 2x + 3$ dengan $x \in [-3,3]$

```
x <- c(seq(-3,3,0.1))
```

```
fx<-x*x+2*x+3
```

```
plot(fx,main="Grafik Fungsi Kuadrat", xlab="X", ylab="Y", col="blue")
```

Selanjutnya, akan dibuat plot dari data suatu fungsi matematika, berikut data yang digunakan

```
set.seed(123)
```

```
x <- seq(from=0, to=10, by=0.1)
```

```
y <- x^2*exp(-x/2)*(1+rnorm(n=length(x), mean=0, sd=0.05))
```

```
par(mfrow=c(1,2),
```

```
  # mengatur margin grafik
```

```
  mar=c(4,4,1.5,1.5),
```

```
  # mengatur margin sumbu
```

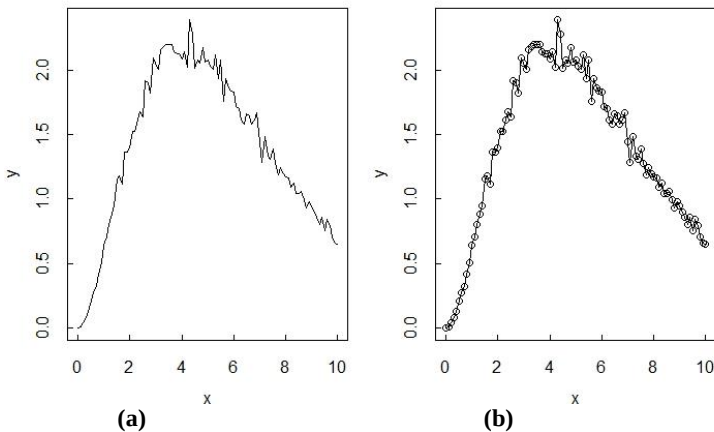
```
  mex=0.8,
```

```
  # arah tick sumbu koordinat
```

```
  tcl=0.3)
```

```
plot(x, y, type="l")
```

```
plot(x, y, type="o")
```



Gambar 3.11 Grafik Fungsi Kuadratik (Sumber: Penulis)

Keterangan :

1. Gambar 3.11 (a) merupakan grafik fungsi kuadratik dengan plot type “l”
2. Gambar 3.11 (a) merupakan grafik fungsi kuadratik dengan plot type “o”

3.5.2 Fungsi Hist ()

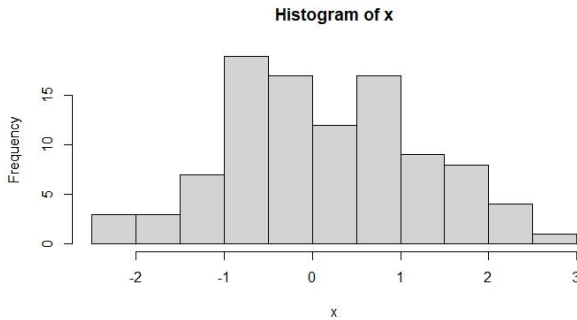
Fungsi *hist* () digunakan untuk membuat histogram pada R.

```
Hist(x, breaks = "Sturges",  
     freq = NULL, probability = !freq,  
     include.lowest = TRUE, right = TRUE,  
     density = NULL, angle = 45, col = "lightgray", border = NULL,  
     main = paste("Histogram of" , xname),  
     xlim = range(breaks), ylim = NULL,  
     xlab = xname, ylab,  
     axes = TRUE, plot = TRUE, labels = FALSE,  
     nclass = NULL, warn.unused = TRUE, ...)
```

Contoh :

Membuat histogram dari data berdistribusi normal (0,1). Selanjutnya diberikan judul “Histogram Data Normal (0,1)”, dan label x diberi nama “Data Simulasi”, dan warna merah

```
x<-rnorm(100)  
hist(x,main="Histogram Data Normal (0,1)", xlab="Data Simulasi")  
hist(x,freq = FALSE, main="Histogram Data Normal (0,1)", xlab="Data  
Simulasi", col="red")  
x<-hist(x)  
objects(x)  
x$breaks  
x$counts  
x$density  
x$mids  
x$xname  
x$equidist
```



Gambar 3.12 Histogram Distribusi Normal (0,1) (Sumber: Penulis)

3.5.3 Menambahkan Jendela Grafik

Untuk menambahkan jendela grafik digunakan fungsi *windows* (), sehingga jendela lama tidak hilang. Misalnya ingin menambahkan *boxplot*.

```
x<-rnorm(100)
hist(x)
windows()
boxplot(x)
```

3.5.4 Membuat Partisi Windows

Jika dalam satu *windows* ingin dipartisi untuk menampilkan beberapa grafik maka digunakan *syntax* sebagai berikut.

```
x<-rnorm(100)
par(mfrow=c(2,1))
hist(x)
boxplot(x)
```

3.5.5 Fungsi Curve ()

Fungsi lain yang dapat digunakan untuk membuat kurva suatu persamaan matematis adalah fungsi *curve* (). Berbeda dengan fungsi *plot* () yang perlu menspesifikasi objek pada sumbu x dan y, fungsi *curve* () hanya perlu menspesifikasi objek sumbu x saja.

```
curve(expr, from = NULL, to = NULL, n = 101, add = FALSE, type = "l",
xname = "x", xlab = xname, ylab = NULL, log = NULL, xlim = NULL, ...)
```

Contoh :

```
Curve(expr=x^2*exp(-x/2), from=0, to=10, add=TRUE)
```

3.5.6 Fungsi *Barplot* ()

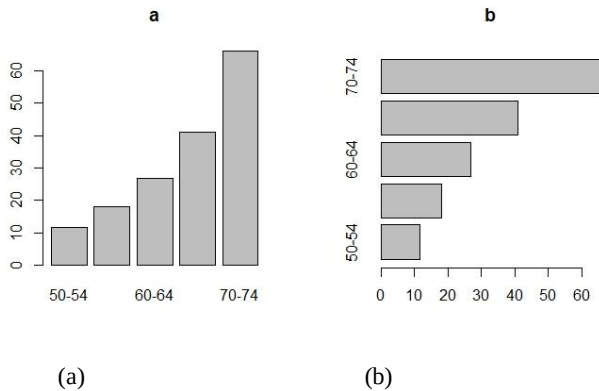
Barplot pada R dapat dibuat dengan menggunakan fungsi *barplot*(). Misalkan kita menggunakan data “*VADeaths*” dan membuat *barplot* untuk variabel ‘*Rural Male*’ *syntax* yang digunakan adalah sebagai berikut,

VADeaths

```
par(mfrow=c(1,2))
```

```
barplot(VADeaths[, "Rural Male"], main="a")
```

```
barplot(VADeaths[, "Rural Male"], main="b", horiz=TRUE)
```



Gambar 3.13 Bar chart Dataset *VADeaths* (Sumber: Penulis)

Keterangan :

1. Gambar 3.13 (a) merupakan *barchart* untuk tipe vertikal
2. Gambar 3.13 (b) merupakan *barchart* untuk tipe horizontal

Jika ingin melakukan kustomisasi pada *barplot* maka dapat digunakan *syntax* berikut,

```
barplot(VADeaths[, "Rural Male"],
```

```
# ubah warna outline menjadi steelblue
```

```
border="steelblue",
```

```
# ubah warna box
```

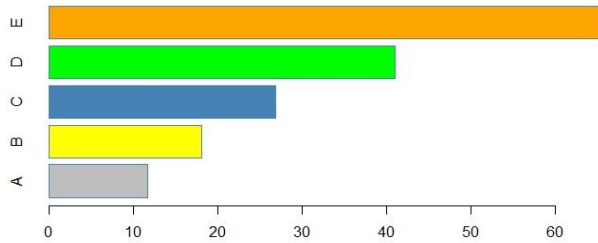
```
col= c("grey", "yellow", "steelblue", "green", "orange"),
```

```
# ubah nama grup dari A sampai E
```

```
names.arg = LETTERS[1:5],
```

```
# ubah orientasi menjadi horizontal
```

```
horiz=TRUE)
```



Gambar 3.14 Bar chart Dataset VADeaths Kustomisasi (Sumber: Penulis)

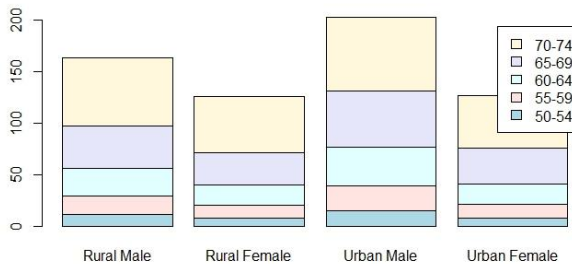
Atau bisa juga dilakukan *stacking*. Berikut *syntax* yang digunakan,

stacked

barplot(VADeaths,

col = c("lightblue", "mistyrose", "lightcyan", "lavender", "cornsilk"),

legend = rownames(VADeaths))



Gambar 3.15 Stacked Bar Chart Dataset VADeaths (Sumber: Penulis)

3.5.7 Fungsi *Density*

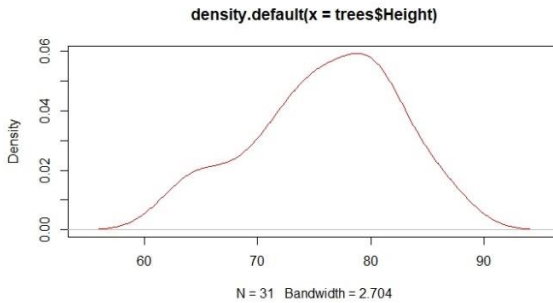
Density pada R dapat dibuat dengan menggunakan fungsi *density* (). Berbeda dengan fungsi *hist* (), fungsi ini tidak langsung menghasilkan grafik densitas. Fungsi *density* hanya menghitung kernel densitas pada data. Data yang telah dihitung selanjutnya diplotkan menggunakan fungsi *plot* (). Misalnya menggunakan *datasets tree*.

menghitung kernel density

dens <- density(trees\$Height)

plot densitas dengan outline merah

```
plot(dens,col="red")
```



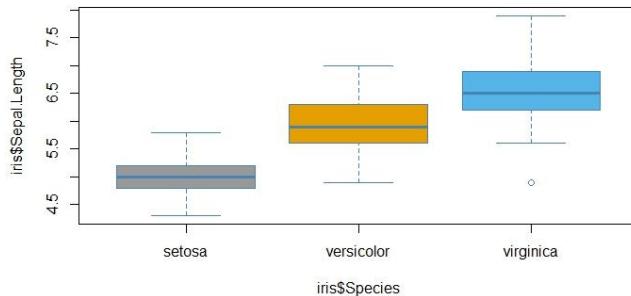
Gambar 3.16 Grafik *Density* Dataset Tree (Sumber: Penulis)

3.5.8 Fungsi *Boxplot*

Boxplot pada R dapat dibuat menggunakan fungsi `boxplot()`. Misalkan kita akan membuat *boxplot* variabel *sepal.length* pada dataset *iris*, `boxplot(iris$Sepal.Length)`.

Boxplot juga dapat dibuat berdasarkan variabel *factor*. Hal ini digunakan untuk melihat perbedaan distribusi data pada masing-masing grup. Misalkan akan dibuat *boxplot* berdasarkan variabel *species*, `boxplot(iris$Sepal.Length~iris$Species)`

Kita dapat juga mengubah warna pada *boxplot*, misalnya sebagai berikut `boxplot(iris$Sepal.Length~iris$Species)`
`boxplot(iris$Sepal.Length~iris$Species,`
 # ubah warna outline menjadi steelblue
 border = "steelblue",
 # ubah warna box berdasarkan grup
 col= c("#999999", "#E69F00", "#56B4E9"))



Gambar 3.17 Boxplot Dataset Iris (Sumber: Penulis)

3.5.9 Plot Dua Dimensi dan Tiga Dimensi

Untuk proyeksi 2 dimensi, fungsi yang digunakan adalah *image* () atau *contour* (). Sedangkan untuk proyeksi 3 dimensi dapat digunakan fungsi *persp* (). Sudut penglihatan dapat diatur melalui argumen *theta* (sudut) dan *phi* () yaitu rotasi. Berikut *syntax* untuk membuat plot 2 dimensi dan 3 dimensi,

#Plot Dua Dimensi

```
n <- 1:20
```

```
x <- sin(n)
```

```
y <- cos(n)*exp(-n/3)
```

```
z <- outer(x,y)
```

```
par(mar=c(3,3,1.5,1.5), mex=0.8, mgp=c(2,0.5,0), tcl=0.3)
```

```
par(mfrow=c(1,2))
```

```
# plot pertama
```

```
image(z, col=gray(1:10/10))
```

```
# plot kedua
```

```
contour(z)
```

#Plot Tiga Dimensi

```
par(mar=c(3,3,1.5,1.5), mex=0.8, mgp=c(2,0.5,0), tcl=0.3)
```

```
par(mfrow=c(1,2))
```

```
# plot pertama
```

```
persp(n,n,z, theta=45, phi=20)
```

```
# plot kedua
```

```
persp(n,n,z, theta=45, phi=20, shade=0.5)
```

3.5.10 Fungsi *Factor* untuk Data Kategorik

Misalkan data kategorik jenis kelamin terdiri dari pria dan wanita, maka pendefinisianannya adalah sebagai berikut.

```
gender <-factor(c("pria", "pria", "wanita","wanita","pria"))
```

Fungsi *factor* akan mengubah tipe data *object* “*gender*” menjadi *numeric*. Karena pada data kategori akan terdapat kategori maka dapat diketahui kategori data tersebut dengan fungsi *levels* ().

```
levels(gender)
```

R akan menentukan urutan kategori sesuai urutan alphabet, sehingga pria terlebih dahulu daripada wanita. Dengan fungsi *factor* (), R akan membuat korespondensi antara kelompok dengan suatu bilangan numerik, yang dapat diketahui dengan fungsi *str* ().

```
str(gender)
```

Contoh lain misalkan data kategori dari kelas siswa.

```
kelas<-factor(c(3,1,2,3,2))
```

```
levels(kelas)
```

```
str(kelas)
```

3.5.11 Grafik Data Kategorik

Untuk menampilkan grafik data kategori, digunakan diagram batang dan diagram lingkaran. Implementasi di R, menggunakan fungsi *barplot* () dan *pie* (). Keduanya akan menyajikan frekuensi setiap kategori. Perintah *barplot* dan *pie* dapat digunakan dengan mengkombinasikan *table* (). Sehingga untuk menampilkan *barplot* dan *pie* digunakan *syntax*:

```
gender <-factor(c("pria", "pria", "wanita","wanita","pria"))
```

```
gender
```

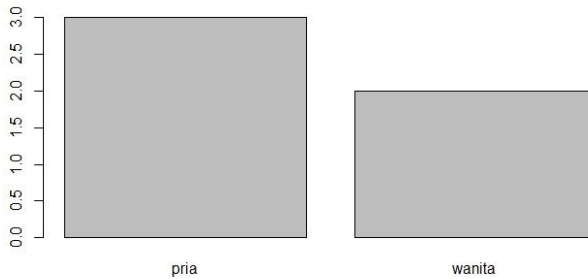
```
levels(gender)
```

```
str(gender)
```

```
tab.gender<-table(gender)
```

```
barplot(tab.gender)
```

```
pie(tab.gender)
```



Gambar 3.18 Grafik Data Kategorik (Sumber: Penulis)

Penjelasan detail terkait *barplot* dijelaskan di Bab V Visualisasi Data. Berikut *syntax* membuat *pie chart*:

`pie(x, labels, radius, main, col, clockwise)`

x : vektor mengandung data *numeric*

labels : deskripsi untuk pengelompokkan

radius : radius *pie chart*

main : membuat judul *pie chart*

col : memberikan warna pada *pie chart*

Contoh :

```
x <- c(21, 62, 10, 53)
```

```
labels <- c("London", "New York", "Singapore", "Mumbai")
```

```
# Memberikan nama pada pie chart
```

```
png(file = "city.png")
```

```
# Plot the chart.
```

```
pie(x, labels)
```

Jika ingin menambahkan warna, *legend*, dapat digunakan *syntax* berikut:

```
x <- c(21, 62, 10, 53)
```

```
labels <- c("London", "New York", "Singapore", "Mumbai")
```

```
piepercent <- round(100*x/sum(x), 1)
```

```
pie(x, labels = labels, main = "City pie chart", col = rainbow(length(x)))
```

```
legend("topright", c("London", "New York", "Singapore", "Mumbai"), cex = 0.8,
```

```
fill = rainbow(length(x)))
```




Gambar 3.19 *Pie Chart* Kota (Sumber: Penulis)

Kita juga bisa membuat 3D *pie chart* dengan *syntax* sebagai berikut:

```
library(plotrix)
x <- c(21, 62, 10, 53)
lbl <- c("London", "New York", "Singapore", "Mumbai")
pie3D(x, labels = lbl, explode = 0.1, main = "Pie Chart of Countries ")
```

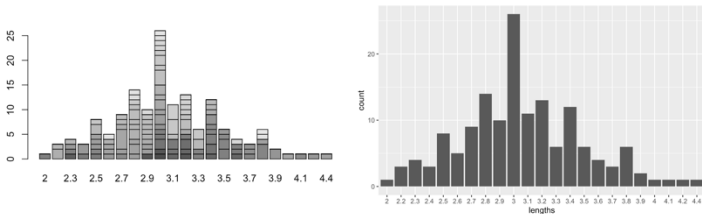
3.5.12 Fungsi ggplot

Secara umum, grafik merupakan pemetaan data ke dalam elemen visual yang memberikan ringkasan visual mengenai data tersebut. Salah satu perbedaan utama antara *R graphics* dan *ggplot2* terletak pada cara pemetaan ini dijelaskan. *R graphics* menggunakan fungsi khusus untuk membuat pemetaan tertentu, seperti fungsi untuk barplot, boxplot, dan grafik kurva. Di sisi lain, *ggplot2* mengambil pendekatan yang berbeda dengan terlebih dahulu mengidentifikasi data yang akan dipetakan dan objek geometris yang disebut “geom”. Fungsi estetika (*aes*) digunakan untuk menghubungkan data dengan objek visual.

Misalnya, dijelaskan rangkuman untuk menunjukkan perbedaan antara *R graphics* dan *ggplot*. Dalam *R barplot*, kita hanya perlu menentukan variabel hitung untuk tinggi batang, sedangkan dalam *ggplot*, variabel tersebut harus berada dalam bentuk data frame dan untuk objek grafis *geom_bar*, variabel tersebut harus berupa faktor atau karakter, serta harus dipetakan menggunakan fungsi estetika (*aes*).

```
#Versi R graphics
barplot(table(iris$Sepal.Length, iris$Sepal.Width))
#Versi ggplot
library(ggplot2)
```

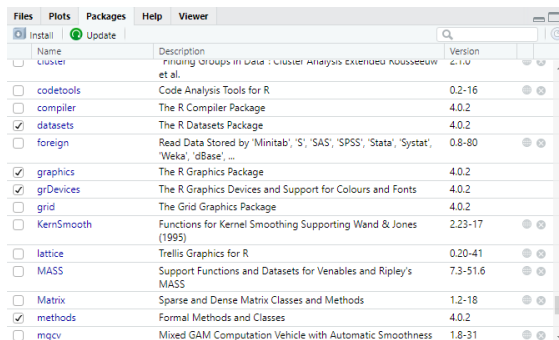
```
df <- data.frame(lengths = factor(iris$Sepal.Width))
ggplot(df, aes(lengths)) + geom_bar()
```



(a) (b)
Gambar 3.20 Plot R graphics (a) dan ggplot (b)

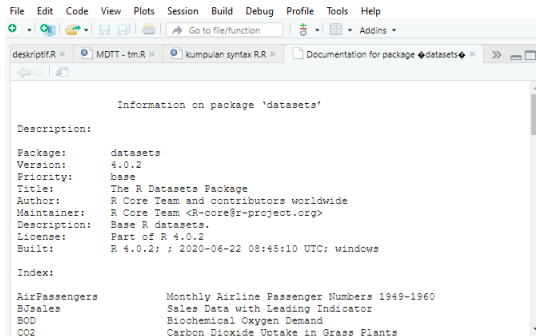
3.5.13 Mengaktifkan Data *Library Datasets*

Users dapat menggunakan beberapa dataset yang tersedia di library R untuk melakukan analisis data. Terdapat banyak dataset seperti “airquality” yaitu data pengukuran kualitas udara di NewYork, “mtcars” yaitu data ukuran kecepatan, lebar, panjang mobil, dan lain-lain.



Gambar 3.21 Mengaktifkan Data *Library Datasets* (Sumber: Penulis)

Untuk melihat beberapa *datasets* yang tersedia di R, dapat menggunakan `library(help="datasets")`



Gambar 3.22 *Datasets* yang Tersedia di R (Sumber: Penulis)

Jika ingin mengaktifkan data misalkan “cars”,
[attach\(cars\)](#)

Latihan Soal

1. Tentukan distribusi kontinu selain distribusi normal. Kemudian bangkitkan data sesuai dengan pilihan distribusi. Tampilkan:
 - a. Histogram
 - b. *Plot density*
 - c. Fungsi densitas $f(x)$
 - d. Fungsi distribusi $F(x)$
 - e. Tunjukkan luasan 5% sebelah kiri

Catatan : Semua gambar ditampilkan dalam satu jendela.

2. Definisikan pengelompokan mahasiswa berdasarkan jalur masuk (SNMPTN, SBMPTN, dan SELEKSI MANDIRI). Buatlah *bar chart* dan *pie chart* nya (2D dan 3D) serta lakukan modifikasi *barchart* dan *pie chart* tersebut dengan menambahkan warna, judul, dan lain-lain.

3.6 Komputasi Aljabar Linear

Setelah mempelajari komputasi aljabar linear. Mahasiswa diharapkan mampu menjelaskan dan menerapkan konsep-konsep aljabar linear dengan menggunakan paket program R.

