

MODUL PRAKTIKUM

ALGORITMA PEMROGRAMAN VI231208

- Level.h
- Level :: L
- Level :: O
- Level :: O
- Level :: G
- Level :: G

DAFTAR ISI

| | |
|--|-----------|
| DAFTAR ISI | 2 |
| PENDAHULUAN | 3 |
| Aturan Kerja Lab Komputer | 4 |
| Panduan Berkegiatan di Laboratorium | 5 |
| TEKNIS PRAKTIKUM | 7 |
| Prosedur Penggunaan Alat | 7 |
| PROSEDUR KEDATANGAN | 9 |
| Kebutuhan saat di LabKom | 9 |
| Prosedur kedatangan di LabKom | 9 |
| PROSEDUR PRAKTIKUM | 10 |
| P1 ALGORITMA PEMROGRAMAN MASM DAN TASM | 10 |
| P2 ALGORITMA PEMROGRAMAN ASSEMBLER SIMULATOR | 11 |
| P3 ALGORITMA PEMROGRAMAN OBJECT ORIENTED PROGRAM | 13 |
| P4 ALGORITMA PEMROGRAMAN XAMARIN | 17 |
| P5 ALGORITMA PEMROGRAMAN BIG DATA | 24 |
| Lampiran Permit to Work | 30 |
| Lampiran JSA (Job Safety Analysis) | 31 |

PENDAHULUAN

Mata kuliah Algoritma Pemrograman di semester II (3 SKS) bertujuan untuk membekali mahasiswa dengan pemahaman mendalam tentang berbagai jenis bahasa pemrograman, mulai dari bahasa tingkat rendah hingga tingkat tinggi. Mahasiswa akan mempelajari prinsip dasar dan lanjutan pemrograman dengan bahasa assembly (TASM dan MASM), serta konsep pemrograman tingkat tinggi seperti operasi dasar, array, string, pointer, dan paradigma berbasis objek. Selain itu, mata kuliah ini juga mencakup prinsip pemrograman database, pemrograman berbasis Android Studio, dan pengembangan multi-platform. Dengan cakupan materi yang luas, mahasiswa diharapkan memiliki dasar yang kuat dalam algoritma dan pemrograman untuk berbagai kebutuhan komputasi.

Bahasa rakitan (assembly) adalah bahasa pemrograman tingkat rendah yang digunakan untuk berinteraksi langsung dengan perangkat keras komputer. Berbeda dengan bahasa pemrograman tingkat tinggi, bahasa assembly memungkinkan kontrol lebih rinci terhadap arsitektur prosesor, register, dan instruksi mesin. Dalam praktikum Algoritma Pemrograman, mahasiswa akan mempelajari dasar-dasar bahasa assembly menggunakan Turbo Assembler (TASM) dan Microsoft Macro Assembler (MASM). TASM adalah assembler yang dikembangkan oleh Borland dan sering digunakan untuk pengembangan perangkat lunak berbasis DOS, sementara MASM, yang dikembangkan oleh Microsoft, lebih fleksibel dan mendukung berbagai arsitektur prosesor x86. Dengan memahami bahasa assembly, mahasiswa dapat meningkatkan pemahaman mereka terhadap cara kerja prosesor, manajemen memori, serta optimasi kode, yang merupakan konsep fundamental dalam ilmu komputer dan teknik instrumentasi.

Xamarin adalah framework open-source yang digunakan untuk mengembangkan aplikasi mobile berbasis Android, iOS, dan Windows dengan menggunakan bahasa pemrograman C#. Dikembangkan oleh Microsoft, Xamarin memungkinkan pengembang untuk menulis satu kode dasar (shared codebase) yang dapat dijalankan di berbagai platform, sehingga meningkatkan efisiensi dalam pengembangan aplikasi. Dengan memanfaatkan .NET dan Mono runtime, Xamarin menyediakan akses langsung ke API native masing-masing sistem operasi, memungkinkan performa yang hampir setara dengan aplikasi native. Selain itu, Xamarin Forms mempermudah pembuatan antarmuka pengguna yang dapat digunakan di berbagai platform dengan satu kode yang sama. Keunggulan utama Xamarin adalah kemampuannya dalam integrasi dengan Visual Studio serta dukungan penuh terhadap fitur-fitur native, sehingga cocok digunakan untuk membangun aplikasi lintas platform dengan pengalaman pengguna yang optimal.

Visual Studio adalah Integrated Development Environment (IDE) yang dikembangkan oleh Microsoft untuk mendukung berbagai bahasa pemrograman, termasuk C. IDE ini menyediakan berbagai fitur seperti editor kode yang canggih, debugging, integrasi dengan sistem kontrol versi, serta tools untuk pengembangan aplikasi berbasis desktop, web, dan mobile. Bahasa C sendiri merupakan bahasa pemrograman tingkat menengah yang sering digunakan dalam pengembangan sistem

operasi, embedded system, dan aplikasi dengan kinerja tinggi. Dengan menggunakan Visual Studio, pengembang dapat menulis, menguji, dan mengelola kode C secara lebih efisien berkat fitur-fitur seperti IntelliSense, debugging visual, dan dukungan terhadap berbagai pustaka serta framework.

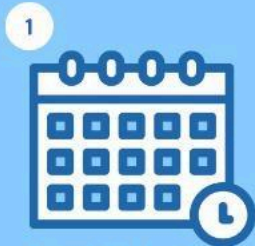
Aturan Kerja Lab Komputer

1. Taatilah Peraturan yang telah ditentukan
2. Mengikuti Safety Induction
3. Ajukan permohonan jika tidak terdapat pengawas kegiatan
4. tanyakan dahulu apakah alat siap untuk digunakan dan telah di maintenance
5. Gunakan APD dengan baik
6. Laksanakan kegiatan sesuai prosedur
7. Memperhatikan benda yang berbahaya
8. Mengelola limbah dengan baik
9. Membersihkan diri setelah kegiatan
10. Mengembalikan alat dan bahan ke tempatnya
11. Menghubungi petugas jika terjadi keadaan darurat