3.6.1 Input Matriks dengan Perintah `cbind`

1. Input data matrik per kolom


```
c1 <- c(1, 2, 3)
c2 <- c(1, 4, 6)
c3 <- c(2, -3, -5)
```

2. Selanjutnya data yang sudah dibuat, digabungkan dengan perintah `cbind`
`A <- cbind(c1, c2, c3)`
3. Nama-nama kolom (`c1`, `c2`, dan `c3`) dapat dihilangkan dengan menggunakan perintah
`colnames(A) <- NULL`
`A`

3.6.2 Input Matriks dengan Perintah `rbind`

Hasil yang sama akan diperoleh dengan menggunakan fungsi `rbind`, yaitu sebagai berikut

```
A <- rbind(c(1, 1, 2), c(2, 4, -3), c(3, 6, -5))
```

3.6.3 Membuat Beberapa Jenis Matriks

1. Matriks `O` atau Matriks Nol merupakan matriks yang semua unsur-unsurnya nol. Sedangkan Matriks Satuan adalah matriks yang semua unsurnya 1 dan dilambangkan dengan `S`.
`O <- matrix(0, nrow=4, ncol=4)`
`S <- matrix(1, nrow=4, ncol=5)`
2. Matriks diagonal merupakan matriks segi yang semua unsurnya nol kecuali unsur-unsur pada diagonal utama
`diag(x=1, nrow, ncol)`
3. Matriks identitas merupakan matriks persegi yang elemen-elemen pada diagonal utamanya 1 dan elemen-elemen diluar diagonal utama bernilai 0
`I <- diag(c(1,1,1))`
4. Fungsi `upper.tri` dan `lower.tri` akan menghasilkan matriks segitiga dengan unsur-unsurnya *logical TRUE* pada segitiga atas (`upper.tri`) dan bawah (`lower.tri`)
`m2 <- matrix(1:20, 4, 5)`
`lower.tri(m2)`
`m2[lower.tri(m2)] <- 0`
`m2`

`m3 <- matrix(1:20, 4, 5)`
`upper.tri(m3)`
`m3[upper.tri(m3)] <- 0`
`m3`

5. Mengubah elemen pada matriks segitiga atas

```
n.row=dim(m2)[1]
n.col=dim(m2)[2]
m2.low=matrix(NA,n.row,n.col) kosong
for (i in 1:n.row){
  for (j in 1:n.col){
    m2.low[i,j]=m2[i,j]/rowSums(m2)[i]
  }
}
m2.low
```

6. Mengubah elemen pada matriks segitiga bawah

```
n.row.m3=dim(m3)[1]
n.col.m3=dim(m3)[2]
m3.upp=matrix(NA,n.row.m3,n.col.m3) #membuat matriks kosong
for (i in 1:n.row.m3){
  for (j in 1:n.col.m3){
    m3.upp[i,j]=m3[i,j]/colSums(m3)[j] #membagi nilai setiap elemen
    dengan jumlahan kolom nya
  }
}
m3.upp
```

3.6.4 Operasi-Operasi Pada Matriks

- a. Misalkan kita mendefinisikan matriks A dan matriks B

```
A <- matrix(c(2,3,-2,1,2,2),3,2)
print(A)
B <- matrix(c(1,4,-2,1,2,1),3,2)
print(B)
```

- b. Penjumlahan dan pengurangan Matriks

```
D=A+B
D
E=A-B
E
```

- c. Perkalian Matriks dengan Skalar

```
C=A*2
C
```

- d. Perkalian Matriks. Untuk melakukan perkalian matriks, aturan perkalian matriks harus dipenuhi

```
F <- matrix(c(2,-2,1,2,3,1),2,3)
```

```
G <- A%*%F
```

```
G
```

- e. Pembagian Matriks

```
A <- matrix(c(2,3,-2,1,2,2),3,2)
```

```
print(A)
```

```
B <- matrix(c(1,4,-2,1,2,1),3,2)
```

```
print(B)
```

```
H <- A/B
```

3.6.5 Fungsi Aljabar Linear Matriks

Misalkan terdapat matriks A

```
A <- matrix(data=c(3, 2, 3, 1, 4, 6, 2, 3, 5), nrow=3)
```

- a. *Transpose* Matriks : `At<-t(A)`

- b. Determinan Matriks : `det.A <- det(A)`

- c. *Invers* Matriks : `inv.A <- solve(A)`

- d. *Rank* Matriks : `R.A <- qr(A)`

- e. Eigen Matriks : `E.A <- eigen(A)`

- f. Kronecker : `kron.A<-kronecker(A,A)`

Contoh kasus pada Sistem Persamaan Linear

$$A = \begin{bmatrix} 3 & 1 & 2 \\ 2 & 4 & 3 \\ 3 & 6 & 5 \end{bmatrix}, \text{ kons} = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} \rightarrow Ax = \text{konst} \Rightarrow x = A^{-1} \text{konst}$$

```
x <- solve(A)%*%kons
```

```
x1 <- solve(A,kons)
```

3.6.6 Fungsi Aljabar Linear Vektor

```
x=c(1,2,3)
```

```
y=seq(2, 6, by = 2)
```

```
z=rep(2, 3)
```

```
#Penjumlahan x dan y dan cetak
```

```
print(x + y)
```

```
#Perkalian z x 2
```

```
print(4*z)
```

```
#perkalian x dan y
print(x*y)
#penjumlahan x dan z
print(x + z)
```

Latihan Soal

Dengan menggunakan komputasi aljabar linear, dapatkan persamaan regresi,

$$b = (X'X)^{-1}(X'Y)$$

Diketahui data,

$$X = \begin{bmatrix} 1 & 4 \\ 1 & 3 \\ 1 & 5 \\ 1 & 6 \\ 1 & 8 \end{bmatrix}, Y = \begin{bmatrix} 12 \\ 14 \\ 13 \\ 17 \\ 18 \end{bmatrix}$$

Buatlah persamaan regresi dari data yang tersedia, kemudian hitunglah *R-Square* nya.

3.7 Distribusi Peluang

Setelah mempelajari distribusi peluang, diharapkan mahasiswa mampu menghitung nilai peluang dari sebuah distribusi dan membangkitkan data berdasarkan distribusi tertentu dengan R.

3.7.1 Distribusi Peluang

Distribusi atau penyebaran peluang adalah penjelasan matematika akan sebuah kejadian acak dalam hal peluang. Salah satu yang terpenting adalah distribusi normal yang berbentuk lonceng simetris. R menyediakan menu untuk melakukan beberapa operasi standar yang berkaitan dengan fungsi distribusi peluang, yaitu:

1. Perhitungan nilai kuantil : q
2. Perhitungan nilai peluang : p
3. Pembuatan plot distribusi atau grafik densitas : d
4. Pembangkitan data atau random data : r

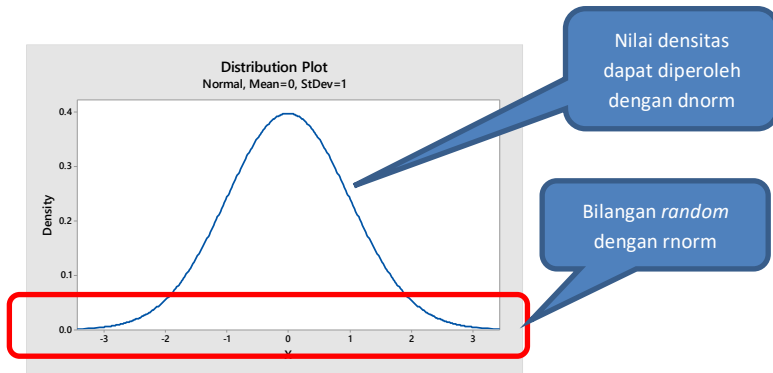
Untuk melihat distribusi apa saja yang terdapat di R dapat menggunakan: [help \(Distributions\)](#)

Distribution	Functions			
Beta	pbeta	qbeta	dbeta	rbeta
Binomial	pbinom	qbinom	dbinom	rbinom
Cauchy	pcauchy	qcauchy	dcauchy	rcauchy
Chi-Square	pchisq	qchisq	dchisq	rchisq
Exponential	pexp	qexp	dexp	rexp
F	pf	qf	df	rf
Gamma	pgamma	qgamma	dgamma	rgamma
Geometric	pgeom	qgeom	dgeom	rgeom

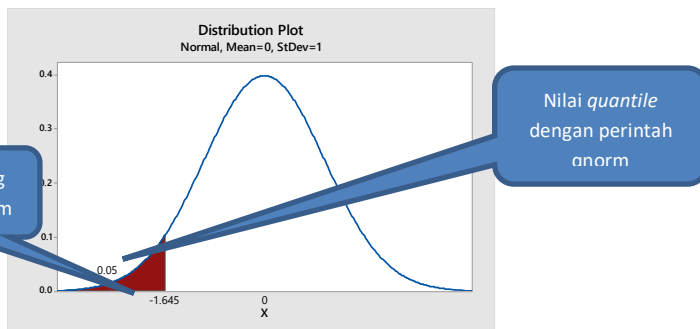
Gambar 3.23 Melihat Distribusi di R (Sumber: Penulis)

Distribution	Functions			
Hypergeometric	phyper	qhyper	dhyper	rhyper
Logistic	plogis	qlogis	dlogis	rlogis
Log Normal	plnorm	qlnorm	dlnorm	rlnorm
Negative Binomial	pnbinom	qnbinom	dnbinom	rnbinom
Normal	pnorm	qnorm	dnorm	rnorm
Poisson	ppois	qpois	dpois	rpois
Student t	pt	qt	dt	rt
Studentized Range	ptukey	qtukey	dtukey	rtukey
Uniform	punif	qunif	dunif	runif
Weibull	pweibull	qweibull	dweibull	rweibull
Wilcoxon Rank Sum Statistic	pwilcox	qwilcox	dwilcox	rwilcox
Wilcoxon Signed Rank Statistic	psignrank	qsignrank	dsignrank	rsignrank

Gambar 3.24 Melihat Distribusi di R (Lanjutan) (Sumber: Penulis)



Gambar 3.25 Penjelasan Plot Distribusi Normal (Sumber: Penulis)



Gambar 3.26 Penjelasan Plot Distribusi Normal (Lanjutan)
(Sumber: Penulis)

3.7.2 Distribusi Peluang Diskrit

Distribusi peluang diskrit yaitu distribusi dimana perubahnya secara teoritis tidak dapat menerima sembarang nilai diantara dua nilai yang diberikan. Sering lebih mudah bila semua peluang suatu peubah acak x dinyatakan dalam suatu rumus. Tetapi juga tidak menutup kemungkinan apabila distribusi diskrit dinyatakan dalam bentuk grafik atau pun dalam bentuk tabel. Contoh distribusi diskrit adalah distribusi Bionomial, distribusi Binomial Negatif, distribusi Geometrik, distribusi Hipergeometrik, distribusi Multinomial dan distribusi Poisson.

Distribusi Binomial

Misalkan saat melempar koin sebanyak n kali selalu menghasilkan gambar dan angka

1. Bangkitan Data

```
data.binom=rbinom(5, 20, 0.5)
```

2. Probability Density Function dan Plot

```
success=seq(0,50,by=1)
```

```
density.binom=dbinom(success,50,0.5)
```

```
plot(success, density.binom, xlab="x", ylab="Density",  
main=expression(paste("Binomial Distribution: ", p, " = 0.5, ",n,"=50,  
")), type="p")
```

3. Cumulative Density Function dan Plot

```
success=0:50
```

```
cdf.binom=pbinom(success,50,0.5)
```

```
plot(success, cdf.binom, xlab="x", ylab="CDF",  
main=expression(paste("Binomial Distribution: ", p, " = 0.5, ",n,"=50,  
")), type="p")
```

4. Invers CDF/Quantile

```
inv.cdf=qbinom(0.0033,50,0.5)
```

Contoh soal Distribusi Binomial

Sebuah koin dilempar 50 kali. Berapa peluang bahwa akan muncul tepat angka sebanyak 10 kali, jika peluang muncul nya angka 0,5.

```
dbinom(10,50,0.5,log=FALSE)
```

```
n=50
```

```
x=10
```

```
p=0.5
```

```
koin=dbinom(10,50,0.5,log = FALSE)
```

Kemungkinan mendapatkan 20 angka atau kurang, dari 50 lemparan koin dimana setiap percobaannya memiliki peluang sukses 0,5

```
pbinom(20,50,0.5,lower.tail = TRUE, log.p = FALSE)
```

3.7.3 Distribusi Peluang Kontinyu

Berbeda dengan variabel random diskrit, sebuah variabel random kontinyu adalah variabel yang dapat mencakup nilai pecahan maupun *range*/rentang nilai tertentu. Karena terdapat bilangan pecahan yang jumlahnya tidak terbatas, kita tidak dapat menuliskan semua nilai yang mungkin bersama dengan probabilitasnya masing-masing dalam bentuk tabel. Beberapa distribusi peluang kontinyu antara lain distribusi uniform, distribusi gamma, distribusi eksponensial, distribusi weibull, distribusi tipe beta, dan distribusi normal.

Distribusi Normal

1. Bangkitan Data

```
b=rnorm(30,1,1)
```

2. Probability Density Function dan Plot

```
sort.b=sort(b,decreasing = FALSE)
```

```
density.b=c(rep(NA,30))
```

```
for (i in 1:length(b)){
```

```
density.b[i]=dnorm(sort.b[i],1,1)
```

```
}
```

```
density.b
```

```
hist(a, probability = TRUE)
```

```
lines(density(a), col=2, lwd=2)
```

```
abline(v=0, col="blue")
```

```
hist(b, probability = TRUE)
```

```
lines(density(b), col=2, lwd=2)
```

```
abline(v=1, col="blue")
```

3. Cumulative Density Function dan Plot

```
cdf.b=c(rep(NA,100))
```

```
for (i in 1:length(b)){
```

```
cdf.b[i]=pnorm(sort.b[i],1,1)
```

```
}
```

```
cdf.b
```

```
plot(sort.b, cdf.b, xlab="x", ylab="Cumulative",
```

```
main=expression(paste("Normal Distribution: ", mu, " = 1, ", sigma, " =
1")),
type="l",xlim=c(min(sort.b),max(sort.b)),ylim=c(min(cdf.b),max(cdf.b)
))
```

4. Invers CDF/Quantile

```
quantil=qnorm(0.05,0,1) #invers CDF/nilai Z atau X pada tabel distribusi
normal standar
```

Contoh soal Distribusi Normal

Dari 200 mahasiswa yang ikut ujian diperoleh rata-rata 60 dan standar deviasi 10. Jika diasumsikan mengikuti kejadian distribusi normal, berapa persen yang mendapat nilai A jika nilai $A \geq 80$

```
A=1-pnorm(80,60,10,lower.tail = TRUE)
```

```
A=pnorm(80,60,10,lower.tail = FALSE) #lower.tail False untuk mencari
luasan >=
```

Latihan Soal

1. Peneliti ingin mengetahui seberapa besar kemungkinan setidaknya terdapat 3 bayi dari 20 rumah tangga kan menderita bronkitis jika peluang penyakit tersebut terjangkit pada satu rumah tangga adalah sebesar 0,05.
2. Anda diminta untuk menghitung probabilitas masa layanan lampu. Diketahui rata-rata layanan lampu produk adalah 750 jam dengan standar deviasi 80 dan diasumsikan mengikuti kejadian distribusi normal, berapa probabilitas lampu dengan masa layanan tepat 830 jam?
3. Bangkitkan dari beberapa distribusi diskrit (poisson dan geometri) dan distribusi kontinyu (eksponensial, weibull, erlang), buat plot pdf dan plot cdf, sertakan contoh kasus dari masing-masing distribusi tersebut.

3.8 Statistika Inferensia

Setelah mempelajari komputasi statistika inferensia, diharapkan mahasiswa mampu melakukan pengujian hipotesis satu populasi, dua populasi, dan lebih dari dua populasi baik *mean* maupun *varians*.

3.8.1 Pengujian Hipotesis Satu Populasi

Dalam pengujian hipotesis bisa digunakan sampel besar dan sampel kecil. Hipotesis yang digunakan dalam uji rata-rata satu populasi adalah sebagai berikut:

a. $H_0: \mu = \mu_0$

$$H_1: \mu \neq \mu_0$$

b. $H_0: \mu = \mu_0$

$$H_1: \mu > \mu_0$$

c. $H_0: \mu = \mu_0$

$$H_1: \mu < \mu_0$$

Statistik uji untuk sampel besar yaitu:

$$Z = \frac{\bar{Y} - \mu_0}{\sigma/\sqrt{n}}, \text{ jika varians populasi diketahui}$$

Kriteria penolakan:

a. Tolak H_0 jika $|Z_{hit}| > Z_{\alpha/2}$

b. Tolak H_0 jika $Z_{hit} > Z_{\alpha}$

c. Tolak H_0 jika $Z_{hit} < -Z_{\alpha}$

Statistik uji untuk sampel kecil yaitu:

$$t = \frac{\bar{Y} - \mu_0}{s/\sqrt{n}}$$

Kriteria penolakan:

a. Tolak H_0 jika $|t_{hit}| > t_{\alpha/2, n-1}$

b. Tolak H_0 jika $t_{hit} > t_{\alpha, n-1}$

c. Tolak H_0 jika $t_{hit} < -t_{\alpha, n-1}$

Implementasi R untuk Z test Satu Populasi:

1. Install package BSDA

```
library("BSDA")
```

2. Syntax Dasar

```
z.test(x, y, alternative='two.sided', mu=0, sigma.x=NULL,  
sigma.y=NULL, conf.level=.95)
```

```
#x          : nilai sampel pertama
```

```
#y          : nilai sampel kedua
```

#alternatif : hipotesis alternatif
 #mu : mean dibawah H0 atau perbedaan mean
 (jika kasus nya 2 sampel)
 #sigma.x : standar deviasi populasi dari sampel pertama
 #sigma.y : standar deviasi populasi dari sampel kedua
 #conf.level : tingkat kepercayaan

Contoh.

Misalkan nilai IQ pada populasi tertentu terdistribusi normal dengan $\mu = 100$ dan $\sigma = 15$. Seorang scientist ingin mengetahui apakah sebuah cara/tindakan baru mempengaruhi tingkat IQ. Sehingga dia merekrut 35 pasien untuuk menguji tindakan baru tersebut.

a. Input data IQ 35 Pasien

```
data = c(88, 92, 94, 94, 96, 97, 97, 97, 99, 99,
105, 109, 109, 109, 110, 112, 112, 113, 114, 115,
95,104,98,94,108,107,99,115,112,114,114,93,92,105,14)
```

1) Menghitung Z-test Satu Sampel

```
z.test(data, alternative='two.sided',mu=100, sigma.x=15, conf.level =
0.95)
```

2) Z Kritis

```
qnorm(0.05/2,lower.tail = TRUE)
qnorm(0.05/2,lower.tail = FALSE)
```

3) P-value

```
pnorm(-1.295,lower.tail = TRUE)+pnorm(1.295, lower.tail = FALSE)
2*pnorm(1.2959,lower.tail = FALSE)
2*pnorm(-1.2959,lower.tail = TRUE)
```

Untuk hipotesis alternative “lebih dari” dan “kurang dari”, maka alternative pada basic syntax diganti “greater” untuk lebih dari dan “less” untuk kurang dari.