Panduan Berkegiatan di Laboratorium



PANDUAN BERKEGIATAN DI LABORATORIUM



1
Taatilah jadwal yang ditentukan



2
Mengikuti Safety Induction



3
Gunakan APD dengan baik



4
Laksanakan kegiatan sesuai prosedur



5
Memperhatikan benda yang berbahaya



6
Jangan makan & minum di laboratorium



7
Membersihkan diri setelah kegiatan

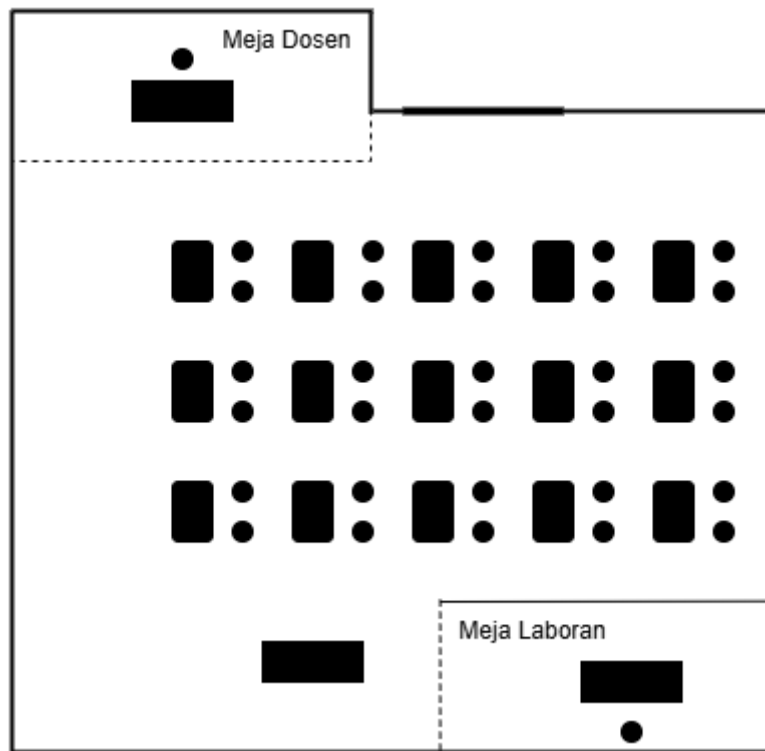


8
Menghubungi petugas jika terjadi kondisi darurat

Let's make our laboratory a safe and fun place for learning!



Denah Lab Komputer



Keterangan :

1. Pintu Masuk

Terletak di sisi kanan atas denah. Ini adalah akses utama untuk keluar-masuk lab komputer.

2. Meja Pengajar (Dosen)

Terletak di bagian depan ruangan, menghadap ke kursi mahasiswa. Meja ini dilengkapi komputer atau perangkat kontrol utama.

3. Meja Komputer Mahasiswa

Tersusun rapi dalam beberapa baris memanjang dari depan ke belakang ruangan. Setiap baris terdiri dari beberapa meja komputer, dipakai untuk masing-masing mahasiswa atau kelompok.

4. Meja Laboran

Berada di sisi kiri bawah denah. Meja ini digunakan oleh laboran untuk membantu operasional praktikum, pengawasan perangkat keras, dan troubleshooting.

TEKNIS PRAKTIKUM

Prosedur Penggunaan Alat

1. Komputer

- a. Nyalakan komputer dengan menekan tombol power pada CPU/laptop.
- b. Tunggu hingga sistem operasi selesai melakukan booting.
- c. Masuk ke akun pengguna jika diperlukan.
- d. Pastikan semua perangkat tambahan (keyboard, mouse, dan monitor) berfungsi dengan baik.
- e. Buka aplikasi yang dibutuhkan sesuai dengan tujuan penggunaan.
- f. Setelah selesai, tutup semua aplikasi dan lakukan shutdown melalui menu Start untuk mematikan komputer dengan aman.

2. TASM

- a. Pastikan TASM telah terinstal di komputer.
- b. Buka Command Prompt (CMD) dan arahkan direktori ke folder tempat TASM berada.
- c. Tulis kode assembly dalam file dengan ekstensi .ASM menggunakan editor teks seperti Notepad++.
- d. Gunakan perintah berikut untuk mengompilasi program:
bash
CopyEdit
tasm namafile.asm
- e. Jika tidak ada error, lanjutkan dengan perintah:
bash
CopyEdit
tlink namafile.obj
- f. Jalankan program dengan mengetikkan:
bash
CopyEdit
namafile.exe
- g. Periksa hasil program, dan lakukan debugging jika diperlukan

3. MASM

- a. Pastikan MASM telah terinstal di komputer.
- b. Tulis kode program assembly dalam file .ASM menggunakan editor teks.
- c. Buka Command Prompt dan arahkan ke direktori tempat file .ASM berada.
- d. Gunakan perintah berikut untuk mengompilasi program:
bash
CopyEdit
ml /c /Zd namafile.asm
- e. Jika tidak ada kesalahan, lanjutkan dengan:

bash
CopyEdit
[link namafile.obj](#)

- f. Jalankan file **.EXE** yang telah dihasilkan untuk melihat outputnya.
- g. Lakukan debugging jika terdapat error atau output tidak sesuai.

4. Xamarin

- a. Pastikan Visual Studio dengan Xamarin telah terinstal.
- b. Buka Visual Studio dan buat proyek baru dengan memilih template Xamarin (Xamarin.Forms atau Xamarin.Android/iOS).
- c. Atur konfigurasi proyek sesuai dengan kebutuhan aplikasi lintas platform.
- d. Tulis kode program menggunakan C# untuk pengembangan aplikasi.
- e. Gunakan emulator atau perangkat fisik untuk menjalankan dan menguji aplikasi.
- f. Debug aplikasi jika ditemukan kesalahan atau bug.
- g. Setelah selesai, kompilasi dan buat file APK/iOS untuk distribusi aplikasi.

5. Visual Studio

- a. Pastikan Visual Studio telah terinstal di komputer.
- b. Buka aplikasi Visual Studio melalui menu Start atau shortcut di desktop.
- c. Pilih "Create a new project" untuk membuat proyek baru atau "Open a project" untuk membuka proyek yang sudah ada.
- d. Pilih bahasa pemrograman yang diinginkan, seperti C, C++, atau C#.
- e. Tulis kode program pada editor Visual Studio.
- f. Gunakan fitur debugging untuk mengecek kesalahan pada program.
- g. Jalankan program dengan menekan tombol "Start" atau menggunakan shortcut F5.
- h. Simpan proyek sebelum keluar dari aplikasi.

6. Xampp

- a. Pastikan Visual Xampp telah terinstal di komputer.
- b. Buka aplikasi Xampp melalui menu Start atau shortcut di desktop.
- c. Pada XAMPP Control Panel, aktifkan modul Apache dan MySQL dengan menekan tombol Start
- d. Pastikan status modul berubah menjadi Running dan tidak ada pesan error
- e. Letakkan file proyek web (HTML, PHP, dsb.) di folder htdocs yang berada dalam direktori instalasi XAMPP (biasanya di C:\xampp\htdocs).
- f. Buka browser dan akses localhost/namafolder untuk menjalankan proyek.
- g. Gunakan phpMyAdmin (bisa diakses lewat localhost/phpmyadmin) untuk mengelola database MySQL.
- h. Setelah selesai, hentikan modul Apache dan MySQL dengan menekan tombol Stop.
- i. Tutup XAMPP Control Panel dengan aman.

PROSEDUR KEDATANGAN

Kebutuhan saat di LabKom

Untuk memastikan kelancaran praktikum di Laboratorium Komputer (LabKom), mahasiswa perlu mempersiapkan beberapa hal berikut:

1. Peralatan Pribadi: Membawa alat tulis, buku catatan, dan flash drive jika diperlukan.
2. Perangkat Lunak: Memastikan software yang dibutuhkan seperti Visual Studio, Xamarin, Xampp, TASM, dan MASM telah tersedia di komputer laboratorium atau laptop pribadi.
3. Materi Praktikum: Membaca modul praktikum sebelum sesi dimulai untuk memahami konsep yang akan dipraktikkan.
4. Akun dan Kredensial: Menyiapkan akun login jika laboratorium menggunakan sistem autentikasi.
5. Disiplin dan Etika: Menggunakan komputer dengan baik, tidak menginstal software tanpa izin, serta menjaga kebersihan dan ketertiban di lingkungan laboratorium.

Prosedur kedatangan di LabKom

- Datang tepat waktu sesuai jadwal yang telah ditentukan.
- Mengisi daftar hadir atau melakukan absensi melalui sistem yang disediakan.
- Menempatkan barang pribadi dengan rapi agar tidak mengganggu jalannya praktikum.
- Menyalakan komputer dan login menggunakan akun yang telah disediakan.
- Membuka software yang akan digunakan dalam sesi praktikum.
- Mengikuti instruksi dari dosen atau asisten laboratorium.
- Mengerjakan tugas praktikum sesuai modul yang diberikan.
- Melakukan diskusi jika mengalami kesulitan, namun tetap menjaga suasana kondusif.
- Menyimpan hasil kerja di penyimpanan yang sesuai (flash drive, cloud, atau sistem lab).
- Menutup semua aplikasi dan logout dari akun laboratorium jika diperlukan.
- Mematikan komputer dan merapikan kembali meja kerja.
- Mengisi jurnal praktikum atau laporan jika diperlukan sebelum meninggalkan laboratorium.

PROSEDUR PRAKTIKUM

P1 ALGORITMA PEMROGRAMAN MASM DAN TASM

A. Tahap Persiapan

1. Pastikan PC dapat dinyalakan dan tidak ada tanda kerusakan
2. Pastikan PC sudah terinstall program MASM, TASM, dan GUI Turbo Assembler
3. Pastikan ketiga program tersebut dapat dibuka dan dijalankan

B. Tahap Praktikum

1. Buka program MASM dan TASM
2. Masukkan kode di bawah ini ke dalam kedua program tersebut

```
MODEL small

.STACK 100h

.DATA
HelloMessage DB 'Hello, world',13,10,'$'

.CODE
.startup
mov ax,@data
mov ds,ax                ;set DS to point to the data
segment
mov ah,9                 ;DOS print string function
mov dx,OFFSET HelloMessage ;point to "Hello, world"
int 21h                  ;display "Hello, world"
mov ah,4ch               ;DOS terminate program function
int 21h                  ;terminate the program
END
```

3. Simpan kemudian compile program di atas
4. Amati hasil compile program yang telah dijalankan

C. Tahap Pasca Praktikum

1. Mempelajari hasil compile program
2. Ambil screenshot dan lakukan analisa sebagai berikut :
 - Perbedaan TASM dan MASM berdasarkan percobaan yang telah dilakukan
 - Program berhasil berjalan dengan baik atau terdapat kendala, beserta alasannya
3. Tutup semua jendela software TASM, MASM, dan GUI Turbo Assembler
4. Hapus file program yang telah disimpan sebelumnya

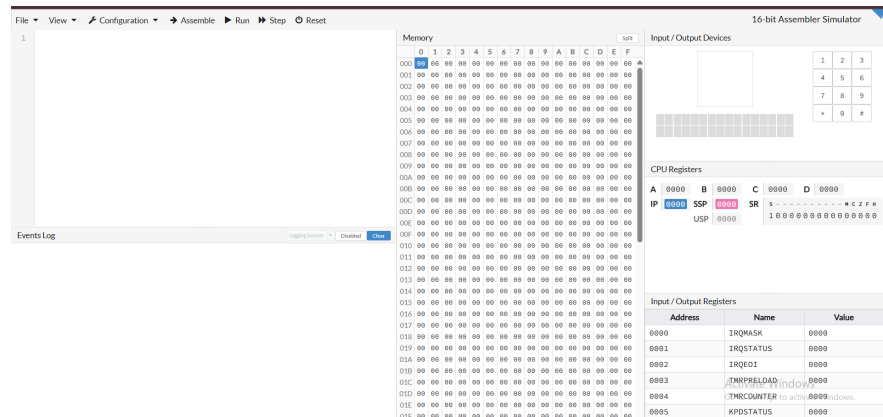
P2 ALGORITMA PEMROGRAMAN ASSEMBLER SIMULATOR

A. Tahap Persiapan

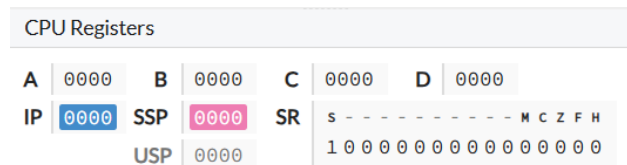
1. Pastikan PC dapat dinyalakan dan tidak ada tanda kerusakan
2. Periksa koneksi internet untuk kelancaran praktikum
3. Pastikan dapat membuka browser dan membuka link <https://parraman.github.io/asm-simulator/>

B. Tahap Praktikum

1. Buka link yang telah tersedia hingga muncul tampilan seperti di bawah ini



2. Ketik `Mov A, B` pada kolom yang kosong
3. Klik “Assemble”, kemudian “step”
4. Untuk mengembalikan ke posisi semula, klik “reset”
5. Amati yang terjadi pada kolom CPU Register dan memory di setiap penambahan code.



6. Masukkan kode di bawah ini secara berurutan

```
MOV [B+2], A
MOV B, [C+2]
MOV [B+2], A
MUL [A+100]
MUL SP
MUL D
SUB C, [A+100]
SUB D, [0x100]
AND A, B
ADD C, [A+100]
```

7. Running program sesuai dengan urutan yang telah disebutkan di poin 3 dan 4

8. Ambil screenshot pada setiap perubahan program yang telah dijalankan

C. Tahap Pasca Praktikum

1. Analisis apa yang terjadi pada setiap penambahan kode dengan melihat perubahan pada kolom CPU Registers dan memory disertai dengan bukti screenshot nya! Contoh: pertama kalian masukan kode `MOV [B+2], A` , lalu amati yang terjadi pada CPU register dan memory serta bukti screenshot, setelah itu diikuti dengan kode kedua yaitu `MOV B, [C+2]`, amati Kembali dan seterusnya hingga kode terakhir
2. Mempelajari setiap code beserta fungsinya dan buat tabel penjelasan instruksi dan fungsinya

| Instruksi | Penjelasan |
|-----------|------------|
| | |
| | |

P3 ALGORITMA PEMROGRAMAN OBJECT ORIENTED PROGRAM

A. Tahap Persiapan

1. Laptop
2. Akses Internet
3. C++ compiler, Website: https://www.onlinegdb.com/online_c++_compiler

B. Tahap Praktikum

1. Dibawah ini, terdapat 3 source code, diharapkan setiap anggota kelompok memilih source code yang berbeda
2. Buka website https://www.onlinegdb.com/online_c++_compiler
3. Bersihkan semua kolom
4. Masukkan source code seperti di bawah ini:

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

class Barang{
public:
    string kodeBarang;
    string namaBarang;
    float hargaBarang;

    void simpanBarang(string x, string y, float z){
        kodeBarang = x;
        namaBarang = y;
        hargaBarang = z;

        ofstream myWriteFile("filename.txt", ios::app);
        myWriteFile << "\n" << kodeBarang << "\t" << namaBarang << "\t" << hargaBarang;
        myWriteFile.close();
    }

    void tampil(){
        string myText;
        ifstream myReadFile("filename.txt");
        while (getline(myReadFile, myText)){
            cout << myText << endl;
        }
        myReadFile.close();
    }
};
```

```
int main()
{
    string kode, nama;
    float harga;

    cout << "INPUT DATA BARANG \n";
    cout << "Kode Barang: ";
    cin >> kode;
    cout << "Nama Barang: ";
    cin >> nama;
    cout << "Harga: ";
    cin >> harga;

    Barang dataBarang;

    dataBarang.simpanBarang(kode, nama, harga);
    dataBarang.tampil();

    return 0;
}
```

5. Run source code
6. Masukkan kode barang, nama barang, dan harga sebanyak 3 kali
7. Selanjutnya buka tab baru dan buka website https://www.onlinegdb.com/online_c++_compiler
8. Bersihkan semua kolom
9. Masukkan source code seperti dibawah ini:

```
#include <iostream>

using namespace std;

class Segitiga
{
public:
    float a;
    float t;
    float luas;

    Segitiga(float alas, float tinggi)
    {
        a=alas;
        t=tinggi;
    }

    void HitungLuas()
    {
        luas=a*t/2;
        cout<<"Luas Segitiga = "<<luas<<endl;
    }
};

class Persegi
{
public:
    float s;
    float luas;

    Persegi(float sisi)
    {
        s=sisi;
    }

    void HitungLuas()
    {
        luas=s*s;
        cout<<"Luas Persegi = "<<luas<<endl;
    }
};
```

```
int main()
{
    float alas;
    float tinggi;

    cout << "INPUT VARIABLE SEGITIGA: \n";
    cout << "Alas: ";
    cin >> alas;
    cout << "Tinggi: ";
    cin >> tinggi;

    Segitiga b (alas,tinggi);
    //b.setA(2);
    //b.setT(3);
    b.HitungLuas();

    float sisi;

    cout << "\n INPUT VARIABLE PERSEGI: \n";
    cout << "Sisi: ";
    cin >> sisi;

    Persegi c (sisi);
    //b.setA(2);
    //b.setT(3);
    c.HitungLuas();
}
```

10. Selanjutnya buka tab baru dan buka website https://www.onlinegdb.com/online_c++_compiler
11. Bersihkan semua kolom
12. Masukkan source code seperti dibawah ini:

```
#include <iostream>
#include <cmath>

using namespace std;

class Bola {
private:
    double radius;
public:
    Bola(double r) : radius(r) {}

    double hitungVolume() {
        return (4.0 / 3.0) * M_PI * pow(radius, 3);
    }

    double hitungLuasPermukaan() {
        return 4 * M_PI * pow(radius, 2);
    }
};
```

```
class Kerucut {
private:
    double radius;
    double tinggi;
public:
    Kerucut(double r, double t) : radius(r), tinggi(t) {}

    double hitungVolume() {
        return (1.0 / 3.0) * M_PI * pow(radius, 2) * tinggi;
    }

    double hitungLuasPermukaan() {
        double s = sqrt(pow(radius, 2) + pow(tinggi, 2));
        return M_PI * radius * (radius + s);
    }
};
```

```

int main() {
    double r, t;

    cout << "Masukkan jari-jari bola: ";
    cin >> r;
    Bola bola(r);
    cout << "Volume Bola: " << bola.hitungVolume() << endl;
    cout << "Luas Permukaan Bola: " << bola.hitungLuasPermukaan() << endl;

    cout << "Masukkan jari-jari alas kerucut: ";
    cin >> r;
    cout << "Masukkan tinggi kerucut: ";
    cin >> t;
    Kerucut kerucut(r, t);
    cout << "Volume Kerucut: " << kerucut.hitungVolume() << endl;
    cout << "Luas Permukaan Kerucut: " << kerucut.hitungLuasPermukaan() << endl;