Implementasi R untuk t test Satu Populasi:

1. Syntax Dasar

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu =
0, paired =FALSE, var.equal = FALSE, conf.level = 0.95)
```

a. Input R dataset

```
data(sleep)
sleep
```

- b. Menghitung t-test Satu Populasi
`t.test(x = sleep$extra, mu=0, alternative = "two.sided")`
- c. t Kritis
`qt(0.05/2,19,lower.tail = TRUE)`
`qt(0.05/2,19,lower.tail = FALSE)`
- d. P-value
`pt(-3.413,19,lower.tail = TRUE)+pt(3.413,19,lower.tail = FALSE)`
`2*pt(3.413,19,lower.tail = FALSE)`
`2*pt(-3.413,19,lower.tail = TRUE)`

3.8.2 Pengujian Hipotesis Dua Populasi

Dalam pengujian hipotesis bisa digunakan sampel besar dan sampel kecil. Hipotesis yang digunakan dalam uji rata-rata dua populasi adalah sebagai berikut:

- a. $H_0: \mu_1 = \mu_2$
 $H_1: \mu_1 \neq \mu_2$
- b. $H_0: \mu_1 = \mu_2$
 $H_1: \mu_1 > \mu_2$
- c. $H_0: \mu_1 = \mu_2$
 $H_1: \mu_1 < \mu_2$

Statistik uji untuk sampel besar yaitu:

- a. $Z = \frac{(\bar{Y}_1 - \bar{Y}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$, jika varians kedua populasi diketahui dan berbeda
- b. $Z = \frac{(\bar{Y}_1 - \bar{Y}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$, jika varians kedua populasi tidak diketahui dan berbeda
- c. $Z = \frac{(\bar{Y}_1 - \bar{Y}_2) - (\mu_1 - \mu_2)}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$, jika varians kedua populasi tidak diketahui dan sama

Kriteria penolakan:

- a. Tolak H_0 jika $|Z_{hit}| > Z_{\alpha/2}$
- b. Tolak H_0 jika $Z_{hit} > Z_{\alpha}$
- c. Tolak H_0 jika $Z_{hit} < -Z_{\alpha}$

Statistik uji untuk sampel kecil yaitu:

a. $t = \frac{(\bar{Y}_1 - \bar{Y}_2) - (\mu_1 - \mu_2)}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}$, jika varians kedua populasi berbeda

b. $t = \frac{(\bar{Y}_1 - \bar{Y}_2) - (\mu_1 - \mu_2)}{\sqrt{s_p^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}$, jika varians kedua populasi sama

Kriteria penolakan:

a. Tolak H_0 jika $|t_{hit}| > t_{\alpha/2, n-2}$

b. Tolak H_0 jika $t_{hit} > t_{\alpha, n-2}$

c. Tolak H_0 jika $t_{hit} < -t_{\alpha, n-2}$

Implementasi R untuk Z test Dua Populasi:

Contoh.

Misalkan tingkat IQ pada setiap individual di 2 kota berbeda diketahui berdistribusi normal dengan setiap populasi memiliki standar deviasi 15. Seorang scientist ingin mengetahui apakah terdapat perbedaan tingkat IQ di kota A dan kota B. Oleh karena itu dia memiliki 35 sampel dari setiap kota dan direcord tingkat IQ nya

1. Input Tingkat IQ untuk 35 Pasien di 2 Kota Berbeda

```
cityA = c(82, 84, 85, 89, 91, 91, 92, 94, 99, 99,
105, 109, 109, 109, 110, 112, 112, 113, 114, 114,
103, 104, 95, 96, 98, 93, 106, 109, 115, 98, 112, 153, 113, 97, 95)
cityB = c(90, 91, 91, 91, 95, 95, 99, 99, 108, 109,
109, 114, 115, 116, 117, 117, 128, 129, 130, 133,
129, 115, 98, 104, 113, 133, 118, 109, 114, 104, 115, 106, 114, 132, 112)
```

a. Menghitung Z-test 2 Populasi

```
z.test(x=cityA, y=cityB, mu=0, sigma.x=15, sigma.y=15)
```

b. Z Kritis

```
qnorm(0.05/2, lower.tail = TRUE)
```

```
qnorm(0.05/2, lower.tail = FALSE)
```

c. P-value

```
pnorm(-2.3267, lower.tail = TRUE) + pnorm(2.3267, lower.tail = FALSE)
```

```
2 * pnorm(2.3267, lower.tail = FALSE)
```

```
2 * pnorm(-2.3267, lower.tail = TRUE)
```


Untuk hipotesis alternative “lebih dari” dan “kurang dari”, maka alternative pada basic syntax diganti “greater” untuk lebih dari dan “less” untuk kurang dari.

Implementasi R untuk t test Dua Populasi:

1. Input R dataset
`data(sleep)`
`sleep`
- a. Memisahkan Groups 1 and 2
`sleep1 <- subset(sleep, group == 1)`
`sleep2 <- subset(sleep, group == 2)`
- b. Homogeneity variance test
`var.test(extra ~ group, data = sleep)`
- c. Variansi kedua populasi sama
`t.test(x = sleep2$extra, y = sleep1$extra, alternative = "two.sided",
paired=FALSE, var.equal = TRUE)`
- d. t Kritis
`qt(0.05/2,18,lower.tail = TRUE)`
`qt(0.05/2,18,lower.tail = FALSE)`
- e. P-value
`pt(-1.8608,18,lower.tail = TRUE)+pt(1.8608,18,lower.tail = FALSE)`
`2*pt(1.8608,18,lower.tail = FALSE)`
`2*pt(-1.8608,18,lower.tail = TRUE)`

3.8.3 Analysis of Variance (ANOVA) One Way

Analisis Varians satu jalur (ANOVA-one-way) merupakan teknik statistika parametrik yang digunakan untuk melakukan pengujian perbedaan beberapa kelompok rata-rata, dimana hanya terdapat satu variabel bebas atau independen yang dibagi dalam beberapa kelompok dan satu variabel terikat atau dependen. Hipotesis dalam ANOVA akan membandingkan rata-rata dari beberapa populasi yang diwakili oleh beberapa kelompok sampel secara bersama. Secara matematis :

$H_0 : \mu_1 = \mu_2 = \dots = \mu_k$ (tidak ada efek treatment)

$H_1 : \text{minimal ada 1 } \mu_j \neq \mu_k$ (minimal ada 1 efek treatment)

Hipotesis alternatif diatas merupakan hipotesis fleksibel, karena tidak menyebutkan μ mana yang berbeda dengan lainnya. Sehingga diperlukan

analisis lanjutan yang sering disebut *Post Hoc* untuk melihat treatment mana yang berbeda. Beberapa analisis *Post-Hoc* adalah *LSD*, *Tukey*, *S-N-K*, *Dunnet*, *Scheffe*.

Implementasi di R

Mengambil data dari directory

```
my_data <- read.delim(file.choose())  
my_data <- read.csv(file.choose())
```

Mengambil data dari library R

```
my_data <- PlantGrowth  
set.seed(1234)  
dplyr::sample_n(my_data, 10)
```

Menampilkan levels

```
levels(my_data$group)  
my_data$group <- ordered(my_data$group, levels = c("ctrl", "trt1", "trt2"))
```

Menghitung statistika deskriptif

```
library(dplyr)  
group_by(my_data, group) %>%  
  summarise(  
    count = n(),  
    mean = mean(weight, na.rm = TRUE),  
    sd = sd(weight, na.rm = TRUE)  
  )
```

Visualisasi Data dengan Bloxplot dan Meanplot

```
ggboxplot(my_data, x = "group", y = "weight",  
  color = "group", palette = c("#00AFBB", "#E7B800", "#FC4E07"),  
  order = c("ctrl", "trt1", "trt2"),  
  ylab = "Weight", xlab = "Treatment")
```

```
ggline(my_data, x = "group", y = "weight",  
      add = c("mean_se", "jitter"),  
      order = c("ctrl", "trt1", "trt2"),  
      ylab = "Weight", xlab = "Treatment")
```

Menghitung ANOVA one-way test

```
res.aov <- aov(weight ~ group, data = my_data)  
summary(res.aov)
```

Uji Perbandingan

```
TukeyHSD(res.aov)
```

Cek Asumsi Homogenitas variansi dan residual normal

```
library(car)  
leveneTest(weight ~ group, data = my_data)  
plot(res.aov, 2)
```

```
aov_residuals <- residuals(object = res.aov )  
shapiro.test(x = aov_residuals )
```

Jika asumsi tidak terpenuhi, maka dapat digunakan alternatif pengujian kelompok dengan Kruskal Wallis

```
kruskal.test(weight ~ group, data = my_data)
```

3.8.4 Analysis of Variance (ANOVA) Two Way

Jika pada ANOVA satu arah kita bisa mengetahui ada atau tidaknya perbedaan beberapa variabel bebas dengan sebuah variabel terikat dan masing-masing variabel tidak mempunyai jenjang, maka dalam ANOVA dua arah kita ingin mengetahui ada atau tidaknya perbedaan beberapa variabel bebas dengan sebuah variabel terikatnya dan masing-masing variabel mempunyai dua jenjang atau lebih. Ada dua jenis ANOVA dua arah yaitu tanpa interaksi dan dengan interaksi. ANOVA dua arah tanpa interaksi yaitu pengujian hipotesis beda tiga rata-rata atau lebih dengan dua faktor

yang berpengaruh dan interaksi antara kedua faktor tersebut ditiadakan. Sedangkan ANOVA dua arah dengan interaksi adalah pengujian beda tiga rata-rata atau lebih dengan dua faktor yang berpengaruh dan pengaruh interaksi antara kedua faktor tersebut diperhitungkan.

Contoh.

Pada data “ToothGrowth” di library R, terdapat variabel terikat yaitu panjang gigi (len). Sedangkan variabel bebas nya adalah jenis suplemen (Supp) yaitu VC dan OJ dan Dosis (dose) yaitu D0.5, D1, dan D2. Peneliti ingin mengetahui apakah terdapat pengaruh jenis suplemen terhadap panjang gigi dan apakah ada pengaruh dosis terhadap panjang gigi. Kemudian apakah terdapat pengaruh interaksi kedua variabel bebas ini terhadap panjang gigi (len).

Misalkan α adalah faktor pertama (supp), β adalah faktor kedua (dose), dan $(\alpha\beta)$ adalah faktor interaksi supp dan dose

Hipotesis yang digunakan adalah sebagai berikut.

Hipotesis 1 untuk faktor pertama (supp)

$$H'_0 = \alpha_1 = \alpha_2 = 0$$

$$H'_1 = \text{minimal ada satu } \alpha_i \neq 0$$

Hipotesis 2 untuk faktor kedua (dose)

$$H''_0 = \beta_1 = \beta_2 = \beta_3 = 0$$

$$H''_1 = \text{minimal ada satu } \beta_j \neq 0$$

Hipotesis 3 untuk faktor interaksi (supp dan dose)

$$H'''_0 = \alpha\beta_{11} = \alpha\beta_{12} = \alpha\beta_{13} = \dots = \alpha\beta_{23} = 0$$

$$H'''_1 = \text{minimal ada satu } \alpha\beta_{ij} \neq 0$$

Hipotesis alternatif diatas merupakan hipotesis fleksibel, karena tidak menyebutkan μ mana yang berbeda dengan lainnya. Sehingga diperlukan analisis lanjutan yang sering disebut *Post Hoc* untuk melihat treatment mana

yang berbeda. Beberapa analisis *Post-Hoc* adalah *LSD*, *Tukey*, *S-N-K*, *Dunnet*, *Scheffe*.

Implementasi di R

Mengambil data dari library R

```
my_data <- ToothGrowth
```

Convert levels menjadi factor

```
my_data$dose <- factor(my_data$dose,  
                        levels = c(0.5, 1, 2),  
                        labels = c("D0.5", "D1", "D2"))  
head(my_data)
```

Menghitung statistika deskriptif

```
require("dplyr")  
group_by(my_data, supp, dose) %>%  
  summarise(  
    count = n(),  
    mean = mean(len, na.rm = TRUE),  
    sd = sd(len, na.rm = TRUE)  
  )
```

Visualisasi Data dengan Bloxplot dan Meanplot

```
library("ggpubr")  
ggboxplot(my_data, x = "dose", y = "len", color = "supp",  
          palette = c("#00AFBB", "#E7B800"))
```

Menghitung ANOVA two-way test

```
res.aov2 <- aov(len ~ supp + dose, data = my_data)  
summary(res.aov2)
```

Menghitung ANOVA two-way test dengan Faktor Interaksi

```
res.aov3 <- aov(len ~ supp * dose, data = my_data)  
res.aov4 <- aov(len ~ supp + dose + supp:dose, data = my_data)  
summary(res.aov3)
```

Uji Perbandingan

```
TukeyHSD(res.aov3, which = "dose")
```

```
TukeyHSD(res.aov3, which = "supp") #we don't need to do this, because  
only 2 levels
```

Cek Asumsi Homogenitas variansi dan residual normal

```
library(car)
```

```
leveneTest(len ~ supp*dose, data = my_data)
```

```
aov_residuals <- residuals(object = res.aov3)
```

```
shapiro.test(x = aov_residuals )
```

3.8.5 Regresi Linear Berganda

Analisis Regresi adalah suatu analisis yang bertujuan untuk menunjukkan hubungan matematis antara variabel respon dan variabel prediktor. Model Regresi Linear Berganda adalah sebagai berikut.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_p X_p + \varepsilon$$

$$Y = X\beta + \varepsilon$$

Estimasi Parameter :

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

$$\text{var}(\hat{\beta}) = (X^T X)^{-1} \sigma^2$$

Asumsi dalam Analisis Regresi

1. Residual Identik
2. Residual Independen
3. Residual Berdistribusi Normal
4. Variabel-Variabel Prediktor tidak terjadi kasus Multikolinearitas.

Implementasi di R

1. Install Packages

```
install.packages("lmtest")
```

```
install.packages("lmtest", lib="C:/Program Files/R/R-3.6.3/library")
```

```
library(lmtest)
```

2. Import Data
`datakemiskinan<-read_excel("C:/Users/Muhammad
Alifian/Downloads/datakemiskinan.xlsx")`
3. Struktur Data
`str(datakemiskinan)`
4. Statistik Deskriptif
`summary(datakemiskinan)`
5. Uji Ramsey's (cek linearitas)
`library(lmtest)`
`resettest(Kemis~JB+Rls+Tpt,power=2,type="regressor",data=datakemiskin
an)`
6. Regresi Linear Berganda
`model_seluruh<-lm(Kemis~JB+Rls+Tpt,data=datakemiskinan)`
`summary(model_seluruh)`
7. uji asumsi Normality dan Heteroskedastisitas
`par(mfrow=c(2,2))`
`plot(model_seluruh)`
8. Uji Multikolinearitas
`library(car)`
`vif(model_seluruh)`
9. Uji Autokorelasi
`library(lmtest)`
`dwtest(model_seluruh)`

Latihan Soal

1. Cari Contoh Kasus yang dapat diselesaikan dengan ANOVA one-way, ANOVA two-way, dan Regresi Linear Berganda
2. Lakukan analisis pada kasus tersebut dengan metode yang relevan (ANOVA one-way, ANOVA two-way, dan Regresi Linear Berganda)

3.9 Optimasi

Optimasi adalah suatu proses untuk mencari kondisi yang optimum atau kondisi paling optimal. Optimasi dapat berupa nilai maksimum ataupun nilai minimum tergantung pada penggunaannya. Jika berkaitan dengan masalah keuntungan, maka keadaan optimum adalah mencari nilai maksimum. Jika berkaitan dengan masalah pengeluaran, maka keadaan optimum adalah mencari nilai pengeluaran minimum. Fungsi yang akan

diminimumkan atau dimaksimumkan disebut juga fungsi objektif (*objective function*).

3.9.1 Fungsi Optimasi Univariat (*optimize & Optimise*)

Fungsi **optimize** dan **optimise** digunakan untuk mencari nilai maksimum dan nilai minimum pada fungsi pada interval tertentu (*lower and upper*). Fungsi **optimize** digunakan ketika persamaan tersebut hanya melibatkan satu peubah, sedangkan fungsi **optim** digunakan ketika persamaan tersebut melibatkan lebih dari satu peubah.

Implementasi pada R untuk Fungsi *Optimize*

Diketahui fungsi $f(x) = 2x^2 + 3x - 7$. Buatlah kurva persamaan dan tentukan nilai minimum fungsi tersebut pada interval bawah 0 dan interval atas 0.5.

```
f6<-function(x) (2*x^2) + 3*x - 7
curve(f6,c(0,1)) #c(0,1) merupakan interval kurva
xmin<-optimize(f6,c(0,1),tol=0.0001) #c(0,0.5) merupakan interval dalam
mencari titik optimum
xmin
```

Implementasi pada R untuk Fungsi *Optimise*

```
fr <- function (x){ # tetap dituliskan dalam sebuah vektor, akan diduga x
  x1<- x[1]
  x2 <- x[2]
  100 * (x2-x1^2)^2 + (1-x1)^2} # ini adalah nilai fungsi objetivenya
optim (c(-1.2,1),fr)# argumen pertama adalah nilai inisial, karena menduga
x vektor berukuran 2 maka dimasukkan nilai inisialnya
```

3.9.2 Fungsi Optimasi Akar Univariat (*uniroot*)

R menyediakan fungsi **uniroot()** untuk mencari akar persamaan suatu fungsi pada rentang spesifik. Fungsi ini menggunakan metode Brent yaitu kombinasi antara *root bracketing*, biseksi, dan interpolasi invers kuadrat.

Implementasi pada R untuk Fungsi *Uniroot*

Diketahui fungsi $f(x) = 2x^2 - 4x - 10$. Buatlah kurva persamaan dan tentukan akar persamaan fungsi tersebut pada interval bawah -4 dan interval atas 4.

```
func2 <- function(x) {  
  2*x^2 - 4*x - 10  
}  
curve(func2, xlim=c(-4,4), col='blue', lwd=2, lty=2, ylab='f(x)')  
abline(h=0)  
abline(v=0)  
uniroot(func2, c(-4,0))  
uniroot(func2, c(4,0))
```

3.9.3 Metode Bisection (*BFfzero*)

Metode bagi dua (*bisection method*) adalah pendekatan untuk menemukan akar dari fungsi kontinu $f(x)$ pada interval $[a, b]$. Metode ini mengambil keuntungan dari akibat wajar dari teorema nilai antara yang disebut teorema **Bolzano** yang menyatakan bahwa nilai $f(a)$ dan $f(b)$ memiliki tanda yang berlawanan, interval harus mengandung setidaknya satu akar. Secara matematis interval akan dibagi menjadi setengah.

$$c = \frac{(a + b)}{2}$$

Fungsi dievaluasi pada c , lokasi akar ditentukan dari subinterval yang terletak di dalam iterasi berikutnya (c diganti dengan a atau b). Proses ini diulangi sampai presisi yang diinginkan. Berikut Algoritma Bisection

Implementasi pada R untuk Metode Bisection (*BFfzero*)

Misalkan diketahui fungsi $f(x) = x^3 - 3x - 5$. Buat Kurva pada interval $[2,3]$, dan tentukan akar persamaan dengan metode bisection.

```
func3 <- function(x) {
  x^3 - 2* x - 5
}
curve(func3, xlim=c(-3,3), col='blue', lwd=1.5, lty=2, ylab='f(x)')
abline(h=0)
abline(v=0)
library (NLRoot)
BFfzero (func3, 2,3)
```

3.9.4 Fungsi Optimasi Advanced (*nlminb*)

Fungsi *nlminb* pada *software* R merupakan fungsi optimasi *unconstrained* dan *box-constrained* menggunakan *PORT routines* yang ditulis oleh Gay (1990) di Bell Labs, selanjutnya selanjutnya diimpor ke *software* R oleh Douglas Bates. Fungsi *nlminb* menggunakan metode optimasi Newton (Nash dan Varadhan, 2011). Beberapa optimisasi *PORT routines* memiliki struktur dan *user interface* yang umum. Semua kontrol, toleransi dan semua penyimpanan awal yang digunakan oleh *routines* terkandung dalam dua *array*: *IV* untuk solusi yang bernilai *integer*, *V* untuk solusi yang bernilai *floating-point* (*REAL* dalam versi presisi tunggal, *DOUBLE PRECISION* dalam versi presisi ganda).

Diberikan suatu fungsi f dari sebanyak p variabel, optimasi *routines* mencoba untuk menemukan p -vektor $\tilde{\phi}$ yang meminimumkan fungsi $f(\tilde{\phi})$. Berbagai macam batasan bisa dilakukan pada $\tilde{\phi}$, yaitu

- tidak ada batasan,
- batasan sederhana $\tilde{b}\tilde{\phi} \leq \tilde{\phi} \leq \tilde{b}\tilde{\phi}$, dimana $\tilde{b}\tilde{\phi}$ dan $\tilde{b}\tilde{\phi}$ adalah vektor, dan
- batasan linier umum $\tilde{b}\tilde{c} \leq \mathbf{C}\tilde{\phi} \leq \tilde{b}\tilde{c}$, dimana $\tilde{b}\tilde{c}$ dan $\tilde{b}\tilde{c}$ merupakan vektor dan $\mathbf{C}$ adalah matriks.

Gradien f pada $\tilde{\phi}$ (vektor turunan parsial pertama dari f) dinotasikan oleh $\nabla f(\tilde{\phi})$, dan Hessian dari f pada $\tilde{\phi}$ (matriks dari turunan parsial kedua dari f) dinotasikan oleh $\nabla^2 f(\tilde{\phi})$.

Optimasi *routines* memiliki dua tingkat, yaitu *forward-communication routine* dan *reverse-communication iteration driver*. *Forward-communication routine* mempelajari tentang $f(\tilde{\phi})$ dengan cara konvensional. Jika diberikan *subroutines*, maka *forward-communication routine* dapat melakukan panggilan untuk menghitung $f(\tilde{\phi})$. Di sisi lain, *reverse-communication driver* kembali melakukan pemanggilan ketika

reverse-communication driver perlu mengetahui $f(\tilde{\phi})$ pada $\tilde{\phi}$ yang baru. Pemanggilan *routines* harus menghitung informasi yang diperlukan (misalnya $f(\tilde{\phi})$ itu sendiri, atau untuk beberapa *routines* regresi, vektor residual) dan memanggil *reverse-communication driver* kembali. Biasanya lebih mudah untuk menggunakan optimasi *forward-communication routine*, tapi kadang-kadang lebih mudah untuk memanggil optimasi *reverse-communication driver*, misalnya ketika penulisan sebuah *subroutine* yang menghitung $f(\tilde{\phi})$ sulit.

Kedua versi *forward-communication* dan *reverse-communication* dari optimasi *routines* menerima IV dan V sebagai parameter. Parameter untuk *reverse communication drivers* juga termasuk $\tilde{\phi}$, salah satu dari nilai $f(\tilde{\phi})$ dan mungkin $\nabla f(\tilde{\phi})$ atau informasi yang cukup untuk menghitung nilai-nilai ini (untuk *routines* regresi). Driver kembali dengan $IV(1) = 1$ ketika menginginkan memiliki f yang dievaluasi pada $\tilde{\phi}$ saat ini dan dengan $IV(1) = 2$ ketika menginginkan ∇f yang dievaluasi. Beberapa *drivers* memiliki kemungkinan lain untuk kembali, seperti pada $IV(1) = -1$ atau -2 .

Fungsi `nlinmb` pada program R adalah sebagai berikut

`nlinmb(start, objective, gradient = NULL, hessian = NULL, ..., scale = 1, control = list(), lower = -Inf, upper = Inf)`

Arguments

1. **Start** : *numeric vector, initial values for the parameters to be optimized.*
2. **Objective** : *Function to be minimized. Must return a scalar value. The first argument to objective is the vector of parameters to be optimized, whose initial values are supplied through start. Further arguments (fixed during the course of the optimization) to objective may be specified as well (see ...).*
3. **Gradient** : *Optional function that takes the same arguments as objective and evaluates the gradient of objective at its first argument. Must return a vector as long as start.*
4. **Hessian** : *Optional function that takes the same arguments as objective and evaluates the hessian of objective at its first argument. Must return a square matrix of order length(start). Only the lower triangle is used.*
5. **Scale** : *See PORT documentation (or leave alone).*
6. **Control** : *A list of control parameters.*

lower, upper vectors of lower and upper bounds, replicated to be as long as start. If unspecified, all parameters are assumed to be unconstrained.

Contoh Aplikasi Fungsi nlminb

```
x <- rnbino(100, mu = 10, size = 10)
hdev <- function(par)-sum(dnbino(x, mu = par[1], size = par[2], log = TRUE))
```

```
nlminb(c(9, 12), hdev)
```

```
nlminb(c(20, 20), hdev, lower = 0, upper = Inf)
```

```
nlminb(c(20, 20), hdev, lower = 0.001, upper = Inf)
```

1. Meminimumkan fungsi *sum of squares* dengan solusi y diketahui

```
sumsq <- function( x, y) {sum((x-y)^2)}
```

```
y <- rep(1,5)
```

```
x0 <- rnorm(length(y))
```

```
nlminb(start = x0, sumsq, y = y)
```

2. Meminimumkan fungsi *sum of squares* dengan menggunakan batas dengan y yang memiliki beberapa komponen diluar batas

```
y <- c( 0, 2, 0, -2, 0)
```

```
nlminb(start = x0, sumsq, lower = -1, upper = 1, y = y)
```

2. Meminimumkan fungsi *sum of squares* dengan menggunakan batas dengan y yang memiliki beberapa komponen diluar batas dan menggunakan gradient

```
sumsq.g <- function(x, y) 2*(x-y)
```

```
nlminb(start = x0, sumsq, sumsq.g,
```

```
lower = -1, upper = 1, y = y)
```

3.9.5 Fungsi Optimasi Regresi dengan *Least Square*

Least Square merupakan salah satu teknik estimasi parameter regresi dengan meminimumkan jumlah kuadrat error.

Misalkan diketahui variabel prediktor (x) yaitu 1,2,3,4,5,6 dan variabel respon (y) yaitu 1,3,5,6,8,12. Lakukan estimasi parameter dengan *least square*

```
data = data.frame(x=c(1,2,3,4,5,6),y=c(1,3,5,6,8,12))
SSE<-function(data,b){
  with(data, sum((b[1]+b[2]*x-y)^2))
}
hasilLS = optim(par=c(1,1), fn = SSE, data = data)
plot(data)
abline(hasilLS$par, col=4)
```

Latihan Soal

1. Diketahui persamaan $f(x) = 3 - 2 \sin(x) \cos(x)$. Buatlah kurva pada interval bawah 0 dan interval atas 4π . Tentukan titik minimum/maksimum pada interval $[4,10]$.
2. Diketahui persamaan $f(x) = x^3 - 2x - 7$. Buatlah kurva dan tentukan akar persamaan pada interval $[-5,5]$.
3. Diketahui persamaan $f(x) = x^3 - 5x - 10$. Buatlah kurva dan tentukan akar persamaan pada interval $[3,7]$ dengan metode bisection.
4. Carilah contoh kasus yang bisa diselesaikan dengan regresi linear, kemudian lakukan optimasi *least square* untuk mendapatkan estimasi parameter.

3.10 Perapihan Data (*Data Tidying*)

Setelah mempelajari topik ini, mahasiswa diharapkan mampu melakukan perapihan data agar lebih mudah untuk dimanipulasi dan divisualisasikan. *Package* yang digunakan adalah *tidyr*.

Materi yang akan dipelajari pada topik ini adalah membuat data frame/set data, merubah struktur data (*reshape data*), Mengatasi *Missing value* (*handling missing value*), memisahkan atau menggabungkan sel (*split and combine cell*).

3.10.1 Membuat Data Tibble

1. Berdasarkan kolom

```
data = tibble(x=1:3, y=c("a","b","c"))
```

Berdasarkan baris

```
data_row = tribble(~x, ~y, 1, "a", 2, "b", 3, "c")
```

2. Merubah format data frame ke format tibble

misalkan data frame nya adalah `dat`, maka untuk mengganti format menjadi tibble digunakan fungsi

```
data_tib = as_tibble(dat)
```

3. Cek apakah data tersebut dengan format tibble

```
is.tibble (data_tib)
```

3.10.2 Merubah Struktur Data (*Reshape Data*)

1. Menggunakan `gather()`

```
gather(data, key, value,...,na.rm=FALSE, convert=FALSE,factor_key  
= FALSE)
```

contoh,

```
Table = tibble(country=c("A","B","C"), "1999"=c("0.7K","37K","212K"),  
"2000"=c("2K","80K","213K"))
```

```
gather(Table,'1999','2000',key = "year", value="cases")
```

			country year cases		
			<chr>	<chr>	<chr>
country	1999	2000	1 A	1999	0.7K
			2 B	1999	37K
			3 C	1999	212K
<chr>	<chr>	<chr>	4 A	2000	2K
1 A	0.7K	2K	5 B	2000	80K
2 B	37K	80K	6 C	2000	213K
3 C	212K	213K			

2. Menggunakan `spread()`

```
spread(data, key, value, fill=NA,convert=FALSE,drop  
=TRUE,sep=NULL)
```

contoh,

```
Table2 = tibble(country=c(rep("A",4),rep("B",4),rep("C",4)),  
year=c('1999','1999','2000','2000','1999','1999','2000','2000','1999','1999','2000','2000'), type
```

```
=c('cases','pop','cases','pop','cases','pop','cases','pop','cases','pop','cases','pop')  
)
```

```
count=c('0.7K','19M','2K','20M','37K','172M','80K','174M','212K','1T','213K','1T'))
```

```
spread(Table2,type,count)
```

	country	year	type	count		country	year	cases	pop
	<chr>	<chr>	<chr>	<chr>		<chr>	<chr>	<chr>	<chr>
1	A	1999	cases	0.7K		1	A	1999	0.7K 19M
2	A	1999	pop	19M		2	A	2000	2K 20M
3	A	2000	cases	2K		3	B	1999	37K 172M
4	A	2000	pop	20M		4	B	2000	80K 174M
5	B	1999	cases	37K		5	C	1999	212K 1T
6	B	1999	pop	172M		6	C	2000	213K 1T
7	B	2000	cases	80K					
8	B	2000	pop	174M					
9	C	1999	cases	212K					
10	C	1999	pop	1T					
11	C	2000	cases	213K					
12	C	2000	pop	1T					

Gambar 3.27 Output Spread (Sumber: Penulis)

3.10.3 Memisahkan dan Menggabungkan Cells

1. Menggunakan *Separate*

`separate(data,col,into,sep = "[^:alnum:]]+",remove=TRUE, convert=FALSE, extra="warn",fill="warn",...)`

Contoh,

```
Table3 = tibble(country=c(rep("A",2),rep("B",2),rep("C",2)),
year=c("1999","2000","1999","2000","1999","2000"),rate=c("0.7K/19M",
"2K/20M","37K/172M","80K/174M","212K/1T","213K/1T"))
separate(Table3,rate,sep="/",into=c("cases","pop"))
```

	country	year	rate		country	year	cases	pop
	<chr>	<chr>	<chr>		<chr>	<chr>	<chr>	<chr>
1	A	1999	0.7K/19M		1	A	1999	0.7K 19M
2	A	2000	2K/20M		2	A	2000	2K 20M
3	B	1999	37K/172M		3	B	1999	37K 172M
4	B	2000	80K/174M		4	B	2000	80K 174M
5	C	1999	212K/1T		5	C	1999	212K 1T
6	C	2000	213K/1T		6	C	2000	213K 1T

Gambar 3.28 Output Separate (Sumber: Penulis)

2. Menggunakan *Separate_rows*

`separate_rows(Table3,rate,sep="/")`

	country	year	rate
	<chr>	<chr>	<chr>
1	A	1999	0.7K
2	A	1999	19M
3	A	2000	2K
4	A	2000	20M
5	B	1999	37K
6	B	1999	172M
7	B	2000	80K
8	B	2000	174M
9	C	1999	212K
10	C	1999	1T
11	C	2000	213K
12	C	2000	1T

Gambar 3.29 Output *Separate_rows* (Sumber: Penulis)

3. Menggunakan *Unite*

`Unite(data,col,...,sep="_",remove=TRUE)`

Contoh,

`Table4=tibble(country=c("Afghan","Afghan","Brazil","Brazil","China","China"),century=c("19","20","19","20","19","20"),year=c("99","00","99","00","99","00"))`

`unite(Table4,century,year,col="year",sep="")`

	country	century	year
	<chr>	<chr>	<chr>
1	Afghan	19	99
2	Afghan	20	00
3	Brazil	19	99
4	Brazil	20	00
5	china	19	99
6	china	20	00

Gambar 3.30 Output *Unite* (Sumber: Penulis)

3.10.4 Cek dan Handling Missing Value

1. Cek *Missing Value*

Untuk cek *missing value*, dapat dilakukan dengan dua cara yaitu dengan `is.na()` atau `!is.na()`. Jika menggunakan `is.na()` maka *default* nya adalah *FALSE*. Jika menggunakan `!is.na()` maka *default* nya adalah *TRUE*.

Contoh,

`as_tibble(!is.na(mtcars))`

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>
1 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
2 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
3 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
4 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
5 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
6 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
7 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
8 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
9 TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
10 TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
# ... with 22 more rows, and 3 more variables:
#   am <lgl>, gear <lgl>, carb <lgl>
```

Gambar 3.31 Output Cek Missing Value (1) (Sumber: Penulis)

`as_tibble(is.na(mtcars))`

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl> <lgl>
1 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
2 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
3 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
4 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
5 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
6 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
7 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
8 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
9 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
10 FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
# ... with 22 more rows, and 3 more variables:
#   am <lgl>, gear <lgl>, carb <lgl>
```

Gambar 3.32 Output Cek Missing Value (2) (Sumber: Penulis)

2. *Handling Missing Value*

Handling missing value dapat dilakukan dengan 3 cara yaitu

`drop_na(data,...)`

menghapus row *missing value*.

Contoh,

`mv = tibble(X1=c("A","B","C","D","E"),X2=c(1,NA,NA,3,NA))`

`drop_na(mv,X2)`

```
# A tibble: 5 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 B      NA
3 C      NA
4 D      3
5 E      NA
```

→

```
# A tibble: 2 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 D      3
```

Gambar 3.33 Output Handling Missing Value (1) (Sumber: Penulis)

`fill(data,...,direction=c("down","up"))`

mengisi *missing value* dari nilai sebelum *missing value* tersebut

`fill(mv,X2)`

```
# A tibble: 5 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 B      NA
3 C      NA
4 D      3
5 E      NA
```

→

```
# A tibble: 5 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 B      1
3 C      1
4 D      3
5 E      3
```

Gambar 3.34 Output Handling Missing Value (2) (Sumber: Penulis)

`replace_na(data, replace = list(),...)`

mengisi *missing value* berdasarkan nilai tertentu

`replace_na(mv,list(X2=2))`

```
# A tibble: 5 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 B      NA
3 C      NA
4 D      3
5 E      NA
```

→

```
# A tibble: 5 x 2
  x1      x2
<chr> <dbl>
1 A      1
2 B      2
3 C      2
4 D      3
5 E      2
```

Gambar 3.35 Output Handling Missing Value (3) (Sumber: Penulis)

3.11 Manipulasi Data (*Data Manipulation*)

Setelah mempelajari topik ini, mahasiswa diharapkan mampu melakukan manipulasi data untuk memudahkan *user* dalam melakukan

analisis berdasarkan data yang diperlukan. *Package* yang digunakan adalah *dplyr*.

Materi yang akan dipelajari pada topik ini adalah manipulasi baris/*cases*, manipulasi kolom/*variabel*, mengkombinasikan tabel, *aggregating* untuk mendapatkan nilai tertentu, dan menyimpulkan berdasarkan baris/kolom.

3.11.1 Manipulasi Baris/*Cases*

1. Mengambil beberapa *cases* berdasarkan nilai tertentu

`filter(data,...,preserve=FALSE)`

Contoh, pada data “*mtcars*”

`as_tibble(mtcars)`

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     6   160   110   3.9   2.62  16.5    0
2  21     6   160   110   3.9   2.88  17.0    0
3  22.8    4   108    93   3.85   2.32  18.6    1
4  21.4    6   258   110   3.08   3.22  19.4    1
5  18.7    8   360   175   3.15   3.44  17.0    0
6  18.1    6   225   105   2.76   3.46  20.2    1
7  14.3    8   360   245   3.21   3.57  15.8    0
8  24.4    4   147.    62   3.69   3.19   20     1
9  22.8    4   141.    95   3.92   3.15  22.9    1
10 19.2    6   168.   123   3.92   3.44  18.3    1
# ... with 22 more rows, and 3 more variables:
#   am <dbl>, gear <dbl>, carb <dbl>
```

Gambar 3.36 Tampilan Data “*mtcars*” (Sumber: Penulis)

`filter(as_tibble(mtcars), mpg>20)`

```
# A tibble: 14 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     6   160   110   3.9   2.62  16.5    0
2  21     6   160   110   3.9   2.88  17.0    0
3  22.8    4   108    93   3.85   2.32  18.6    1
4  21.4    6   258   110   3.08   3.22  19.4    1
5  24.4    4   147.    62   3.69   3.19   20     1
6  22.8    4   141.    95   3.92   3.15  22.9    1
7  32.4    4   78.7    66   4.08   2.2   19.5    1
8  30.4    4   75.7    52   4.93   1.62  18.5    1
9  33.9    4   71.1    65   4.22   1.84  19.9    1
10 21.5    4   120.    97   3.7    2.46  20.0    1
11 27.3    4    79     66   4.08   1.94  18.9    1
12  26     4   120.    91   4.43   2.14  16.7    0
13 30.4    4   95.1   113   3.77   1.51  16.9    1
14 21.4    4   121    109   4.11   2.78  18.6    1
# ... with 3 more variables: am <dbl>, gear <dbl>,
#   carb <dbl>
```

Gambar 3.37 Output Filter (Sumber: Penulis)

Output diatas menunjukkan bahwa variabel mpg hanya menampilkan nilai > 20

Berikut beberapa operator *logical* dan *boolean* yang digunakan untuk code `filter ( )`

Tabel 3.5 Operator Logika untuk Filter ()

Operator	Fungsi
<code>==</code>	Sama dengan
<code>!=</code>	Tidak sama dengan
<code><</code>	Lebih kecil
<code>></code>	Lebih besar
<code><=</code>	Lebih kecil sama dengan
<code>>=</code>	Lebih besar sama dengan
<code>is.na</code>	Deteksi <i>missing value</i>
<code>!is.na</code>	Deteksi <i>missing value</i>
<code>%in%</code>	Identifikasi apakah sebuah nilai merupakan bagian dari vektor
<code>!</code>	Operator tidak
<code> </code>	Operator atau
<code>&</code>	Operator dan
<code>xor</code>	<i>Exclusive or</i>

2. Menghilangkan baris yang memiliki nilai yang sama

`distinct(as_tibble(mtcars),cyl)`

```
# A tibble: 3 x 1
  cyl
<dbl>
1     6
2     4
3     8
```

Gambar 3.38 Output Distinct (Sumber: Penulis)

Output diatas menunjukkan bahwa pada variabel cyl, nilai yang sama yaitu 6,4, dan 8

3. Memilih baris berdasarkan posisi

`slice(data,...,preserve =FALSE)`

`slice(as_tibble(mtcars),3:9)`

Contoh,

```
# A tibble: 7 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  22.8     4  108     93  3.85  2.32  18.6     1
2  21.4     6  258    110  3.08  3.22  19.4     1
3  18.7     8  360    175  3.15  3.44  17.0     0
4  18.1     6  225    105  2.76  3.46  20.2     1
5  14.3     8  360    245  3.21  3.57  15.8     0
6  24.4     4  147.     62  3.69  3.19   20     1
7  22.8     4  141.     95  3.92  3.15  22.9     1
# ... with 3 more variables: am <dbl>, gear <dbl>,
#   carb <dbl>
```

Gambar 3.39 Output Slice (Sumber: Penulis)

Output diatas menunjukkan data “mtcars” mulai dari row ke 3 sampai dengan row ke 9.

4. Memilih baris secara random

`slice_sample(data,..., n, prop, weight_by =
NULL, replace = FALSE)`

`slice_sample(as_tibble(mtcars),n=5,replace=TRUE)`

Replace = TRUE menunjukkan bahwa data diambil kemudian dilakukan pengembalian

contoh,

```
# A tibble: 5 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs    am  gear
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  18.1     6  225    105  2.76  3.46  20.2     1     0     3
2  18.1     6  225    105  2.76  3.46  20.2     1     0     3
3  22.8     4  108     93  3.85  2.32  18.6     1     1     4
4  19.2     8  400    175  3.08  3.84  17.0     0     0     3
5  21.4     6  258    110  3.08  3.22  19.4     1     0     3
# ... with 1 more variable: carb <dbl>
```

Gambar 3.40 Output Slice (1) (Sumber: Penulis)

5. Memilih baris berdasarkan urutan pada variabel

`slice_min(data, order_by,...,n, prop, with_ties = TRUE)`

`slice_max(data, order_by,...,n, prop, with_ties = TRUE)`

`slice_min ( )` digunakan untuk memilih baris berdasarkan urutan pada variabel tertentu dan diurutkan berdasarkan dari yang terkecil hingga ke yang terbesar. Sedangkan untuk `slice_max ( )` digunakan untuk memilih baris dari yang memiliki nilai terbesar ke yang terkecil.

Contoh,

`slice_min(as_tibble(mtcars),mpg,prop=0.25)`

```
# A tibble: 8 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  10.4     8  472   205  2.93  5.25  18.0     0   0     3
2  10.4     8  460   215   3     5.42  17.8     0   0     3
3  13.3     8  350   245  3.73  3.84  15.4     0   0     3
4  14.3     8  360   245  3.21  3.57  15.8     0   0     3
5  14.7     8  440   230  3.23  5.34  17.4     0   0     3
6  15      8  301   335  3.54  3.57  14.6     0   1     5
7  15.2     8  276   180  3.07  3.78  18      0   0     3
8  15.2     8  304   150  3.15  3.44  17.3     0   0     3
# ... with 1 more variable: carb <dbl>
```

Gambar 3.41 Output Slice (2) (Sumber: Penulis)

Output diatas menunjukkan baris dengan urutan nilai terkecil hingga ke terbesar berdasarkan variabel mpg dan sebanyak 25% dari seluruh baris.

`slice_max(as_tibble(mtcars),mpg,prop=0.25)`

```
# A tibble: 9 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  33.9     4  71.1    65  4.22  1.84  19.9     1   1     4
2  32.4     4  78.7    66  4.08  2.2   19.5     1   1     4
3  30.4     4  75.7    52  4.93  1.62  18.5     1   1     4
4  30.4     4  95.1   113  3.77  1.51  16.9     1   1     5
5  27.3     4   79     66  4.08  1.94  18.9     1   1     4
6  26      4  120     91  4.43  2.14  16.7     0   1     5
7  24.4     4  147     62  3.69  3.19  20      1   0     4
8  22.8     4  108     93  3.85  2.32  18.6     1   1     4
9  22.8     4  141     95  3.92  3.15  22.9     1   0     4
# ... with 1 more variable: carb <dbl>
```

Gambar 3.42 Output Slice (3) (Sumber: Penulis)

Output diatas menunjukkan baris dengan urutan nilai terbesar hingga ke terkecil berdasarkan variabel mpg dan sebanyak 25% dari seluruh baris.

6. Memilih baris berdasarkan baris teratas/terbawah

`slice_head(as_tibble(mtcars),n=5)`

slice head digunakan untuk memilih baris teratas sebanyak tertentu.

Contoh,

```
# A tibble: 5 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     6   160   110  3.9    2.62  16.5    0    1    4
2  21     6   160   110  3.9    2.88  17.0    0    1    4
3  22.8    4   108    93  3.85   2.32  18.6    1    1    4
4  21.4    6   258   110  3.08   3.22  19.4    1    0    3
5  18.7    8   360   175  3.15   3.44  17.0    0    0    3
# ... with 1 more variable: carb <dbl>
```

Gambar 3.43 Output Slice (4) (Sumber: Penulis)

Output diatas menampilkan 5 data teratas dari “*mtcars*” dataset

`slice_tail(as_tibble(mtcars),n=5)`

```
# A tibble: 5 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  30.4    4  95.1   113  3.77  1.51  16.9    1    1    5
2  15.8    8 351    264  4.22  3.17  14.5    0    1    5
3  19.7    6 145    175  3.62  2.77  15.5    0    1    5
4  15     8 301    335  3.54  3.57  14.6    0    1    5
5  21.4    4 121    109  4.11  2.78  18.6    1    1    4
# ... with 1 more variable: carb <dbl>
```

Gambar 3.44 Output Slice (5) (Sumber: Penulis)

Output diatas menampilkan 5 data terbawah dari “*mtcars*” dataset.

7. Mengurutkan Baris

`arrange(as_tibble(mtcars),desc(mpg))`

fungsi tersebut digunakan untuk mengurutkan nilai dari yang terbesar hingga terkecil berdasarkan variabel tertentu.

Contoh,

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  33.9     4  71.1    65  4.22  1.84  19.9
2  32.4     4  78.7    66  4.08  2.2   19.5
3  30.4     4  75.7    52  4.93  1.62  18.5
4  30.4     4  95.1   113  3.77  1.51  16.9
5  27.3     4   79     66  4.08  1.94  18.9
6  26      4  120.    91  4.43  2.14  16.7
7  24.4     4  147.    62  3.69  3.19  20
8  22.8     4  108     93  3.85  2.32  18.6
9  22.8     4  141.    95  3.92  3.15  22.9
10 21.5     4  120.    97  3.7   2.46  20.0
# ... with 22 more rows, and 4 more variables:
#   vs <dbl>, am <dbl>, gear <dbl>, carb <dbl>
```

Gambar 3.45 Output Arrange (Sumber: Penulis)

Output diatas mengurutkan baris dari yang memiliki nilai terbesar hingga terkecil berdasarkan variabel mpg.

Selanjutnya jika ingin mengurutkan baris dari yang terkecil hingga terbesar, dapat digunakan fungsi

`arrange(as_tibble(mtcars), mpg)`

Contoh,

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  10.4     8  472    205  2.93  5.25  18.0
2  10.4     8  460    215   3     5.42  17.8
3  13.3     8  350    245  3.73  3.84  15.4
4  14.3     8  360    245  3.21  3.57  15.8
5  14.7     8  440    230  3.23  5.34  17.4
6  15      8  301    335  3.54  3.57  14.6
7  15.2     8  276    180  3.07  3.78  18
8  15.2     8  304    150  3.15  3.44  17.3
9  15.5     8  318    150  2.76  3.52  16.9
10 15.8     8  351    264  4.22  3.17  14.5
# ... with 22 more rows, and 4 more variables:
#   vs <dbl>, am <dbl>, gear <dbl>, carb <dbl>
```

Gambar 3.46 Output Arrange (1) (Sumber: Penulis)

Output diatas mengurutkan baris dari yang terkecil hingga terbesar berdasarkan variabel mpg.