    return 0;
}

```

13. Run source code
14. Masukkan nilai yang diperlukan

Tahap Pasca Praktikum

1. Screenshoot hasil praktikum
2. Screenshot hasil percobaan praktikum
3. Analisa syntax yang digunakan

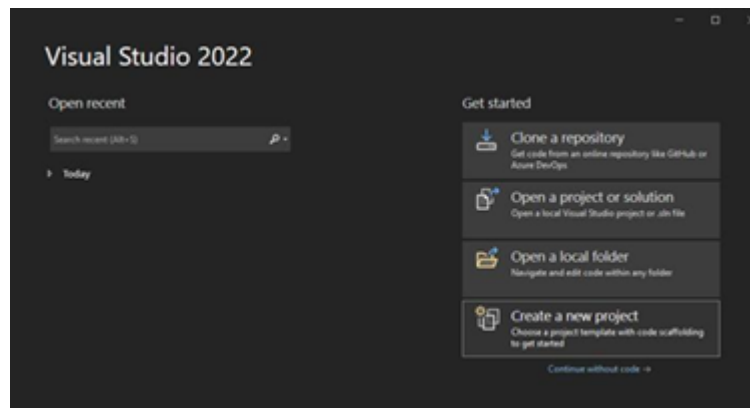
P4 ALGORITMA PEMROGRAMAN XAMARIN

A. Tahap Persiapan

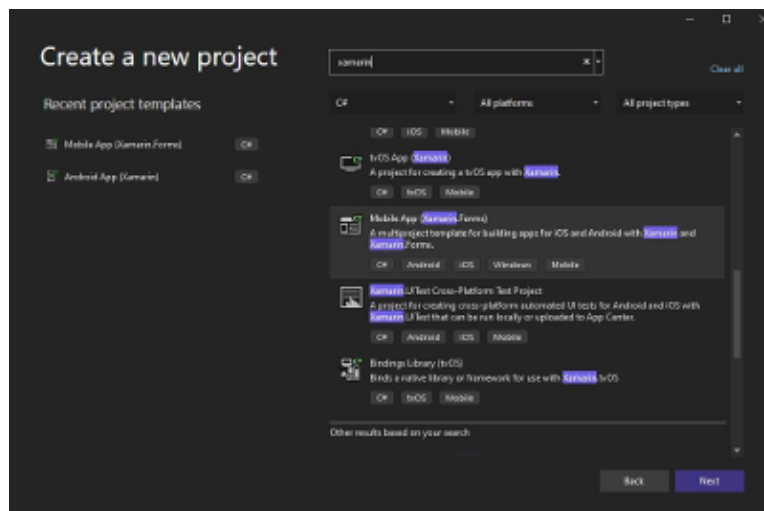
1. Laptop
2. Akses Internet
3. Visual Studio

B. Tahap Praktikum

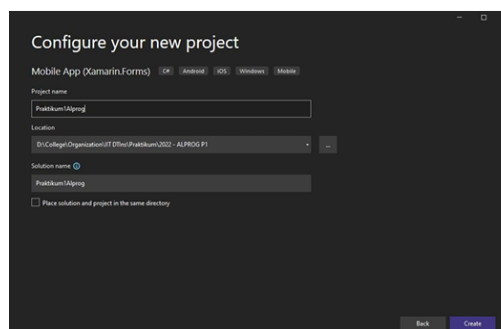
1. Membaca dengan seksama modul praktikum agar memahami maksud dan tujuan serta teori dasar praktikum.
2. Menyiapkan peralatan praktikum,
3. Membuka aplikasi visual studio, lalu klik “Create New Project”



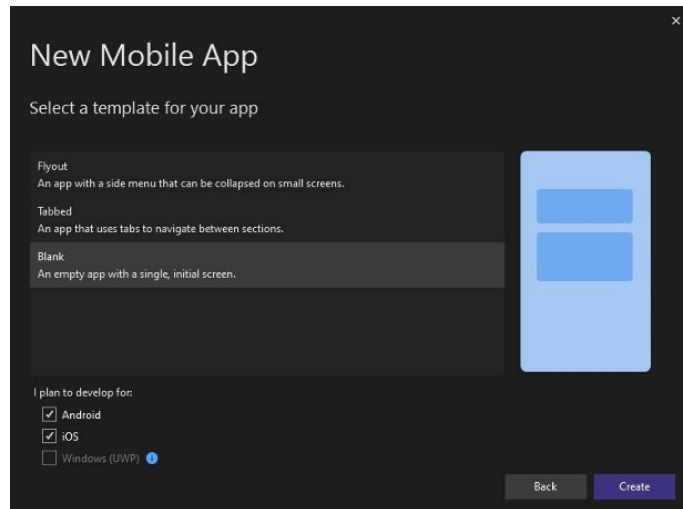
4. Pilih “Mobile App (Xamarin.Forms)” > klik next



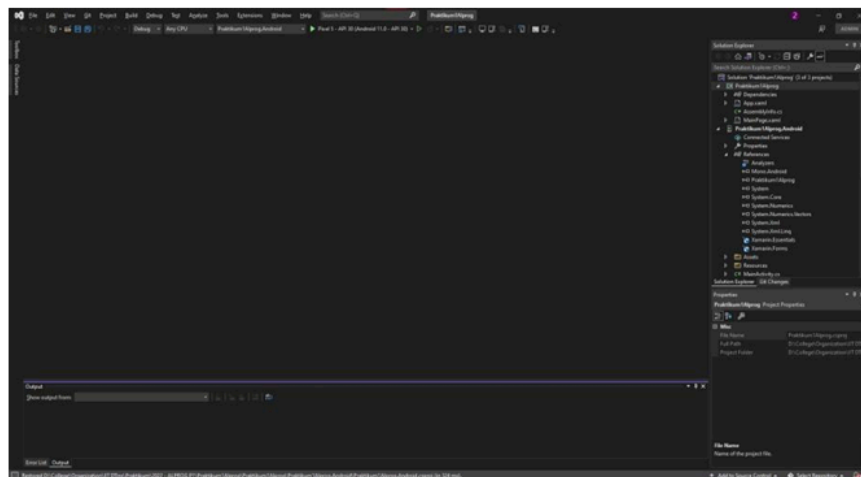
5. Beri penamaan project, serta lokasi penyimpanan file project > klik create.



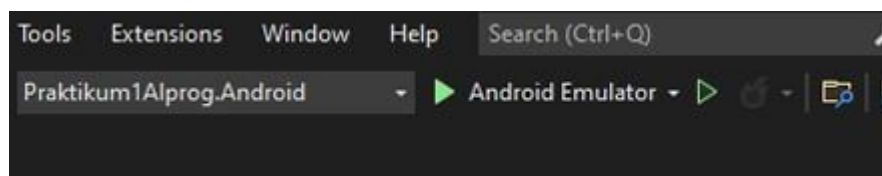
6. Pilih “Blank” > klik create. Selanjutnya tunggu hingga loading pembuatan project selesai.



7. Jika project berhasil dibuat dan tidak ada error (Seperti gambar dibawah ini), maka telah siap digunakan untuk percobaan praktikum.

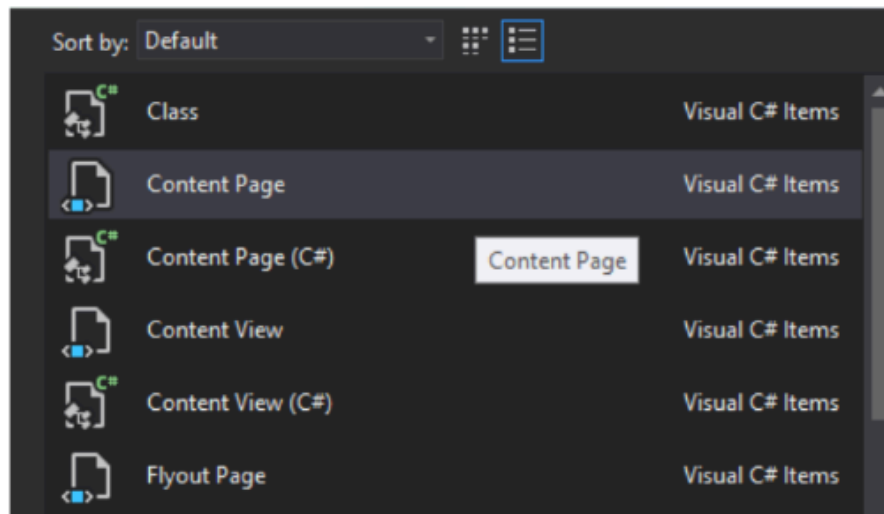
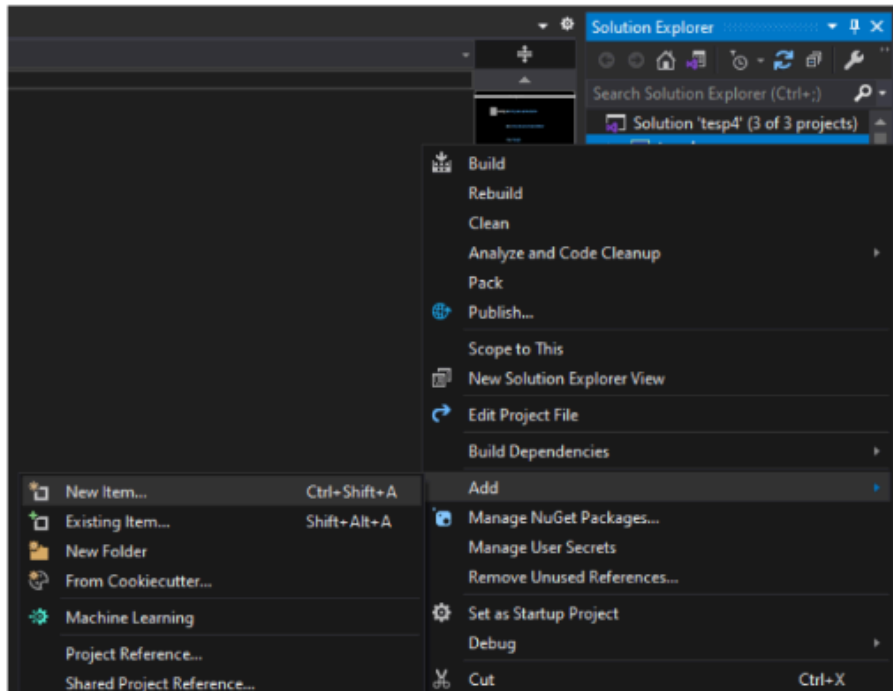


8. Untuk melakukan running program, klik ikon “play” (Seperti gambar dibawah ini) yang terdapat pada toolbar bagian atas. Terdapat 2 pilihan untuk dapat menjalankan program yaitu menggunakan emulator/smartphone.



9. Jika ingin menggunakan smartphone. Buka “Pengaturan” pada smartphone > pilih menu “Tentang Ponsel” > klik “Build Number/Software Version” selama 7 kali hingga muncul notifikasi “Developer mode telah terbuka”

10. Buat halaman baru dengan cara klik kanan nama project lalu Add > New Item > Content Page



11. Beri nama CashierPage
12. Buka App.xaml.cs, kemudian ganti kodingan menjadi