8. Menambah Baris

```
add_row(cars,speed=1,dist=1)
```

Contoh,

Dengan menggunakan dataset ‘cars’

37	19	46		37	19	46
38	19	68		38	19	68
39	20	32		39	20	32
40	20	48		40	20	48
41	20	52		41	20	52
42	20	56	→	42	20	56
43	20	64		43	20	64
44	22	66		44	22	66
45	23	54		45	23	54
46	24	70		46	24	70
47	24	92		47	24	92
48	24	93		48	24	93
49	24	120		49	24	120
50	25	85		50	25	85
				51	1	1

Gambar 3.47 Output Add_Row (Sumber: Penulis)

3.11.2 Manipulasi Kolom/Variabel

1. Mengekstrak nilai pada kolom sebagai vektor

```
pull(as_tibble(mtcars),wt)
```

digunakan untuk mengekstrak nilai pada kolom menggunakan nama kolom “wt”

```
pull(as_tibble(mtcars),6)
```

digunakan untuk mengekstrak nilai pada kolom menggunakan *index* kolom “wt” yaitu 6

```
[1] 2.620 2.875 2.320 3.215 3.440 3.460 3.570
[8] 3.190 3.150 3.440 3.440 4.070 3.730 3.780
[15] 5.250 5.424 5.345 2.200 1.615 1.835 2.465
[22] 3.520 3.435 3.840 3.845 1.935 2.140 1.513
[29] 3.170 2.770 3.570 2.780
```

Gambar 3.48 Output Pull (Sumber: Penulis)

Output diatas menunjukkan nilai pada variabel “wt” yang diekstrak menjadi vektor.

2. Mengekstrak nilai pada kolom sebagai tabel

`select(as_tibble(mtcars),mpg,wt)`

```
# A tibble: 32 x 11
  mpg   cyl  disp    hp  drat    wt   qsec    vs
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    21     6   160   110   3.9    2.62  16.5    0
2    21     6   160   110   3.9    2.88  17.0    0
3   22.8     4   108    93   3.85   2.32  18.6    1
4   21.4     6   258   110   3.08   3.22  19.4    1
5   18.7     8   360   175   3.15   3.44  17.0    0
6   18.1     6   225   105   2.76   3.46  20.2    1
7   14.3     8   360   245   3.21   3.57  15.8    0
8   24.4     4  147.    62   3.69   3.19   20     1
9   22.8     4  141.    95   3.92   3.15  22.9    1
10  19.2     6  168.   123   3.92   3.44  18.3    1
# ... with 22 more rows, and 3 more variables:
#   am <dbl>, gear <dbl>, carb <dbl>
```

→

```
# A tibble: 32 x 2
  mpg   wt
  <dbl> <dbl>
1    21  2.62
2    21  2.88
3   22.8  2.32
4   21.4  3.22
5   18.7  3.44
6   18.1  3.46
7   14.3  3.57
8   24.4  3.19
9   22.8  3.15
10  19.2  3.44
# ... with 22 more rows
```

Gambar 3.49 Output Select (Sumber: Penulis)

Berikut ini merupakan bantuan untuk fungsi `select ( )`

Tabel 3.6 Fungsi Select ()

Fungsi

Tujuan

`Starts_with ( )` Memilih variabel berdasarkan prefix.
Contoh pada data iris,

`select(as_tibble(iris),
starts_with("Sepal"))`

```
# A tibble: 150 x 2
  Sepal.Length Sepal.width
  <dbl>         <dbl>
1     5.1         3.5
2     4.9         3
3     4.7         3.2
4     4.6         3.1
5     5           3.6
6     5.4         3.9
7     4.6         3.4
8     5           3.4
9     4.4         2.9
10    4.9         3.1
# ... with 140 more rows
```

Output diatas menampilkan variabel dengan Prefix “Sepal”

Ends_with () Memilih variabel berdasarkan suffix.

Contoh pada data iris,

```
select(as_tibble(iris),
ends_with("Width"))
```

```
# A tibble: 150 x 2
  Sepal.Width Petal.Width
    <dbl>         <dbl>
1      3.5         0.2
2      3          0.2
3      3.2         0.2
4      3.1         0.2
5      3.6         0.2
6      3.9         0.4
7      3.4         0.3
8      3.4         0.2
9      2.9         0.2
10     3.1         0.1
# ... with 140 more rows
```

Output diatas menampilkan variabel dengan Prefix “Width”

Contains () Memilih variabel yang menngandung kata tertentu. Contoh,

```
select(as_tibble(iris), contains("al"))
```

```
# A tibble: 150 x 4
  Sepal.Length Sepal.Width Petal.Length Petal.Width
    <dbl>         <dbl>         <dbl>         <dbl>
1      5.1         3.5         1.4         0.2
2      4.9         3          1.4         0.2
3      4.7         3.2         1.3         0.2
4      4.6         3.1         1.5         0.2
5      5          3.6         1.4         0.2
6      5.4         3.9         1.7         0.4
7      4.6         3.4         1.4         0.3
8      5          3.4         1.5         0.2
9      4.4         2.9         1.4         0.2
10     4.9         3.1         1.5         0.1
# ... with 140 more rows
```

Output diatas menampilkan variabel yang menggunakan kata“al”

Matches () Memilih variabel dengan kata terentu

```
select(as_tibble(iris), matches("[pt]al"))
```

```
# A tibble: 150 x 4
  Sepal.Length Sepal.Width Petal.Length
    <dbl>         <dbl>         <dbl>
1      5.1         3.5         1.4
2      4.9         3         1.4
3      4.7         3.2         1.3
4      4.6         3.1         1.5
5      5         3.6         1.4
6      5.4         3.9         1.7
7      4.6         3.4         1.4
8      5         3.4         1.5
9      4.4         2.9         1.4
10     4.9         3.1         1.5
# ... with 140 more rows, and 1 more variable:
#   Petal.Width <dbl>
```

Num_range () Memilih variabel berdasarkan prefix dan range nya. Contoh dengan menggunakan dataset “billboard”

```
select(billboard,num_range("wk", 10:15))
```

```
# A tibble: 317 x 6
  wk10 wk11 wk12 wk13 wk14 wk15
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    NA    NA    NA    NA    NA    NA
2    NA    NA    NA    NA    NA    NA
3    51    51    51    47    44    38
4    61    61    59    61    66    72
5    57    64    70    75    76    78
6     6     7    22    29    36    47
7    NA    NA    NA    NA    NA    NA
8    36    37    37    38    49    61
9    10     9     8     6     1     2
10   59    66    68    61    67    59
# ... with 307 more rows
```

Output diatas hanya menampilkan variabel dengan prefix “wk” dan range number 10 sampai dengan 15

3. Memindahkan kolom ke kolom yang lain

`relocate(as_tibble(mtcars),mpg,cyl,last_col())`

```
# A tibble: 32 x 11
  mpg   cyl carb disp  hp drat   wt
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    21     6   4  160  110  3.9  2.62
2    21     6   4  160  110  3.9  2.88
3   22.8     4   1  108   93  3.85  2.32
4   21.4     6   1  258  110  3.08  3.22
5   18.7     8   2  360  175  3.15  3.44
6   18.1     6   1  225  105  2.76  3.46
7   14.3     8   4  360  245  3.21  3.57
8   24.4     4   2  147.   62  3.69  3.19
9   22.8     4   2  141.   95  3.92  3.15
10  19.2     6   4  168.  123  3.92  3.44
# ... with 22 more rows, and 4 more variables:
#   qsec <dbl>, vs <dbl>, am <dbl>, gear <dbl>
```

Gambar 3.50 Output Relocate (Sumber: Penulis)

Output diatas menunjukkan bahwa variabel “carb” diletakkan di setelah “mpg” dan “cyl”.

3.11.3 Membuat Variabel Baru

1. Menggunakan *mutate*

Fungsi ini akan membantu untuk membuat variabel baru dengan tidak menghilangkan variabel yang lainnya.

`mutate(data,...,keep="all",before=NULL,after=NULL)`

Contoh,

Membuat variabel baru dengan nama variabel “gpm” yang merupakan 1/mpg

`a = mutate(as_tibble(mtcars), gpm=1/mpg)`

```
# A tibble: 32 x 12
  mpg   cyl disp  hp drat   wt  qsec  vs
<dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    21     6  160  110  3.9  2.62  16.5    0
2    21     6  160  110  3.9  2.88  17.0    0
3   22.8     4  108   93  3.85  2.32  18.6    1
4   21.4     6  258  110  3.08  3.22  19.4    1
5   18.7     8  360  175  3.15  3.44  17.0    0
6   18.1     6  225  105  2.76  3.46  20.2    1
7   14.3     8  360  245  3.21  3.57  15.8    0
8   24.4     4  147.   62  3.69  3.19  20.0    1
9   22.8     4  141.   95  3.92  3.15  22.9    1
10  19.2     6  168.  123  3.92  3.44  18.3    1
# ... with 22 more rows, and 4 more variables:
#   am <dbl>, gear <dbl>, carb <dbl>, gpm <dbl>
```

Gambar 3.51 Output Mutate (Sumber: Penulis)

Output diatas menunjukkan bahwa pada dataset terdapat variabel baru yaitu “gpm”. Untuk melihat nilai dari variabel tersebut dapat digunakan fungsi dibawah ini

`b = a[,6:12]`

```
# A tibble: 32 x 7
      wt    qsec  vs    am  gear  carb    gpm
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  2.62  16.5    0     1     4     4  0.0476
2  2.88  17.0    0     1     4     4  0.0476
3  2.32  18.6    1     1     4     1  0.0439
4  3.22  19.4    1     0     3     1  0.0467
5  3.44  17.0    0     0     3     2  0.0535
6  3.46  20.2    1     0     3     1  0.0552
7  3.57  15.8    0     0     3     4  0.0699
8  3.19  20      1     0     4     2  0.0410
9  3.15  22.9    1     0     4     2  0.0439
10 3.44  18.3    1     0     4     4  0.0521
# ... with 22 more rows
```

Gambar 3.52 Output Nilai Variabel Baru (Sumber: Penulis)

Output tersebut menunjukkan variabel ke 6 hingga variabel ke 12 (variabel baru) dari dataset “mtcars”.

2. Menggunakan *transmute*

Fungsi ini akan membantu untuk membuat variabel baru dengan menghilangkan variabel yang lainnya.

`transmute(as_tibble(mtcars), gpm = 1 / mpg)`

```
# A tibble: 32 x 1
      gpm
  <dbl>
1  0.0476
2  0.0476
3  0.0439
4  0.0467
5  0.0535
6  0.0552
7  0.0699
8  0.0410
9  0.0439
10 0.0521
# ... with 22 more rows
```

Gambar 3.53 Output Transmute (Sumber: Penulis)

Berikut adalah beberapa fungsi yang dapat digunakan bersamaan dengan *mutate* () dan *transmute* ().

Tabel 3.7 Fungsi <i>Mutate ()</i> dan <i>Transmute ()</i>		
Fungsi	Kode	Tujuan
Offset	lag ()	Menghilangkan elemen 1
	lead ()	Menghilangkan elemen -1
Cumulative aggregate	cumall ()	Cumulative all
	cumany()	Cumulative any
	cummax ()	Cumulative max
	cummean()	Cumulative mean
	cummin()	Cumulative min
	cumprod ()	Cumulative prod
	cumsum ()	Cumulative sum
	cume_dist()	Menghitung proporsi semua nilai <= nilai sekarang. Misal, <code>x <- c(5, 1, 3, 2, 2, NA)</code> <code>cume_dist(x)</code> <code>[1] 1.0 0.2 0.8 0.6 0.6 NA</code>
Ranking	dense_rank()	<code>x <- c(5, 1, 3, 2, 2, NA)</code> <code>dense_rank(x)</code> <code>[1] 4 1 3 2 2 NA</code>
	min_rank()	<code>x <- c(5, 1, 3, 2, 2, NA)</code>

		<code>min_rank(x)</code>
		<code>[1] 5 1 4 2 3 NA</code>
	<code>ntile ()</code>	<code>x <- c(5, 1, 3, 2, 2, NA)</code> <code>ntile(x,2)</code> <code>[1] 2 1 2 1 1 NA</code>
	<code>percent_rank()</code>	<i>Rescaling min_rank between 0-1</i> <code>[1] 1.00 0.00 0.75 0.25 0.25 NA</code>
	<code>row_number ()</code>	Memberikan urutan <i>row</i> <code>[1] 5 1 4 2 3 NA</code>
<i>Math</i>	<code>+, -, *, /, ^, %/%, %%</code>	Operator aritmetika
	<code>log(), log2(), log10 ()</code>	Operator log
	<code><, <=, >, >=, !=, ==</code>	Operator perbandingan
	<code>between ()</code>	<code>x>=left & x<=right</code>
<i>Miscelannaneous</i>	<code>case_when ()</code>	<i>Multi-case if_else</i>
	<code>If_else</code>	<code>If () + else ()</code>
	<code>Na_if ()</code>	Mengganti nilai tertentu dengan NA
	<code>pmax</code>	Parallel max (Jika terdapat 2 <i>vector</i> akan dibandingkan mana yang memiliki nilai max setiap elemen)


```

x1 <- c(2, 8, 3, 4, 1, 5)
x2 <- c(0, 7, 5, 5, 6, 1)
pmax(x1,x2)
[1] 2 8 5 5 6 5

pmin
Parallel min (Jika
terdapat 2 vector akan
dibandingkan mana
yang memiliki nilai
min setiap elemen)

Misal,
x1 <- c(2, 8, 3, 4, 1, 5)
x2 <- c(0, 7, 5, 5, 6, 1)
pmin(x1,x2)
[1] 0 7 3 4 1 1

```

3. Merubah nama kolom dengan “*rename*”

```
rename(data,...)
```

Contoh,

```
rename(as_tibble(mtcars),CYL = cyl)
```

```

# A tibble: 32 x 11
   mpg   CYL  disp  hp  drat   wt   qsec    vs
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1    21     6   160   110   3.9   2.62  16.5    0
2    21     6   160   110   3.9   2.88  17.0    0
3   22.8     4   108    93   3.85   2.32  18.6    1
4   21.4     6   258   110   3.08   3.22  19.4    1
5   18.7     8   360   175   3.15   3.44  17.0    0
6   18.1     6   225   105   2.76   3.46  20.2    1
7   14.3     8   360   245   3.21   3.57  15.8    0
8   24.4     4   147.    62   3.69   3.19  20     1
9   22.8     4   141.    95   3.92   3.15  22.9    1
10  19.2     6   168.   123   3.92   3.44  18.3    1
# ... with 22 more rows, and 3 more variables:
#   am <dbl>, gear <dbl>, carb <dbl>

```

Gambar 3.54 Output Rename (Sumber: Penulis)

Output diatas menunjukkan bahwa variabel “cyl” telah di rename menjadi variabel “CYL”

3.11.4 Menyimpulkan Baris dan Kolom

1. Menggunakan *Summaries*

`summarize(data,...)`

contoh,

`summarise(mtcars, avg=mean(mpg))`

Fungsi ini akan menghasilkan rata-rata dari variabel “mpg”

2. Menggunakan *Count*

`count(data,...,wt = NULL, sort = FALSE, name = NULL)`

Contoh,

`count(mtcars,cyl)`

fungsi ini menampilkan jumlah nilai unik dalam variabel “cyl”.

cyl n

1 4 11

2 6 7

3 8 14

Beberapa function yang bisa digunakan bersama dengan *summarise ()* adalah sebagai berikut.

Tabel 3.8 Fungsi *Summarise ()*

Fungsi	Kode	Tujuan
<i>Count</i>	<code>n()</code>	Jumlah baris
	<code>n_distinct()</code>	Jumlah nilai unik pada baris
	<code>Sum(!is.na())</code>	Jumlah data yang tidak <i>missing value</i>
<i>Position</i>	<code>mean ()</code>	Rata-rata
	<code>mean (!is.na())</code>	Rata-rata data yang tidak <i>missing value</i>
	<code>median ()</code>	Nilai tengah
<i>Order</i>	<code>first ()</code>	Nilai awal
	<code>last ()</code>	Nilai akhir

	quantile ()	Kuantil ke-n
Rank	min ()	Nilai minimum
	max ()	Nilai maksimum
	IQR ()	<i>Inter-quartile Range</i>
Spread	mad ()	<i>Median absolute deviation</i>
	sd ()	Standar deviasi
	var ()	variansi

3.11.5 Operator Pipes

Operator pipes digunakan untuk membuat fungsi R. Misalkan $f(a,x)$ adalah fungsi di R dengan argumen a dan x , kemudian fungsi $g(b,z)$ adalah fungsi lain di R dengan argumen b dan z . Dengan menggunakan operator `%>%` kita dapat menuliskannya sebagai berikut.

fungsi $f(a, x)$

$f(a, x)$ # atau

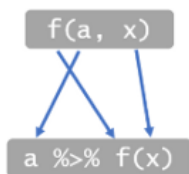
$a \%>\% f(x)$

fungsi $g(b, z)$

$g(b, z)$ # atau

$b \%>\% g(z)$

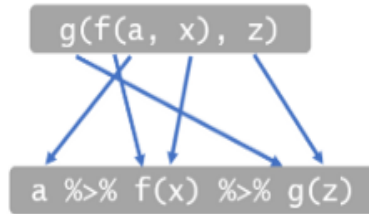
Dari contoh diatas dapat dilihat bahwa a adalah argumen pertama untuk fungsi $f()$ dan b adalah argumen pertama untuk fungsi $g()$. Operator `%>%` “menyampaikan” objek a sebagai nilai untuk mengisi argumen pertama pada fungsi $f()$.



Ilustrasi Pipes 1

Gambar 3.55 Ilustrasi Pipes 1 (Sumber: Penulis)

Misalkan objek a menjadi argumen pertama fungsi $f()$ dengan $a \%>\% f(x)$. Kemudian hasil dari $a \%>\% f(x)$ dijadikan argumen pertama dari fungsi $g()$. Dengan kata lain $b <- a \%>\% f(x)$ sehingga $b \%>\% g(z)$.



Ilustrasi Pipes 2

Gambar 3.56 Ilustrasi Pipes 2 (Sumber: Penulis)

Fungsi $f()$ diterapkan terhadap objek a sebagai argumen pertama dan x sebagai argumen kedua dari fungsi $f()$ yang kemudian hasilnya digunakan sebagai argumen pertama pada fungsi $g()$ dengan z sebagai argumen kedua.

Sebagai contoh kita gunakan `select()`, `filter()`, dan `arrange()` untuk eksplorasi dataset “mtcars”.

```
as_tibble(mtcars)
mtcars %>%
  select(cyl, disp, hp, wt, vs) %>%
  filter(cyl == 6 & between(wt, 2, 4)) %>%
  arrange(disp, desc(hp))
```

Syntax diatas menunjukkan untuk mengambil variabel `cyl`, `disp`, `hp`, `wt`, dan `vs` dari dataset `mtcars`. Kemudian dilakukan filter berdasarkan variabel `cyl` yang bernilai 6 dan variabel `wt` yang bernilai 2 sampai 4. Selanjutnya diurutkan secara *ascending* berdasarkan variabel `disp` serta diurutkan secara *descending* berdasarkan variabel `hp`.

	cyl	disp	hp	wt	vs
Ferrari Dino	6	145.0	175	2.770	0
Mazda RX4	6	160.0	110	2.620	0
Mazda RX4 wag	6	160.0	110	2.875	0
Merc 280	6	167.6	123	3.440	1
Merc 280C	6	167.6	123	3.440	1
valiant	6	225.0	105	3.460	1
Hornet 4 Drive	6	258.0	110	3.215	1

Gambar 3.57 Output Select, Filter, Arrange (Sumber: Penulis)

3.11.6 Mengkategorikan Baris

1. Menggunakan *Group_By*

Group_by digunakan untuk mengkategorikan berdasarkan variabel/kolom. Contoh,

```
as_tibble(mtcars)
```

```
mtcars%>%
```

```
  group_by(cyl)%>%
```

```
  summarise(avg=mean(mpg))
```

Syntax tersebut akan menampilkan rata-rata dari setiap nilai pada variabel cyl.

	cyl	avg
	<dbl>	<dbl>
1	4	26.7
2	6	19.7
3	8	15.1

Gambar 3.58 Output *Group_by* (Sumber: Penulis)

Output diatas menunjukkan rata-rata dari setiap kelompok nilai pada variabel cyl.

2. Menggunakan *Rowwise*

Rowwise digunakan untuk mengkategorikan semua baris berdasarkan beberapa kolom.

Contoh,

```
a = mtcars %>%
```

```
  rowwise() %>%
```

```
mutate(average_3_var = mean(c(mpg,cyl)))
```

Syntax tersebut akan membuat variabel baru dengan nama average_3_var yang merupakan nilai rata-rata dari variabel mpg dan cyl untuk semua baris

```
# A tibble: 32 x 1
# Rowwise:
  average_3_var
    <dbl>
1      13.5
2      13.5
3      13.4
4      13.7
5      13.4
6      12.0
7      11.2
8      14.2
9      13.4
10     12.6
# ... with 22 more rows
```

Gambar 3.59 Output Rowwise (Sumber: Penulis)

3.11.7 Memberi Nama Baris

Pada *tidy* data biasanya nama baris tidak diberikan, yang terletak diluar kolom. Untuk memberi nama tersebut maka digunakan fungsi `rownames_to_column( )`. Atau sebaliknya menggunakan `column_to_rownames( )`.

Contoh,

```
b=rownames_to_column(mtcars,var="Jenis Mobil")
```

```
b[1:10,]
```

	Jenis Mobil	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
2	Mazda RX4 wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
6	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4

Gambar 3.60 Output Rownames (Sumber: Penulis)

3.11.8 Mengkombinasikan Tabel atau Variabel

1. Mengkombinasikan variabel

a =band_members

b =band_instruments

bind_cols(a,b)

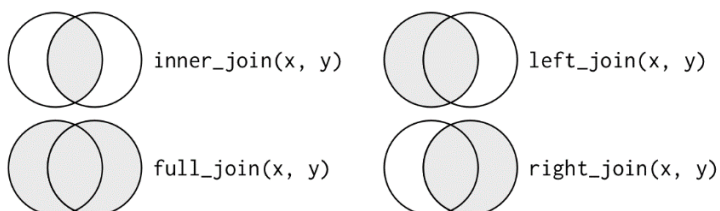
```
# A tibble: 3 x 2      # A tibble: 3 x 2
  name band          name plays
<chr> <chr>      <chr> <chr>
1 Mick  Stones      1 John  guitar
2 John  Beatles     2 Paul  bass
3 Paul  Beatles     3 Keith guitar

+

New names:
* name -> name...1
* name -> name...3
# A tibble: 3 x 4
  name...1 band      name...3 plays
<chr>    <chr>    <chr>    <chr>
1 Mick    Stones    John    guitar
2 John    Beatles   Paul    bass
3 Paul    Beatles   Keith   guitar
```

Gambar 3.61 Output Mengkombinasikan Variabel (Sumber: Penulis)

2. Menggabungkan Data yang Saling Berhubungan (*Relational Data*) Berdasarkan Kolom



Gambar 3.62 Ilustrasi Penggabungan Data (Sumber: Penulis)

a =band_members

b=band_instruments

band_members %>% left_join(band_instruments)

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 3 x 3
  name band plays
  <chr> <chr> <chr>
1 Mick Stones NA
2 John Beatles guitar
3 Paul Beatles bass

```

Gambar 3.63 Output *Left_join* (Sumber: Penulis)

Output tersebut menggabungkan data yang saling berhubungan berdasarkan dataset yang sebelah kiri.

```
a =band_members
```

```
b=band_instruments
```

```
band_members %>% right_join(band_instruments, by="name")
```

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 3 x 3
  name band plays
  <chr> <chr> <chr>
1 John Beatles guitar
2 Paul Beatles bass
3 Keith NA guitar

```

Gambar 3.64 Output *Right_join* (Sumber: Penulis)

Output tersebut menggabungkan data yang saling berhubungan berdasarkan dataset yang sebelah kanan.

```
a =band_members
```

```
b=band_instruments
```



```
band_members %>% inner_join(band_instruments, by="name")

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick  Stones
2 John  Beatles
3 Paul  Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John  guitar
2 Paul  bass
3 Keith guitar

# A tibble: 2 x 3
  name band plays
  <chr> <chr> <chr>
1 John Beatles guitar
2 Paul Beatles bass
```

Gambar 3.65 Output Inner_join (Sumber: Penulis)

Output tersebut menggabungkan data yang saling berhubungan (beririsan).

```
a =band_members
b=band_instruments
band_members %>% full_join(band_instruments, by="name")

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick  Stones
2 John  Beatles
3 Paul  Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John  guitar
2 Paul  bass
3 Keith guitar

# A tibble: 4 x 3
  name band plays
  <chr> <chr> <chr>
1 Mick  Stones NA
2 John  Beatles guitar
3 Paul  Beatles bass
4 Keith NA guitar
```

Gambar 3.66 Output Full_join Berdasarkan Variabel Bernilai Sama (Sumber: Penulis)

Output tersebut menggabungkan semua data dari variabel yang memiliki nilai yang sama.

Jika join variabel memiliki nama variabel yang berbeda maka dapat digunakan “by”. Contoh,

```

a=band_members
b=band_instruments2
band_members %>% full_join(band_instruments2, by = c("name" =
"artist"))
# A tibble: 3 x 2          # A tibble: 3 x 2
  name band              artist plays
  <chr> <chr>              <chr> <chr>
1 Mick Stones          1 John guitar
2 John Beatles        2 Paul bass
3 Paul Beatles        3 Keith guitar

+

# A tibble: 4 x 3
  name band plays
  <chr> <chr> <chr>
1 Mick Stones NA
2 John Beatles guitar
3 Paul Beatles bass
4 Keith NA guitar

```

Gambar 3.67 Output *Full_join* Berdasarkan Nama Variabel yang Berbeda (Sumber: Penulis)

3.11.9 Mengkombinasikan Baris/Cases/Row

1. Mengkombinasikan Baris

a=band_members

b=band_instruments

c=band_instruments2

bind_rows(a,b)

a

b

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 6 x 3
  name band plays
  <chr> <chr> <chr>
1 Mick Stones NA
2 John Beatles NA
3 Paul Beatles NA
4 John NA guitar
5 Paul NA bass
6 Keith NA guitar

```

Gambar 3.68 Output Mengkombinasikan Baris (Sumber: Penulis)

`bind_rows(a,c)`

```

a                                     c
# A tibble: 3 x 2                  # A tibble: 3 x 2
  name band                        artist plays
  <chr> <chr>                      <chr> <chr>
1 Mick Stones                    1 John guitar
2 John Beatles                  2 Paul bass
3 Paul Beatles                  3 Keith guitar

+

# A tibble: 6 x 4
  name band artist plays
  <chr> <chr> <chr> <chr>
1 Mick Stones NA NA
2 John Beatles NA NA
3 Paul Beatles NA NA
4 NA NA John guitar
5 NA NA Paul bass
6 NA NA Keith guitar

```

Gambar 3.69 Output Mengkombinasikan Baris (1) (Sumber: Penulis)

2. Filter Baris

`a=band_members`

`b=band_instruments`

`band_members %>% semi_join(band_instruments, by = "name")`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 2 x 2
  name band
  <chr> <chr>
1 John Beatles
2 Paul Beatles

```

Gambar 3.70 Output Filter Baris *Semi_join* Berdasarkan Nama Variabel yang Sama (Sumber: Penulis)

`a=band_members`

`c=band_instruments2`

`band_members %>% semi_join(band_instruments2, by = c("name" = "artist"))`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  artist plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 2 x 2
  name band
  <chr> <chr>
1 John Beatles
2 Paul Beatles

```

Gambar 3.71 Output *Semi_join* Berdasarkan Nama Variabel yang Berbeda (Sumber: Penulis)

`a=band_members`

`b=band_instruments`

`band_members %>% anti_join(band_instruments, by = "name")`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 1 x 2
  name band
  <chr> <chr>
1 Mick Stones

```

Gambar 3.72 Output Filter Baris *Anti_join* Berdasarkan Nama Variabel yang Sama (Sumber: Penulis)

`a=band_members`

`c=band_instruments2`

`band_members %>% anti_join(band_instruments2, by = c("name" = "artist"))`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  artist plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 1 x 2
  name band
  <chr> <chr>
1 Mick Stones

```

Gambar 3.73 Output *Anti_join* Berdasarkan Nama Variabel yang Berbeda (Sumber: Penulis)

`a=band_members`

`b=band_instruments`

`band_members %>% nest_join(band_instruments, by = "name")`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

# A tibble: 3 x 2
  name plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 3 x 3
  name band band_instruments
  <chr> <chr> <list>
1 Mick Stones <tibble [0 x 1]>
2 John Beatles <tibble [1 x 1]>
3 Paul Beatles <tibble [1 x 1]>

```

Gambar 3.74 Output Filter Baris *Nest_join* Berdasarkan Nama Variabel yang Sama
(Sumber: Penulis)

`a=band_members`

`c=band_instruments2`

`band_members %>% nest_join(band_instruments2, by = c("name" = "artist"))`

```

# A tibble: 3 x 2
  name band
  <chr> <chr>
1 Mick Stones
2 John Beatles
3 Paul Beatles

+

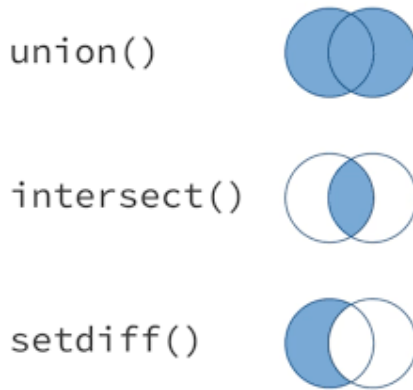
# A tibble: 3 x 2
  artist plays
  <chr> <chr>
1 John guitar
2 Paul bass
3 Keith guitar

# A tibble: 3 x 3
  name band band_instruments2
  <chr> <chr> <list>
1 Mick Stones <tibble [0 x 1]>
2 John Beatles <tibble [1 x 1]>
3 Paul Beatles <tibble [1 x 1]>

```

Gambar 3.75 Output Filter Baris *Nest_join* Berdasarkan Nama Variabel yang Berbeda
(Sumber: Penulis)

3. Menggabungkan Data yang Saling Berhubungan (*Relational Data*) Berdasarkan Baris



Gambar 3.76 Ilustrasi Penggabungan *Relational Data* (Sumber: Penulis)

```
mtcars
first <- mtcars[1:20, ]
second <- mtcars[10:32, ]
intersect(first, second)
```

Kode diatas digunakan untuk mengambil baris yang sama antara baris 1 sampai dengan ke 20 dan baris 10 sampai dengan 32.

	mpg	cyl	disp	hp	drat	wt	qsec
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90
Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98
Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82
Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42
Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90
	vs	am	gear	carb			
Merc 280	1	0	4	4			
Merc 280C	1	0	4	4			
Merc 450SE	0	0	3	3			
Merc 450SL	0	0	3	3			
Merc 450SLC	0	0	3	3			
Cadillac Fleetwood	0	0	3	4			
Lincoln Continental	0	0	3	4			
Chrysler Imperial	0	0	3	4			
Fiat 128	1	1	4	1			
Honda Civic	1	1	4	2			
Toyota Corolla	1	1	4	1			

Gambar 3.77 Output Intersect (Sumber: Penulis)

```
mtcars
```

```
first <- mtcars[1:20, ]
```

```
second <- mtcars[10:32, ]
```

```
union_all(first,second)
```

Kode diatas digunakan untuk menggabungkan baris yang sama antara baris 1 sampai dengan ke 20 dan baris 10 sampai dengan 32.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280...10	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C...11	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE...12	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL...13	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC...14	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood...15	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental...16	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial...17	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128...18	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic...19	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla...20	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Merc 280...21	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C...22	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE...23	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
Merc 450SL...24	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Merc 450SLC...25	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood...26	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
Lincoln Continental...27	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
Chrysler Imperial...28	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
Fiat 128...29	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
Honda Civic...30	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Toyota Corolla...31	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2

Gambar 3.78 Output *Union_all* (Sumber: Penulis)


```
a <- data.frame(column = c(1:10, 10))
```

```
b <- data.frame(column = c(1:5, 5))
```

```
setdiff(a,b)
```

Kode diatas digunakan untuk menampilkan baris yang muncul di dataset a namun tidak muncul di dataset b.

a	b	setdiff(a,b)
column		
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
	column	
	1	
	2	
	3	
	4	
	5	
	6	
		column
		1
		2
		3
		4
		5

Gambar 3.79 Output Setdiff (Sumber: Penulis)

3.12 R Project dan R Markdown

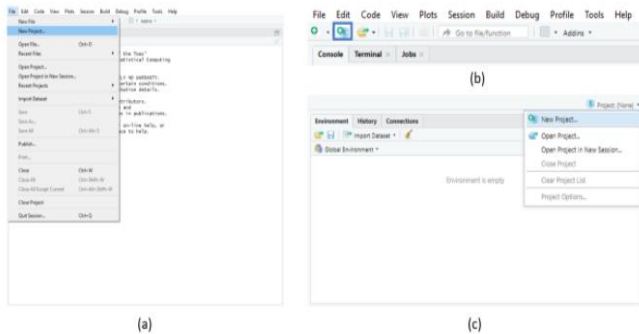
Setelah mempelajari R Project dan R Markdown, diharapkan mahasiswa mampu membuat analisis statistika sederhana menggunakan R Project dan R Markdown.

R markdown adalah format *file* yang digunakan untuk membuat dokumen dinamis menggunakan R *software*. Manfaat dari R markdown mempermudah melakukan konversi dari penulisan format *plain text* (text yang memberikan suatu informasi dan tidak menggunakan proses formatting text yang khusus) ke format *html*, *xhtml*, *pdf/latex* dan mempermudah jika ingin *sharing code*/mempublish kode secara online.

3.12.1 R Project

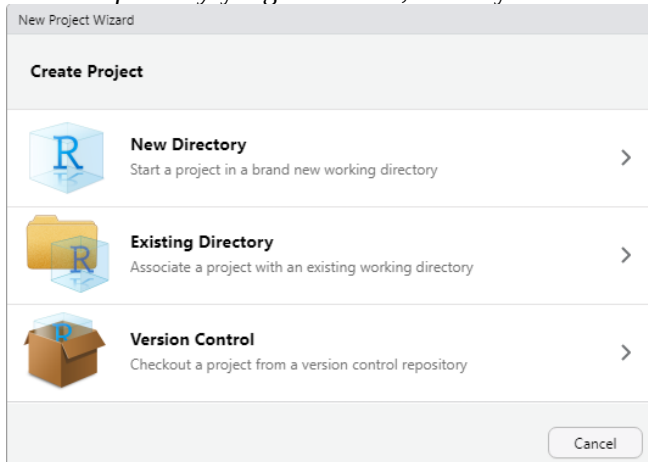
Hal pertama yang dilakukan ketika menggunakan R adalah membuat sebuah *project* untuk setiap pekerjaan yang berbeda. Langkah-langkah yang dilakukan untuk membuat suatu *project* baru di R adalah sebagai berikut

1. Pilih *New Project* dari 3 menu yang tersedia a) dari menu *File*, b) dari *Toolbar*, atau c) dari menu pojok kanan seperti pada gambar dibawah ini



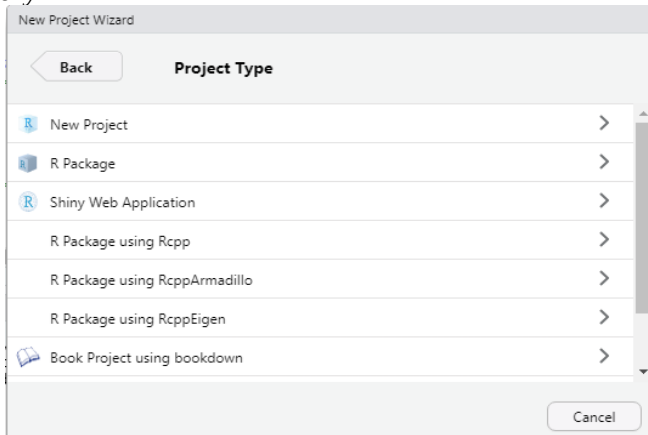
Gambar 3.80 Menampilkan *New Project* (Sumber: Penulis)

- Pilih jenis *project* yang ingin dibuat. Ada beberapa pilihan yaitu *New Directory* (Jika belum mempunyai folder yang dijadikan sebagai *working directory* untuk *project* tersebut). *Existing Directory* (Jika folder sudah ada sebelumnya). *Version Control* (Jika akan membuat *project* berdasarkan *repository* yang sudah ada, misalnya GitHub atau SVN



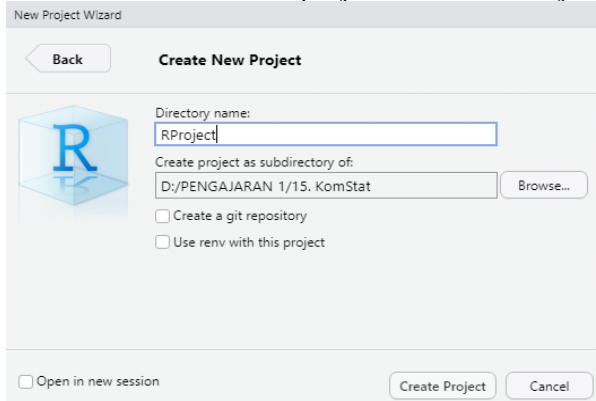
Gambar 3.81 Memilih Jenis *Project* (Sumber: Penulis)

3. Karena kita belum ada folder sebelumnya, maka digunakan *New Directory*



Gambar 3.82 Tampilan *New Directory* (Sumber: Penulis)

4. Pilih *Project Type* sesuai dengan *project* yang akan dibuat.
5. Tuliskan nama *project* yang juga akan menjadi folder baru sebagai subfolder di dalam folder “*Create project as subdirectory of*”



Gambar 3.83 Tampilan *Create New Project* (Sumber: Penulis)

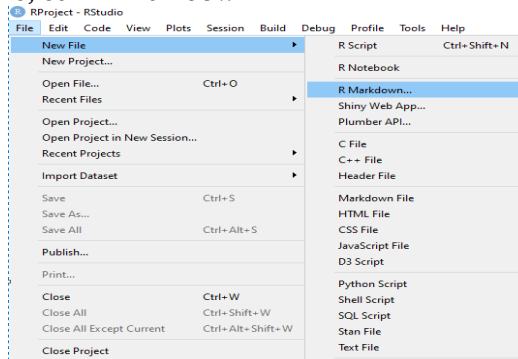
6. Klik *Create Project*

3.12.2 R Markdown

Selanjutnya *user* dapat menggunakan fasilitas dalam R yaitu R-Markdown. R-Markdown merupakan jenis *file* dalam R Studio yang

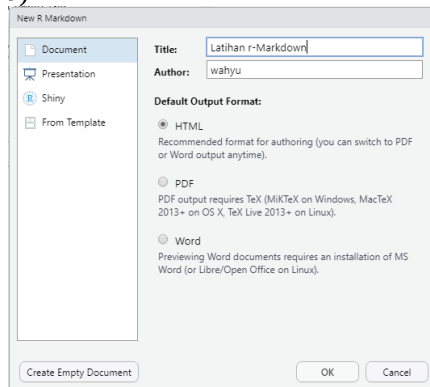
menggabungkan kemampuan *editing text* dan perintah *console*. Langkah-langkah membuka R-Markdown adalah sebagai berikut.

1. Klik *New File*, dan R-Markdown



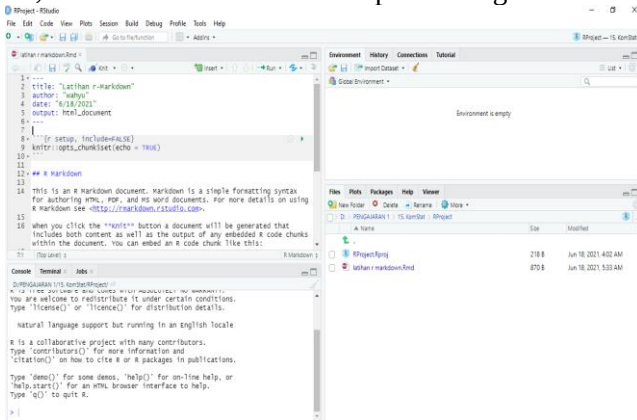
Gambar 3.84 Langkah Awal Membuka R-Markdown (Sumber: Penulis)

2. Tuliskan nama *title* R-markdown dan pilih *default output* (HTML, PDF, dan Word)

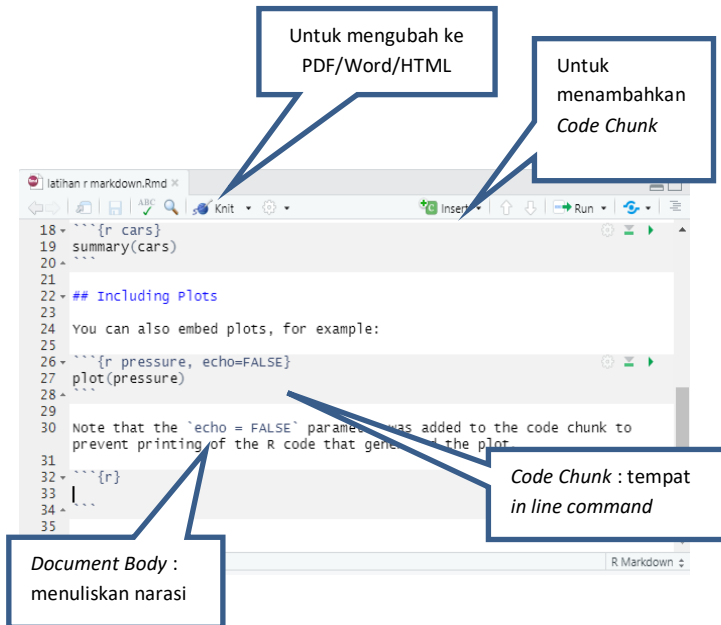


Gambar 3.85 Tampilan New R-Markdown (Sumber: Penulis)

3. Klik OK, kemudian akan muncul tampilan sebagai berikut



Gambar 3.86 Tampilan R Setelah Menggunakan R-Markdown (Sumber: Penulis)



Gambar 3.87 Penjelasan Tentang Tampilan R-Markdown (Sumber: Penulis)

Berikut merupakan contoh R markdown statistika deskriptif sederhana

Install rmarkdown dan knitr pada Rscript,

```
install.packages("rmarkdown")
```

```
install.packages("knitr")
```

```
library(rmarkdown)
```

```
library(knitr)
```

Selanjutnya tulis code pada file Rmd yang telah dibuat,

```
---
```

```
title: "Analysis of ToothGrowth Dataset"
```

```
author: "Muhammad Alifian"
```

```
date: "21/08/2021"
```

```
output : html_document
```

```
---
```

```
#Data Descriptive
```

```
```${r}```
```

```
dim(ToothGrowth)
```

```
head(ToothGrowth)
```

```
```
```

```
#Descriptive Statistics
```

```
```${r}```
```

```
summary(ToothGrowth)
```

```
sd(ToothGrowth$len)
```

```
```
```

```
The average length of the tooth is `${r} mean(ToothGrowth$len)` and  
`${r} sd(ToothGrowth$len)`
```

```
#Graphical Representation
```

```
```${r}```
```

```
hist(ToothGrowth$len)
```

```
boxplot(ToothGrowth$supp, ToothGrowth$len)
```

```
```
```

Latihan Soal

Butlah R Markdown untuk analisis statistika sederhana menggunakan salah satu R *datasets* meliputi statistika deskriptif, visualisasi (histogram, *scatterplot*, *boxplot*, dll), analisis korelasi, dan model regresi linear sederhana.

3.13 R Shiny

Shiny merupakan sebuah paket dalam R yang mengizinkan penggunaanya membangun *web apps* yang aplikatif. Shiny menggabungkan antara komputasi statistika R dan interaksinya dengan web modern. Di masa lalu, membuat aplikasi web sulit bagi sebagian besar pengguna R karena :

1. Kita membutuhkan pengetahuan yang mendalam tentang teknologi web seperti HTML, CSS, dan JavaScript.
2. Membuat aplikasi interaktif yang kompleks membutuhkan analisis yang cermat atas alur interaksi untuk memastikan bahwa saat masukan berubah, hanya output/keluaran terkait yang diperbaharui.