```
2 references
public App()
{
    InitializeComponent();

    MainPage = new NavigationPage(new MainPage());
}
```

13. Buka MainPage.xaml. Buat frame dengan ketentuan berikut ini

```
<Frame BackgroundColor="#2196F3" Padding="24" CornerRadius="0">
  <StackLayout Orientation="Vertical">
    <Label Text="Kasir Sederhana" HorizontalTextAlignment="Center" TextColor="White" FontSize="36"/>
    <Label Text="Nama Lengkap - 2042221xxx" HorizontalTextAlignment="Center" TextColor="White" FontSize="20"/>
  </StackLayout>
</Frame>
```

14. Buat 2 button yang pertama untuk mengarah ke halaman kasir, kedua untuk halaman Riwayat. Pada button kasir beri attribute “Clicked” yang berisi method yang berfungsi pindah halaman

```
<Button Text="Kasir" Clicked="OnButtonCashierClicked" BackgroundColor="#7321F3" CornerRadius="20"
  TextColor="White" Padding="20" Margin="20, 10"/>

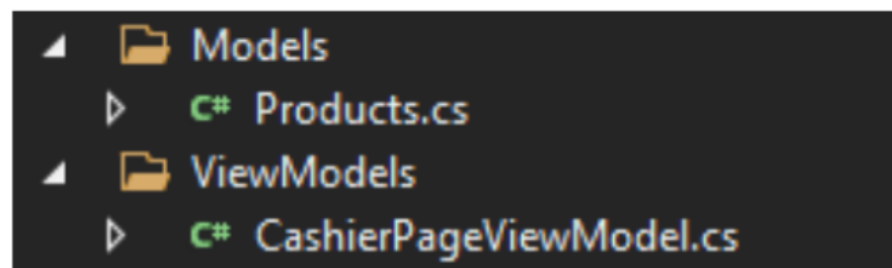
<Button Text="Riwayat" BackgroundColor="#A477E9" CornerRadius="20" TextColor="White"
  Padding="20" Margin="20, 10"/>
```

15. Buka MainPage.xaml.cs, tambahkan method OnButtonCashierClicked yang sesuai dengan atribute “Clicked” di xaml

```
5 references
public partial class MainPage : ContentPage
{
    1 reference
    public MainPage()
    {
        InitializeComponent();
    }

    0 references
    async void OnButtonCashierClicked(object sender, EventArgs args)
    {
        await Navigation.PushAsync(new CashierPage());
    }
}
```

16. Untuk running aplikasi, buat 2 folder dengan masing-masing kelas seperti gambar di bawah ini



17. Buka Models > Products.cs dahulu, ketik kode seperti di bawah

```
namespace tesp4.Models
{
    5 references
    class Products
    {
        1 reference
        public int ProductId { get; set; }

        2 references
        public string ProductName { get; set; }

        2 references
        public int ProductPrice { get; set; }
    }
}
```

18. Buka ViewModels > CashierPageViewModel.cs, ketik kode di bawah ini

```
2 references
public List<Products> ProductsList { get; set; }

1 reference
public CashierPageViewModel()
{
    try
    {
        ProductsList = new List<Products>();

        ProductsList.Add(new Products
        {
            ProductId = 1,
            ProductName = "Ikan Segar",
            ProductPrice = 1000
        });
    } catch (Exception ex)
    {
    }
}
```

19. BERIKAN VARIAN DATA LAIN, BUAT HINGGA 3 DATA

```
Products selectedRecord;

1 reference
public Products SelectedRecord
{
    get { return selectedRecord; }
    set
    {
        if (selectedRecord != value)
        {
            selectedRecord = value;
        }
    }
}
```

20. Buka CashierPage.xaml.cs, ketik kode seperti di bawah

```
5 references
public partial class CashierPage : ContentPage
{
    CashierPageViewModel ProductModel = null;
    1 reference
    public CashierPage()
    {
        InitializeComponent();
        ProductModel = new CashierPageViewModel();
        BindingContext = ProductModel;
    }
}
```

21. Buka CashierPage.xaml, ketik code seperti di bawah. Pastikan pemberian atribut x:name pada entry benar

```
<StackLayout>
    <StackLayout Padding="30, 10">
        <Label Text="Nama Pembeli"/>
        <Entry x:Name="CustNameInput" Placeholder="Nama Pembeli..."/>
    </StackLayout>

    <StackLayout Padding="30, 10">
        <Label Text="Produk"/>
        <Picker ItemDisplayBinding="{Binding ProductName}" x:Name="ProductInput" Title="Pilih Produk"
            ItemsSource="{Binding ProductsList}" SelectedItem="{Binding SelectedRecord}"/>
    </StackLayout>

    <StackLayout Padding="30, 10">
        <Label Text="Jumlah Pembelian"/>
        <Entry x:Name="QuantityInput" Placeholder="Jumlah Pembelian..." Keyboard="Numeric"/>
    </StackLayout>

    <StackLayout Padding="30, 10">
        <Label Text="Jumlah Bayar"/>
        <Entry x:Name="PaymentInput" Placeholder="Jumlah Bayar..." Keyboard="Numeric"/>
    </StackLayout>