Shiny mempermudah programmer R untuk membuat aplikasi web dengan,

1. Menyediakan sekumpulan fungsi antarmuka pengguna (disingkat UI) yang dikurasi dengan cermat yang menghasilkan HTML, CSS, dan JavaScript yang diperlukan untuk tugas umum. Ini berarti kita tidak perlu mengetahui detail HTML/CSS/JavaScript
2. Memperkenalkan gaya baru pemrograman yang disebut pemrograman reaktif yang secara otomatis melacak ketergantungan potongan *code*.

Shiny terdiri dari 3 komponen, yaitu :

1. User Interface (UI)

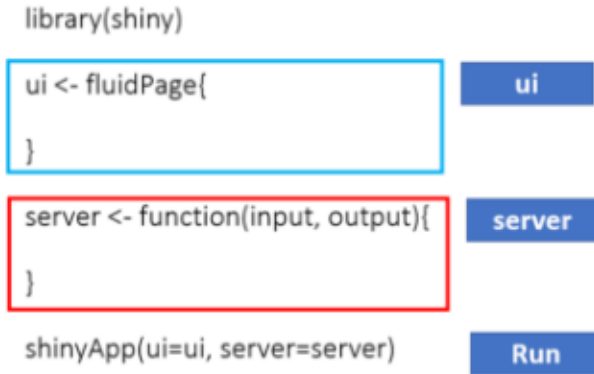
Fungsi yang mendefinisikan tampilan web dari aplikasi kita. Fungsinya memuat seluruh input dan *output* yang akan ditampilkan dalam app

2. Server

Fungsi yang mendefinisikan logika kerja analisis dari sisi *server* pada aplikasi

3. ShinyApp (shinyapp)

Fungsi dari aplikasi yang memanggil UI dan *server* untuk menjalankan aplikasi

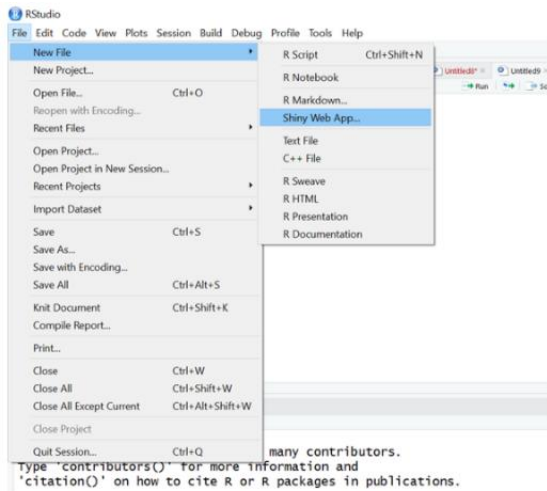


Gambar 3.88 Komponen Shiny (Sumber: Penulis)

3.13.1 Aplikasi R Shiny Sederhana

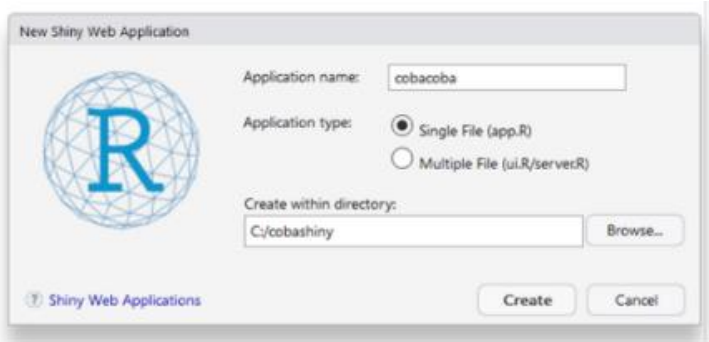
Berikut merupakan langkah-langkah membuat aplikasi R Shiny sederhana

1. Klik *File*, *New File*, pilih *Shiny Web App*



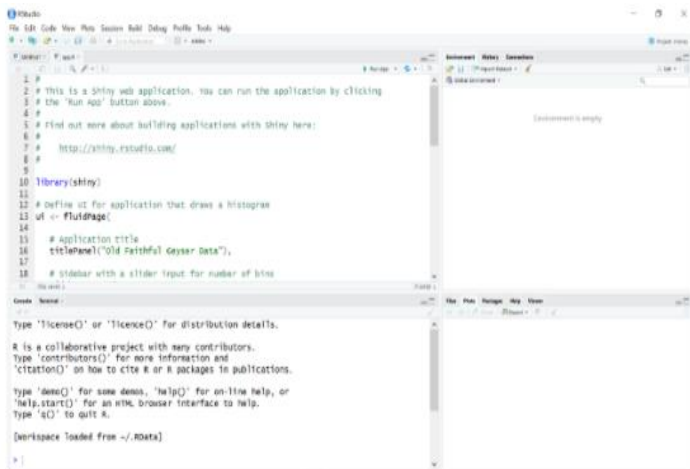
Gambar 3.89 Langkah Awal Membuat Aplikasi R Shiny (Sumber: Penulis)

2. Kemudian akan muncul seperti berikut, dan buatlah folder di C:/cobashiny dan pilih “Single File (app.R)” terlebih dahulu



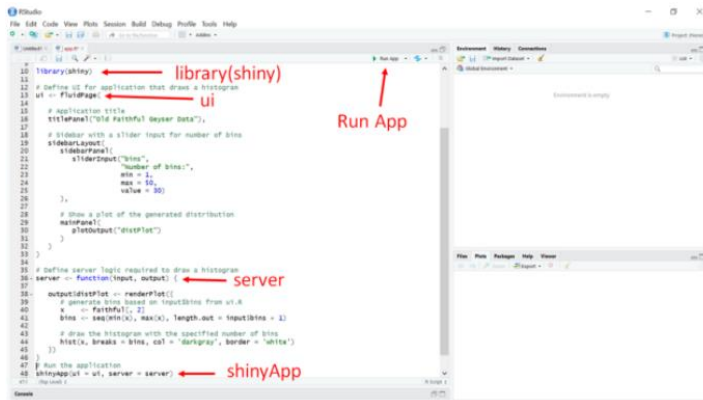
Gambar 3.90 Membuat Folder di R Shiny (Sumber: Penulis)

3. Kemudian klik “Create” sehingga layar R Studio akan memberikan tampilan sebagai berikut,



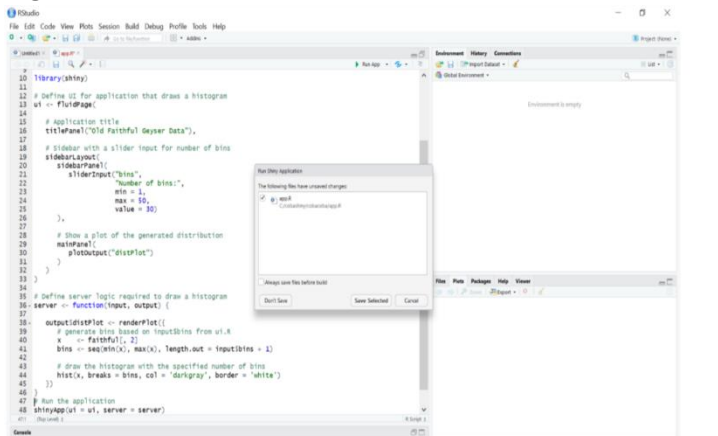
Gambar 3.91 Tampilan R Studio Setelah Menggunakan R Shiny (Sumber: Penulis)

4. Selanjutnya buat *code* untuk menampilkan histogram



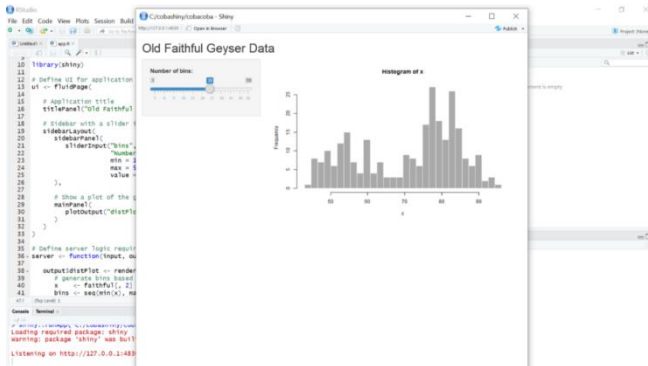
Gambar 3.92 Langkah Untuk Menampilkan Histogram (Sumber: Penulis)

5. Jika kita klik “*Run App*” maka koding akan diproses dengan hasil sebagai berikut



Gambar 3.93 Hasil Proses Coding (Sumber: Penulis)

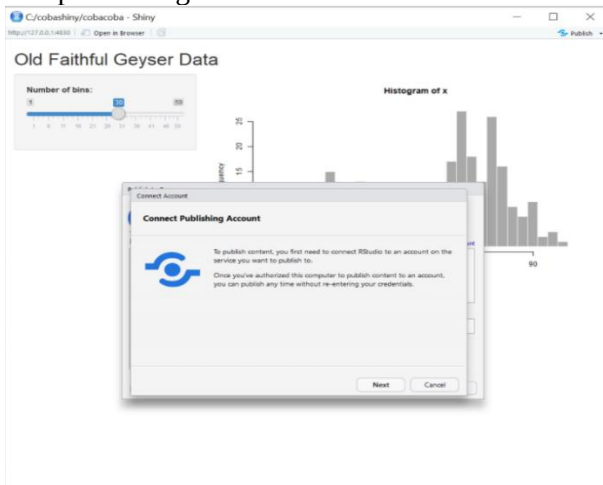
Klik “*Save Selected*” dan terlihat sebuah *file* “*app.R*” (koding yang disimpan dalam file sebaiknya diberi nama “*app.R*”) telah terbentuk, kemudian akan muncul hasil



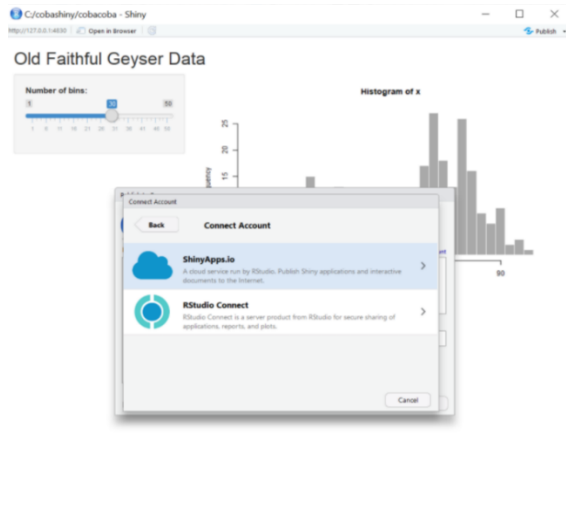
Gambar 3.94 Tampilan Histogram Interaktif (Sumber: Penulis)

Maka akan muncul histogram interaktif. Dari sini tampak bahwa tugas dari “ui” adalah tampilan depan aplikasi (*website*) sebagai *user interface* dan tugas dari “server” adalah dapur belakang pemrosesan dan pemodelan.

6. Jika ingin di publikasikan, maka klik *publish* pada pojok kanan, dan muncul tampilan sebagai berikut



Gambar 3.95 Tampilan Untuk Melakukan Publikasi (Sumber: Penulis)
dan muncul tampilan,



Gambar 3.96 Tampilan Untuk Melakukan Publikasi (Lanjutan)
(Sumber: Penulis)

7. Ikuti langkah-langkah yang dipaparkan dalam setiap jendela kecil yang muncul sampai nanti dapat di *publish*.

Latihan Soal

Buatlah aplikasi statistika deskriptif sederhana (sum, average, standard deviation, variance) menggunakan R Shiny

Latihan Soal

- Buka website <https://archive.ics.uci.edu/ml/index.php>
- *Download* sembarang data, kemudian baca di Excel, Minitab dan buat aplikasi menggunakan R Shiny.

3.13 R Shiny

BAGIAN IV

KOMPUTASI STATISTIKA LANJUT

Capaian Pembelajaran: Setelah mempelajari bagian ini mahasiswa diharapkan mampu:

1. Menerapkan algoritma membangkitkan bilangan random
2. Menerapkan metode Monte Carlo
3. Melakukan estimasi dengan metode Jackknife dan Bootstrap

4.1 Metode untuk Membangkitkan Variabel Acak

Kemampuan untuk mensimulasikan variabel acak dari distribusi tertentu merupakan salah satu fondasi dalam komputasi statistik. Dalam banyak kasus, kita perlu menghasilkan variabel acak untuk berbagai keperluan, seperti simulasi, analisis Monte Carlo, atau untuk mengeksplorasi sifat-sifat distribusi statistik yang kompleks.

Meskipun R menyediakan generator untuk banyak distribusi ini (misalnya, `rbeta`, `rgeom`, `rchisq`, dan lainnya), metode yang akan dijelaskan di bawah ini bersifat umum dan dapat diterapkan pada berbagai jenis distribusi lainnya.

4.1.1 Metode Transformasi Invers

Jika dimisalkan X adalah sebuah variabel acak kontinu yang memiliki fungsi distribusi kumulatif (CDF) $F_X(x)$. Berdasarkan transformasi integral probabilitas, kita dapat menyatakan bahwa variabel acak $U = F_X(X)$ akan mengikuti distribusi uniform pada interval $(0, 1)$.

Metode transformasi invers untuk menghasilkan variabel acak menggunakan prinsip ini. Transformasi invers dapat dinyatakan sebagai berikut:

$$F_X^{-1}(u) = \inf\{x: F_X(x) = u\}, 0 < u < 1$$

Di mana, jika U merupakan variabel acak yang mengikuti distribusi Uniform(0, 1), maka untuk setiap $x \in R$:

$$\begin{aligned} P(F_X^{-1}(U) \leq x) &= P(\inf\{t: F_X(t) = U\} \leq x) \\ &= P(U \leq F_X(x)) = F_U(F_X(x)) = F_X(x) \end{aligned}$$

Dengan kata lain, $F_X^{-1}(U)$ akan memiliki distribusi yang sama dengan X . Untuk menghasilkan sebuah observasi acak X , kita pertama-tama

menghasilkan angka acak u dari distribusi seragam pada interval $(0,1)$, lalu menghitung nilai $F_X^{-1}(u)$. Metode ini sangat berguna dan dapat diterapkan baik untuk variabel acak kontinu maupun diskrit, selama F_X^{-1} dapat dihitung dengan mudah.

4.1.2 Aplikasi dalam Distribusi Kontinu

Misalnya, untuk distribusi eksponensial dengan parameter λ , CDF-nya adalah:

Dalam contoh ini, kita akan menggunakan metode transformasi invers untuk mensimulasikan sampel acak dari distribusi dengan fungsi densitas $f_X(x) = 3x^2$ pada interval $0 < x < 1$.

Langkah-langkah yang perlu dilakukan adalah sebagai berikut:

1. Menentukan Fungsi Distribusi Kumulatif (CDF): Untuk distribusi dengan densitas $f_X(x) = 3x^2$, kita dapat menghitung fungsi distribusi kumulatif (CDF) $F_X(x)$ sebagai:

$$F_X(x) = x^3 \quad \text{untuk } 0 < x < 1$$

Oleh karena itu, fungsi invers dari CDF, $F^{-1}(u)$, adalah:

$$F^{-1}(u) = u^{1/3}$$

2. Menghasilkan Sampel Acak: Untuk menghasilkan sampel acak dari distribusi ini, kita pertama-tama menghasilkan angka acak u yang tersebar secara uniform pada interval $[0, 1]$. Kemudian, kita menerapkan fungsi $F^{-1}(u)$ untuk mendapatkan sampel x .

Berikut adalah implementasi dalam bentuk R:

```
> n <- 1000
> u <- runif(n)
> x <- u(1/3) # Sampel dari distribusi yang diinginkan
> hist(x, prob = TRUE, main = "Histogram dan Kurva Densitas", xlab =
"x", ylab = "Densitas")
> y <- seq(0, 1, .01)
> lines(y, 3 * y2, col = "blue", lwd = 2)
```

4.1.3 Aplikasi dalam Distribusi Diskrit

Contoh ini menerapkan metode transformasi invers untuk menghasilkan sampel acak dari variabel Bernoulli dengan parameter $p = 0,4$. Meskipun terdapat metode yang lebih sederhana untuk menghasilkan distribusi dua titik di R, contoh ini berguna untuk mengilustrasikan cara

menghitung invers dari fungsi distribusi kumulatif (CDF) variabel acak diskrit dalam kasus yang paling sederhana.

Pada kasus ini, fungsi distribusi kumulatif $F_X(x)$ didefinisikan sebagai berikut:

- $F_X(0) = P(X = 0) = 1 - p$
- $F_X(1) = P(X = 1) = 1$

Oleh karena itu, fungsi invers dari CDF $F^{-1}(u)$ dapat dinyatakan sebagai:

- $F^{-1}(u) = 0$ jika $u \leq 0,6$
- $F^{-1}(u) = 1$ jika $u > 0,6$

Generator dalam contoh ini akan memberikan nilai numerik dari ekspresi logis $u > 0,6$.

Berikut adalah implementasi dalam bentuk R:

```
> n <- 1000
> p <- 0.4
> u <- runif(n)
> x <- as.integer(u > 0.6) # Menghasilkan sampel Bernoulli
> mean(x)
[1] 0.41
> var(x)
[1] 0.2421421
```

4.1.4 Metode *Acceptance-Rejection*

Misalkan X dan Y adalah variabel acak dengan fungsi densitas (pmf) adalah f dan g . Jika terdapat konstanta c sehingga $f(t) \leq c \cdot g(t)$ untuk semua t dimana $f(t) > 0$, maka metode ini dapat diterapkan untuk membangkitkan variabel acak X .

Langkah-langkah Metode *Acceptance-Rejection* dapat dijelaskan sebagai berikut:

1. Pilih Variabel Acak Y : Tentukan variabel acak Y dengan fungsi densitas g yang memenuhi $\frac{f(t)}{g(t)} \leq c$ untuk semua t dimana $f(t) > 0$. Sediakan metode untuk menghasilkan Y secara acak.
2. Prosedur untuk Setiap Variabel Acak:
 - a) Hasilkan nilai acak y dari distribusi dengan fungsi densitas g .
 - b) Hasilkan nilai acak u dari distribusi Uniform(0, 1).

- c) Jika $u < \frac{f(y)}{c \cdot g(y)}$, terima y dan gunakan $x = y$; jika tidak, tolak y dan ulangi dari Langkah 2 a).

Perhatikan bahwa pada Langkah 2 c) kesetaraan terakhir ini hanya mengevaluasi CDF dari U :

$$P(\text{accepted}|Y) = P\left(U < \frac{f(Y)}{c \cdot g(Y)} \middle| Y\right) = \frac{f(Y)}{c \cdot g(Y)}.$$

Contoh dibawah ini mengilustrasikan Metode *Acceptance-Rejection* untuk distribusi beta.

- Berapa rata-rata angka acak yang harus disimulasikan untuk menghasilkan 1000 variabel dari distribusi Beta($\alpha = 2, \beta = 2$) menggunakan metode ini? Hal ini bergantung pada batas atas c dari $\frac{f(x)}{g(x)}$, yang tergantung pada pilihan fungsi $g(x)$.

Fungsi densitas Beta(2,2) adalah $f(x) = 6x(1-x)$ untuk $0 < x < 1$. Misalkan $g(x)$ adalah fungsi densitas Uniform(0,1). Maka, $\frac{f(x)}{g(x)} \leq 6$ untuk semua $0 < x < 1$, sehingga $c = 6$. Sebuah nilai acak x dari $g(x)$ diterima jika:

$$\frac{f(x)}{c \cdot g(x)} = \frac{6x(1-x)}{6} = x(1-x) > u.$$

Pada umumnya, diperlukan $cn = 6000$ iterasi (12000 angka acak) digunakan untuk menghasilkan ukuran sampel sebanyak 1000. Dalam simulasi berikut, j untuk iterasi tidak terlalu diperlukan, tetapi akan disertakan untuk mencatat berapa banyak iterasi yang sebenarnya diperlukan untuk menghasilkan 1000 variabel beta.

```
n <- 1000
k <- 0 #counter untuk yang diterima
j <- 0 #jumlah iterasi
y <- numeric(n)
while (k < n) {
  u <- runif(1)
  j <- j + 1
  x <- runif(1) #nilai acak dari distribusi g
  if (x * (1-x) > u) {
    #terima x
    k <- k + 1
    y[k] <- x
  }
}
```

```

}
}
print(j)
[1] 5873

```

Pada simulasi ini, dibuktikan sebanyak 5873 iterasi (11746 angka acak) diperlukan untuk menghasilkan 1000 variabel beta.

4.1.5 Metode Transformasi

Berbagai teknik transformasi digunakan untuk menghasilkan variabel acak dari distribusi tertentu. Teknik-teknik ini sering kali digunakan ketika metode lain, seperti metode inverse transform, tidak praktis atau sulit diimplementasikan. Berikut adalah beberapa metode transformasi yang umum digunakan:

1. Jika Z mengikuti distribusi $N(0,1)$, maka $V = Z^2$ memiliki distribusi $\chi^2(1)$.
2. Jika U memiliki distribusi $\chi^2(m)$ dan V memiliki distribusi $\chi^2(n)$ yang keduanya independen, maka $F = \frac{U/m}{V/n}$ mengikuti distribusi F dengan derajat kebebasan (m, n) .
3. Jika Z mengikuti distribusi $N(0,1)$ dan V memiliki distribusi $\chi^2(n)$, serta keduanya independen, maka $T = \frac{Z}{\sqrt{V/n}}$ mengikuti distribusi t-Student dengan n derajat kebebasan.
4. Jika U dan V keduanya mengikuti distribusi $Unif(0,1)$ dan bersifat independen, maka

$$Z_1 = \sqrt{-2 \log U} \cos(2\pi V),$$

$$Z_2 = \sqrt{-2 \log U} \sin(2\pi V),$$

akan menghasilkan dua variabel acak normal standar yang independen.

5. Jika U mengikuti distribusi Gamma (r, λ) dan V mengikuti distribusi Gamma (s, λ) , serta keduanya independen, maka $X = \frac{U}{U+V}$ mengikuti distribusi Beta (r, s) .
6. Jika U dan V keduanya mengikuti distribusi $Unif(0,1)$ dan bersifat independen, maka

$$X = \left\lfloor 1 + \frac{\log V}{\log(1 - (1 - \theta)U)} \right\rfloor$$

akan mengikuti distribusi logaritmik dengan parameter θ , di mana $[x]$ merupakan bagian integer dari x .