    <Button Text="Submit" Clicked="OnButtonClicked" Margin="30, 10"/>
</StackLayout>
```

22. Buka CashierPage.xaml.cs kembali, ketik kode seperti di bawah

```
0 references
async void OnButtonClicked(object sender, EventArgs args)
{
    }
}
```

23. Running Aplikasi

```
var CustNameEntry = CustNameInput.Text;
var ProductEntry = ProductModel.SelectedRecord;
var QuantityEntry = int.Parse(QuantityInput.Text);
var PaymentEntry = int.Parse(PaymentInput.Text);

int TotalPayment = ProductEntry.ProductPrice * QuantityEntry;

if (PaymentEntry >= TotalPayment)
{
    int Change = PaymentEntry - TotalPayment;

    await DisplayAlert("Berhasil!",
        $"Nama Pembeli : {CustNameEntry}\n" +
        $"Produk : {ProductEntry.ProductName}\n" +
        $"Jumlah Beli : {QuantityEntry}\n" +
        $"Total Bayar : Rp{TotalPayment}\n" +
        $"Bayar : Rp{PaymentEntry}\n" +
        $"Kembali : Rp{Change}",
        "OK");
}
else
{
    await DisplayAlert("Gagal!",
        "Pembayaran Gagal",
        "OK");
}
```

Kesuluruhan source code seperti ini

```
0 references
async void OnButtonClicked(object sender, EventArgs args)
{
    var CustNameEntry = CustNameInput.Text;
    var ProductEntry = ProductModel.SelectedRecord;
    var QuantityEntry = int.Parse(QuantityInput.Text);
    var PaymentEntry = int.Parse(PaymentInput.Text);

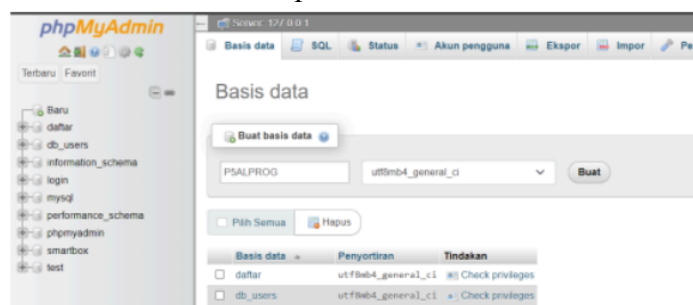
    int TotalPayment = ProductEntry.ProductPrice * QuantityEntry;

    if (PaymentEntry >= TotalPayment)
    {
        int Change = PaymentEntry - TotalPayment;

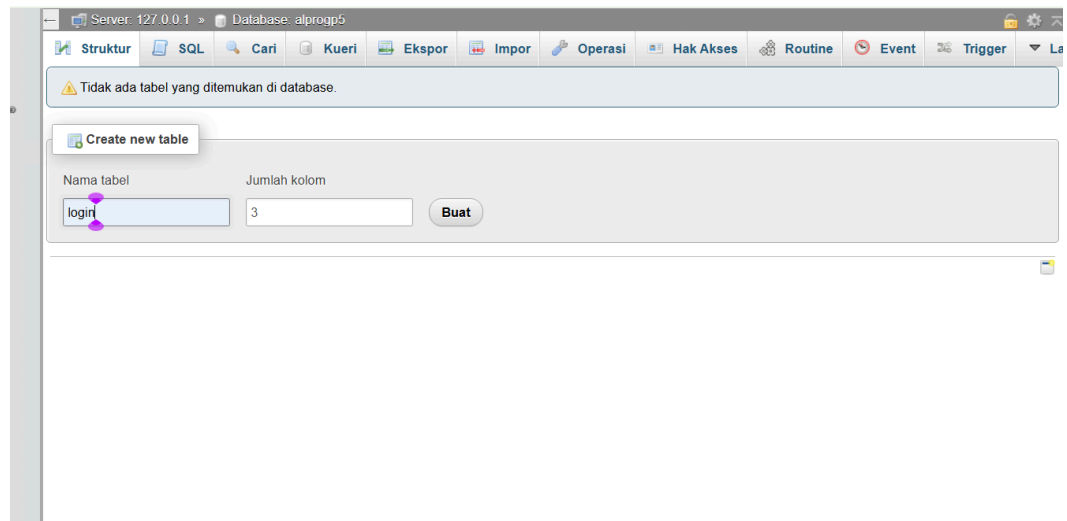
        await DisplayAlert("Berhasil!",
            $"Nama Pembeli : {CustNameEntry}\n" +
            $"Produk : {ProductEntry.ProductName}\n" +
            $"Jumlah Beli : {QuantityEntry}\n" +
            $"Total Bayar : Rp{TotalPayment}\n" +
            $"Bayar : Rp{PaymentEntry}\n" +
            $"Kembali : Rp{Change}",
            "OK");
    }
    else
    {
        await DisplayAlert("Gagal!",
            "Pembayaran Gagal",
            "OK");
    }
}
```

Tahap Pasca Pratikum

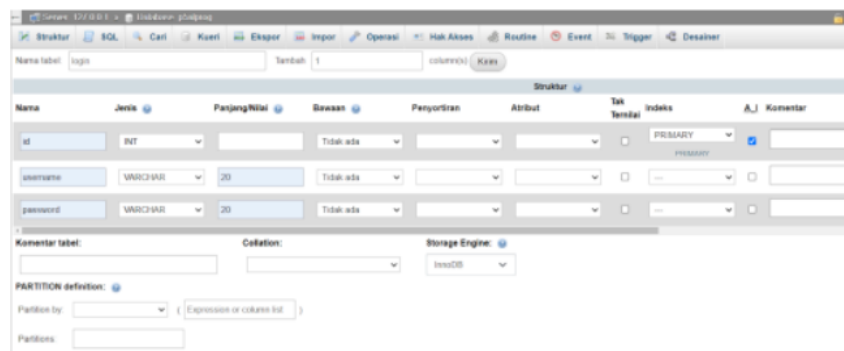
1. Praktikan memvariasikan fitur yang ada di xamarin seperti diskon, pajak, dan cancel barang.
2. Membuat video dari hasil tugas pasca pratikum yang diupload di youtube dan linknya dilampirkan pada lampiran laporan.



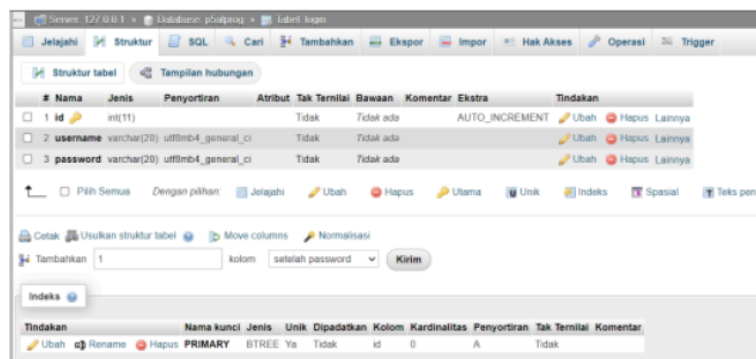
4. Buat Nama Tabel dan Jumlah Kolom



5. Selanjutnya lengkapi tabel seperti gambar di bawah pada ID jangan lupa centang AUTO INCREMENT lalu simpan

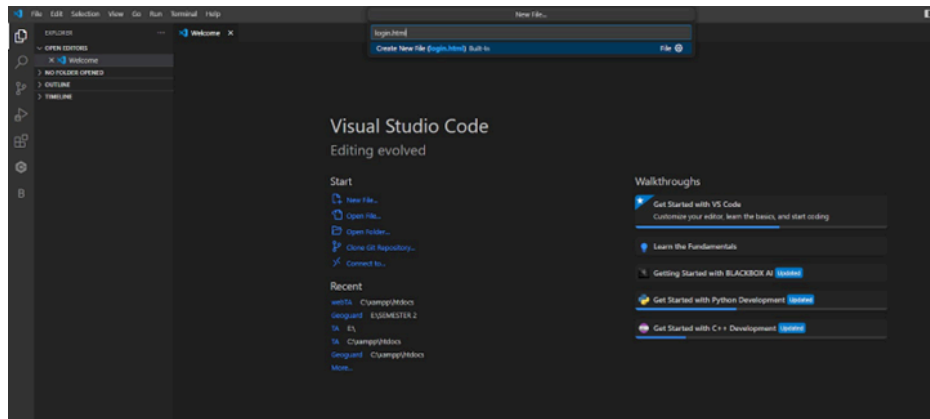


6. Tampilan database yang sudah jadi



Membuat Form Login di Visual Studio Code

1. Buka Aplikasi Visual Studio Code
2. Pada menu start pilih New File dan beri nama (login.html) klik Enter setelah itu buat folder untuk menyimpan file (login.html) dan klik Create File



3. Buat file HTML untuk form login.