Berikut adalah implementasi poin (5) dalam bentuk R untuk menghasilkan variabel acak Beta(a , b).

1. Bangkitkan bilangan acak u dari distribusi Gamma(a , 1).
2. Bangkitkan bilangan acak v dari distribusi Gamma(b , 1).
3. Hitung dan keluarkan nilai $x = \frac{u}{u+v}$

Metode ini diterapkan untuk menghasilkan sampel acak dari pemisalan distribusi Beta(3, 2).

```
n <- 1000
a <- 3
b <- 2
u <- rgamma(n, shape=a, rate=1)
v <- rgamma(n, shape=b, rate=1)
x <- u / (u + v)
```

Data sampel ini dapat dibandingkan dengan distribusi Beta(3, 2) menggunakan plot kuantil-kuantil (QQ plot). Jika distribusi sampel sesuai dengan Beta(3, 2), plot QQ tersebut harus mendekati linear. Garis identitas $x = q$ ditambahkan ke dalam plot sebagai referensi.

```
q <- qbeta(ppoints(n), a, b)
qqplot(q, x, cex=0.25, xlab="Beta(3, 2)", ylab="Sample")
abline(0, 1)
```

Berikut adalah implementasi poin (6) dalam bentuk R untuk Distribusi logaritmik $Log(\theta)$:

1. Hasilkan dua bilangan acak independen U dan V dari distribusi Uniform(0,1).
2. Hitung nilai X sebagai berikut:

$$X = \left\lfloor \frac{1 + \log V}{\log(1 - (1 - \theta)U)} \right\rfloor$$

Di mana $[x]$ menunjukkan bagian integer dari x . Transformasi ini memberikan distribusi logaritmik dengan parameter θ .

Metode ini diterapkan untuk membandingkan distribusi Logaritmik (0.5) dengan sampel yang dihasilkan menggunakan transformasi (3.3):

```
n <- 1000
theta <- 0.5
u <- runif(n) #membangkitkan sampel logaritmik
```

```
v <- runif(n)
x <- floor(1 + log(v) / log(1 - (1 - theta)^u))
k <- 1:max(x)
p <- -1 / log(1 - theta) * theta^k / k
se <- sqrt(p*(1-p)/n)
p.hat <- tabulate(x)/n
> print(round(rbind(p.hat, p, se), 3))
```

4.2 Integrasi Monte Carlo

Misalkan suatu fungsi $g(x)$, dengan tujuan menghitung nilai integral $\int_a^b g(x) dx$. Kita asumsikan bahwa integral ini memang ada atau dapat dihitung. Mengingat jika X adalah sebuah variabel acak dengan fungsi densitas $f(x)$, maka nilai harapan (rata-rata yang diharapkan) dari variabel acak Y , yang didefinisikan sebagai $Y = g(X)$, adalah:

$$E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x) dx.$$

Jika didapatkan sampel acak dari distribusi X , kita dapat memperkirakan nilai $E[g(X)]$ dengan menghitung rata-rata dari sampel tersebut.

4.2.1 Estimator Sederhana Monte Carlo

Masalah utama dalam mengestimasi nilai dari $\theta = \int_0^1 g(x) dx$ adalah jika $X_1, X_2, \dots, X_m$ merupakan sampel acak yang diambil dari distribusi seragam Uniform(0,1), sehingga estimator Monte Carlo untuk θ dapat ditulis sebagai:

$$\hat{\theta} = \overline{g_m(X)} = \frac{1}{m} \sum_{i=1}^m g(X_i)$$

Berdasarkan *Strong Law of Large Numbers*, estimator $\hat{\theta}$ akan konvergen ke $E[g(X)] = \theta$ dengan probabilitas 1.

Contoh soal 1:

- Hitung estimasi Monte Carlo untuk $\theta = \int_0^1 e^{-x} dx$ dan bandingkan hasil estimasi tersebut dengan nilai eksak.

```
m <- 10000
x <- runif(m)
theta.hat <- mean(exp(-x))
```

```
print(theta.hat)
print(1 - exp(-1))
[1] 0.6355289
[1] 0.6321206
```

Didapatkan berdasarkan estimasi menghasilkan $\hat{\theta} = 0.6355$, sementara nilai eksak dari $\theta = 1 - e^{-1} \approx 0.6321$.

Contoh soal 2:

3. Hitung estimasi Monte Carlo untuk $\theta = \int_2^4 e^{-x} dx$ dan bandingkan hasil estimasi tersebut dengan nilai eksak.

```
m <- 10000
x <- runif(m, min = 2, max = 4)
theta.hat <- mean(exp(-x)) * 2
print(theta.hat)
print(exp(-2) - exp(-4))
[1] 0.1172158
[1] 0.1170196
```

Didapatkan berdasarkan estimasi menghasilkan $\hat{\theta} = 0.1172$, sementara nilai eksak dari $\theta = 1 - e^{-1} \approx 0.1170$.

4.2.2 Variance Reduction

Integrasi Monte Carlo sering digunakan untuk mengestimasi nilai ekspektasi dari suatu fungsi, dituliskan sebagai $E[g(X)]$. Terdapat beberapa teknik untuk mengurangi varians dalam proses estimasi ini, yang bertujuan untuk meningkatkan keakuratan estimasi tanpa perlu meningkatkan jumlah sampel secara signifikan.

Misalkan terdapat dua estimator, $\hat{\theta}_1$ dan $\hat{\theta}_2$, yang keduanya mengestimasi parameter θ . Jika varians dari estimator kedua lebih kecil daripada varians dari estimator pertama, $\text{Var}(\hat{\theta}_2) < \text{Var}(\hat{\theta}_1)$, maka persentase pengurangan varians yang diperoleh dengan menggunakan estimator kedua dibandingkan dengan yang pertama dituliskan sebagai berikut:

$$\left(\frac{\text{Var}(\hat{\theta}_1) - \text{Var}(\hat{\theta}_2)}{\text{Var}(\hat{\theta}_1)} \right) \times 100$$

Untuk mengestimasi $\theta = E[g(X)]$ dengan pendekatan Monte Carlo, kita menghitung rata-rata sampel dari $g(X)$ berdasarkan sejumlah besar replika m yang diambil dari distribusi fungsi tersebut.

Misalkan kita memiliki sampel independen $X_{j1}, \dots, X_{jn}$ untuk $j = 1, \dots, m$, kemudian kita hitung replika $Y_j = g(X_{j1}, \dots, X_{jn})$. Karena $Y_1, \dots, Y_m$ adalah sampel independen dengan distribusi yang sama, nilai harapannya adalah:

$$E[\bar{Y}] \left[\frac{1}{m} \sum_{j=1}^m Y_j \right] = \theta$$

Dengan demikian, estimator Monte Carlo $\hat{\theta} = \bar{Y}$ tidak bias terhadap $\theta = E[Y]$. Varians dari estimator Monte Carlo:

$$\text{Var}(\hat{\theta}) = \text{Var}\bar{Y} = \frac{\text{Var}_f(g(X))}{m}$$

Meningkatkan jumlah sampel m akan mengurangi varians, namun dengan biaya komputasi yang meningkat. Sebagai contoh, untuk mengurangi kesalahan standar dari 0,01 menjadi 0,0001, kita memerlukan sekitar 10.000 kali jumlah sampel. Dalam situasi seperti ini, kita sering kali perlu mencari metode lain yang lebih efisien untuk mengurangi varians tanpa menambah biaya komputasi secara signifikan.

4.3 Bootstrap dan Jackknife

Bootstrap dan Jackknife adalah dua teknik resampling yang sering digunakan dalam statistika untuk mengestimasi ketidakpastian (misalnya, varians, kesalahan standar, dan interval kepercayaan) dari suatu estimasi parameter. Kedua metode ini sangat bermanfaat ketika analisis teoretis varians dan distribusi sampling sulit atau tidak mungkin dilakukan secara langsung. Keduanya bergantung pada ide dasar pengambilan sampel ulang dari data yang tersedia untuk mensimulasikan distribusi sampling.

4.3.1 Bootstrap

Metode bootstrap adalah bagian dari metode non-parametrik Monte Carlo yang digunakan untuk mengestimasi distribusi populasi melalui teknik resampling. Metode resampling memperlakukan sampel yang diamati sebagai populasi terbatas, dan sampel acak diambil dari sampel ini untuk

mengestimasi karakteristik populasi dan membuat inferensi tentang populasi yang disampling. Metode bootstrap sering digunakan ketika distribusi populasi target tidak diketahui; sampel yang tersedia menjadi satu-satunya informasi yang ada.

Estimasi bootstrap dari distribusi sampling mirip dengan konsep estimasi densitas. Jika metode lain menghasilkan sampel acak dari densitas yang ditentukan sepenuhnya, bootstrap menghasilkan sampel acak dari distribusi empiris sampel tersebut.

Misalkan $x = (x_1, x_2, \dots, x_n)$ adalah sampel acak yang diamati dari suatu distribusi dengan CDF $F(x)$. Jika X^* dipilih secara acak dari x , maka:

$$P(X^* = x_i) = \frac{1}{n}, \quad i = 1, \dots, n.$$

Resampling menghasilkan sampel acak $X_1^* X_2^* \dots, X_n^*$ dengan mengambil sampel dengan penggantian dari x . Variabel acak X_i^* adalah identik dan independen, terdistribusi secara seragam pada himpunan $\{x_1, x_2, \dots, x_n\}$.

Contoh soal 1: *Bootstrap estimate of standard error*

Data yang digunakan dalam contoh ini berasal dari *package bootstrap* pada aplikasi R. Data ini mencakup skor LSAT (rata-rata skor ujian masuk sekolah hukum) dan GPA dari 15 sekolah hukum.

4. LSAT: 576, 635, 558, 578, 666, 580, 555, 661, 651, 605, 653, 575, 545, 572, 594
5. GPA: 339, 330, 281, 303, 344, 307, 300, 343, 336, 313, 312, 274, 276, 288, 296

Contoh data yang digunakan merupakan sampel acak dari total 82 sekolah hukum yang tercakup dalam dataset law82 dari paket bootstrap. Tujuan kita adalah mengestimasi korelasi antara skor LSAT dan GPA, serta menghitung estimasi bootstrap untuk *standard error* dari korelasi sampel.

1. Untuk setiap replika bootstrap, dengan indeks $b = 1, \dots, B$:
 - a) Bangkitkan sampel bootstrap $x^{(b)} = (x_1', \dots, x_n')$ dengan pengambilan sampel berulang (dengan penggantian) dari sampel yang diamati $x_1, \dots, x_n$.
 - b) Lakukan perhitungan replika b^{th} , $\hat{\theta}(b)$, dari sampel bootstrap, di mana $\hat{\theta}$ adalah korelasi sampel R antara (LSAT, GPA).
2. Estimasi bootstrap untuk *standard error* (R) adalah simpangan baku sampel dari replika $\hat{\theta}(1), \dots, \hat{\theta}(B) = R(1), \dots, R(B)$.

```
library(bootstrap)
print(cor(law$LSAT, law$GPA))
[1] 0.7763745
print(cor(law82$LSAT, law82$GPA))
[1] 0.7599979
```

Korelasi pada sampel adalah $R = 0.7763745$. Korelasi untuk populasi yang terdiri dari 82 sekolah hukum adalah $R = 0.7599979$. Selanjutnya akan digunakan metode *bootstrap* untuk mengestimasi *standard error* dari statistik korelasi yang dihitung dari sampel skor dalam *dataset law*

```
B <- 200 # jumlah replika bootstrap
n <- nrow(law) # ukuran sampel
R <- numeric(B) # penyimpanan untuk replica
# Estimasi bootstrap dari kesalahan standar R
for (b in 1:B) {
  i <- sample(1:n, size = n, replace = TRUE)
  LSAT <- law$LSAT[i]
  GPA <- law$GPA[i]
  R[b] <- cor(LSAT, GPA)
}
print(se.R <- sd(R)) # hitung simpangan baku dari replika
[1] 0.1358393
hist(R, prob = TRUE) # menampilkan histogram distribusi
```

Diketahui estimasi bootstrap untuk $se(R)$ adalah 0.1358393. Sedangkan diketahui estimasi *standard error* berdasarkan teori normal adalah 0.115. Nilai estimasi bootstrap sedikit lebih tinggi daripada estimasi berdasarkan teori normal. Hal ini dapat menunjukkan bahwa asumsi distribusi normal tidak sepenuhnya cocok untuk data ini, dan metode bootstrap dapat memberikan estimasi yang lebih realistis.

Contoh soal 1a: *Bootstrap estimate of standard error (boot)*

Implementasi dibawah ini menunjukkan alternatif dari contoh diatas tetapi menggunakan fungsi boot dari *package boot* di R. Jika data adalah x dan vektor indeks adalah i , dibutuhkan $x[i, 1]$ untuk mengekstrak variabel resampling pertama dan $x[i, 2]$ untuk variabel kedua.

```
r <- function(x, i) {
  # Menghitung korelasi antara kolom 1 dan 2
  cor(x[i, 1], x[i, 2])
}
```



```

}
library(boot)
obj <- boot(data = law, statistic = r, R = 2000)
print(obj)
ORDINARY NONPARAMETRIC BOOTSTRAP

```

Call: boot(data = law, statistic = r, R = 2000)

Bootstrap Statistics :

| | original | bias | std. error |
|-----|-----------|--------------|------------|
| t1* | 0.7763745 | -0.004795305 | 0.1303343 |

4.3.2 Jackknife

Metode *jackknife* mirip dengan teknik “leave-one-out” dalam *cross-validation*. Misalkan $x = (x_1, x_2, \dots, x_n)$ adalah sampel acak yang diamati, dan didefinisikan sampel *jackknife ke $-i$* , $x_{(i)}$, sebagai subset dari x menghilangkan pengamatan *ke $-i$* yaitu x_i . Dengan kata lain:

$$x_{(i)} = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n).$$

Jika $\hat{\theta} = T(x)$, maka definisi replika jackknife ke- i , $\hat{\theta}(i) = T(x_{(i)})$, $i = 1, \dots, n$. Misalkan parameter $\theta = t(F)$ adalah fungsi dari distribusi F . Biarkan F_n menjadi distribusi empiris kumulatif (ecdf) dari suatu sampel acak dari distribusi F . Estimasi “plug-in” dari θ adalah $\hat{\theta} = t(F_n)$.

4.3.3 Pengaplikasian Cross

Validasi silang atau *cross validation* adalah metode pembagian data yang dapat digunakan untuk menilai stabilitas estimasi parameter, akurasi dalam klasifikasi algoritma, dan berbagai aplikasi lainnya. Jackknife dapat dianggap sebagai kasus khusus dari validasi silang, karena metode ini digunakan untuk mengestimasi bias dan *standard error* suatu estimator.

Versi lain dari validasi silang adalah “n-fold cross validation,” yang membagi data menjadi n himpunan uji (atau disebut juga titik uji). Prosedur “leave-one-out” ini mirip dengan metode jackknife. Data dapat dibagi menjadi sejumlah partisi K , sehingga terdapat K himpunan uji. Kemudian, fitting model dilakukan dengan mengabaikan satu himpunan uji pada setiap giliran, sehingga model di-fitting sebanyak K kali.

Contoh soal 1: Pemilihan model

Data *ironslag* dari *package* DAAG memiliki 53 pengukuran kandungan besi menggunakan dua metode, yaitu metode kimia dan magnetik.

Ada beberapa langkah dalam pemilihan model, tetapi kita akan fokus pada kesalahan prediksi. Kesalahan prediksi dapat diestimasi melalui validasi silang, tanpa membuat asumsi distribusi yang kuat tentang variabel kesalahan. Model yang diusulkan untuk memprediksi pengukuran magnetik (Y) berdasarkan pengukuran kimia (X) adalah:

1. Linier: $Y = \beta_0 + \beta_1 X + \epsilon$.
2. Kuadratik: $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \epsilon$.
3. Eksponensial: $\log(Y) = \log(\beta_0) + \beta_1 X + \epsilon$.
4. Log-Log: $\log(Y) = \beta_0 + \beta_1 \log(X) + \epsilon$.

Berikut adalah implementasi untuk masing-masing model dalam bentuk R:

```
library(DAAG)
attach(ironslag)
a <- seq(10, 40, .1) # urutan untuk plotting
```

```
L1 <- lm(magnetic ~ chemical)
plot(chemical, magnetic, main="Linear", pch=16)
yhat1 <- L1$coef[1] + L1$coef[2] * a
lines(a, yhat1, lwd=2)
```

```
L2 <- lm(magnetic ~ chemical + I(chemical^2))
plot(chemical, magnetic, main="Quadratic", pch=16)
yhat2 <- L2$coef[1] + L2$coef[2] * a + L2$coef[3] * a^2
lines(a, yhat2, lwd=2)
```

```
L3 <- lm(log(magnetic) ~ chemical)
plot(chemical, magnetic, main="Exponential", pch=16)
logyhat3 <- L3$coef[1] + L3$coef[2] * a
yhat3 <- exp(logyhat3)
lines(a, yhat3, lwd=2)
```

```
L4 <- lm(log(magnetic) ~ log(chemical))
```

```
plot(log(chemical), log(magnetic), main="Log-Log", pch=16)
logyhat4 <- L4$coef[1] + L4$coef[2] * log(a)
lines(log(a), logyhat4, lwd=2)
```

Setelah mendapatkan hasil model estimasi, *cross validation* dapat digunakan untuk mengestimasi kesalahan prediksi.

1. Untuk $k = 1, \dots, n$, ambil observasi (x_k, y_k) sebagai titik uji dan gunakan observasi sisanya untuk memasang model.
 - a) Pasang model menggunakan hanya $n - 1$ observasi dalam himpunan pelatihan, (x_i, y_i) , dengan $i \neq k$.
 - b) Hitung respons prediksi $\widehat{y}_k = \widehat{\beta}_0 + \widehat{\beta}_1 x_k$.
 - c) Hitung kesalahan prediksi $e_k = y_k - \widehat{y}_k$.
2. Estimasi rata-rata dari kuadrat kesalahan prediksi:

$$\widehat{\sigma\epsilon^2} = \frac{1}{n} \sum_k^n 1 e_k^2$$

```
n <- length(magnetic)
e1 <- e2 <- e3 <- e4 <- numeric(n) # menyimpan kesalahan prediksi untuk
cross validation n-fold
```

pasang model pada sampel *leave-one-out*

```
for (k in 1:n) {
  y <- magnetic[-k]
  x <- chemical[-k]
```

Model Linear

```
J1 <- lm(y ~ x)
yhat1 <- J1$coef[1] + J1$coef[2] * chemical[k]
e1[k] <- magnetic[k] - yhat1
```

Model Kuadratik

```
J2 <- lm(y ~ x + I(x^2))
yhat2 <- J2$coef[1] + J2$coef[2] * chemical[k] + J2$coef[3] *
chemical[k]^2
e2[k] <- magnetic[k] - yhat2
```

Model Eksponensial

```
J3 <- lm(log(y) ~ x)
logyhat3 <- J3$coef[1] + J3$coef[2] * chemical[k]
yhat3 <- exp(logyhat3)
```

```
e3[k] <- magnetic[k] - yhat3
```

```
# Model Log-Log
```

```
J4 <- lm(log(y) ~ log(x))
```

```
logyhat4 <- J4$coef[1] + J4$coef[2] * log(chemical[k])
```

```
yhat4 <- exp(logyhat4)
```

```
e4[k] <- magnetic[k] - yhat4
```

```
}
```

Estimasi kesalahan prediksi yang diperoleh dari *cross validation n-fold* adalah sebagai berikut:

```
c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))
```

```
[1] 19.55644 17.85248 18.44188 20.45424
```

Model kuadratik (Model 2) memiliki nilai rata-rata kuadrat kesalahan prediksi terendah, yaitu sebesar 17.85248 oleh karena itu, model kuadratik memberikan prediksi yang lebih akurat dibandingkan model lainnya.

TENTANG PENULIS



Wahyu Wibowo menyelesaikan pendidikan sarjana di Institut Teknologi Sepuluh Nopember (ITS), Surabaya pada program studi Statistika. Selanjutnya pendidikan magister dan doktor pada program studi Matematika diselesaikan di Universitas Gadjah Mada Yogyakarta. Kariernya dimulai semenjak tahun 1998 sebagai dosen ITS di Departemen Statistika, kemudian mulai tahun 2017 karena kebijakan kampus, berpindah ke Departemen Statistika Bisnis Fakultas Vokasi. Selama menjadi dosen, banyak kegiatan Tri Dharma perguruan tinggi yang dilakukan hingga di tahun 2023 diangkat sebagai Guru Besar pada bidang Pemodelan Regresi di Departemen Statistika Bisnis.



Muhammad Alifian Nuriman menyelesaikan pendidikan sarjana dan magister pada program studi Statistika, Institut Teknologi Sepuluh Nopember, Surabaya. Penulis adalah dosen tetap di program studi sarjana terapan Statistika Bisnis, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya. Penulis mengampu mata kuliah komputasi statistika, eksplorasi dan visualisasi data (EVD), riset operasi, manajemen rantai pasok, dan statistika pengendalian kualitas. Penulis juga aktif menulis artikel yang di publikasikan di jurnal nasional terindeks dan prosiding internasional terindeks dengan bidang keahlian statistika industri. Saat ini penulis sedang melanjutkan studi doctoral dengan topik penelitian *statistical process control* di Universiti Sains Malaysia (USM), Penang.

DAFTAR PUSTAKA

- R Core Team. (2020). *An Introduction to R*.
- Scherber, C. (2007). *An introduction to statistical data analysis using R*.
R_Manual Goettingen
- Venables, D. M., & Smith, D. M. (2009). *An Introduction to R*. Network
Theory Ltd.
- Wickham, H., & Golemund, G. (2016). *R for Data Science*. Canada:
O'Reilly.
- Rizzo, M.L. (2019) *Statistical computing with R*. 2nd edn. Boca Raton, New
York: CRC Press, Taylor & Francis Group.