```
C: > xampp > htdocs > p5alprog > login.html > html > body > div.container > h2
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Login Page</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <div class="container">
11    <h2>Login</h2>
12    <form action="dashboard.html">
13      <input type="text" placeholder="Username" required>
14      <input type="password" placeholder="Password" required>
15      <input type="submit" value="Login">
16    </form>
17    <div class="register-link">
18      <p>Don't have an account? <a href="register.html">Register</a></p>
19    </div>
20  </div>
21 </body>
22 </html>
23
```

4. Selanjutnya, buat file HTML untuk form register

```
> register.html > ...
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Registration Page</title>
7   <link rel="stylesheet" href="styles.css">
8 </head>
9 <body>
10  <div class="container">
11    <h2>Register</h2>
12    <form action="register.php" method="POST">
13      <input type="text" name="username" placeholder="Username" required>
14      <input type="password" name="password" placeholder="Password" required>
15      <input type="password" name="confirm_password" placeholder="Confirm Password" required>
16      <input type="submit" value="Register">
17    </form>
18    <div class="login-link">
19      <p>Already have an account? <a href="login.html">Login</a></p>
20    </div>
21  </div>
22 </body>
23 </html>
24
```

5. Buat file CSS untuk styles dari form login dan register

```
# styles.css > body
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #f4f4f4;
4   margin: 0;
5   padding: 0;
6   display: flex;
7   justify-content: center;
8   align-items: center;
9   height: 100vh;
10 }
11
12 .container {
13   background-color: #fff;
14   padding: 20px;
15   border-radius: 5px;
16   box-shadow: 0 2px 5px rgba(0, 0, 0, 0.1);
17   width: 300px;
18   text-align: center;
19 }
20
21 input[type="text"], input[type="password"] {
22   width: 100%;
23   padding: 10px;
24   margin: 10px 0;
25   border: 1px solid #ccc;
26   border-radius: 5px;
27   box-sizing: border-box;
28 }
29
30 input[type="submit"] {
31   width: 100%;
32   background-color: #4caf50;
33   color: white;
34   padding: 10px;
35   border: none;
36   border-radius: 5px;
37   cursor: pointer;
38   font-size: 16px;
39 }
40
41 input[type="submit"]:hover {
42   background-color: #45a049;
43 }
44
45 .register-link {
46   margin-top: 10px;
47   font-size: 14px;
48 }
```

6. Buat file PHP untuk form Login. (pastikan untuk Koneksi ke database dan Query untuk mencari user dengan username dan password yang sesuai yang sudah kita buat di dataBase XAMPP tadi).

```
login.php
1 <?php
2 session_start();
3
4 if(isset($_POST['username'], $_POST['password'])) {
5   $username = $_POST['username'];
6   $password = $_POST['password'];
7
8   // Koneksi ke database
9   $conn = new mysqli('localhost', 'root', '', 'p5alprog');
10
11   // Periksa koneksi
12   if($conn->connect_error) {
13     die("Koneksi gagal: " . $conn->connect_error);
14   }
15
16   // Query untuk mencari user dengan username dan password yang sesuai
17   $query = "SELECT * FROM login WHERE username='$username' AND password='$password'";
18   $result = $conn->query($query);
19
20   // Periksa apakah hasil query menghasilkan satu baris (user yang valid)
21   if($result->num_rows == 1) {
22     // Login berhasil
23     $_SESSION['username'] = $username;
24     header("location: dashboard.html"); // Redirect ke halaman dashboard atau halaman lainnya
25   } else {
26     // Login gagal
27     echo "Login gagal. Periksa kembali username dan password Anda.";
28   }
29
30   // Tutup koneksi database
31   $conn->close();
32 }
33 >>
```

7. Buat file PHP untuk form register. (pastikan untuk Koneksi ke database dan Query untuk mencari user dengan username dan password yang sesuai yang sudah kita buat di dataBase XAMPP tadi).

```

register.php
1  <?php
2  if(isset($_POST['username'], $_POST['password'])) {
3      $username = $_POST['username'];
4      $password = $_POST['password'];
5
6      // Koneksi ke database
7      $conn = new mysqli('localhost', 'root', '', 'p5alprog');
8
9      // Periksa koneksi
10     if($conn->connect_error) {
11         die("Koneksi gagal: " . $conn->connect_error);
12     }
13
14     // Query untuk menambahkan user baru ke dalam tabel
15     $query = "INSERT INTO login (username, password) VALUES ('$username', '$password')";
16     if($conn->query($query) === TRUE) {
17         echo "Registrasi berhasil.";
18     } else {
19         echo "Error: " . $query . "<br>" . $conn->error;
20     }
21
22     // Tutup koneksi database
23     $conn->close();
24 }
25 ?>

```

8. Buat file PHP untuk form dashboard.html

```

<> dashboard.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Dashboard</title>
7      <link rel="stylesheet" href="styles.css">
8  </head>
9  <body>
10     <div class="container">
11         <h2>Welcome to Your Dashboard</h2>
12         <p>Selamat datang</p>
13     </div>
14 </body>
15 </html>
16

```

9. Tampilan Form login yang sudah kita buat

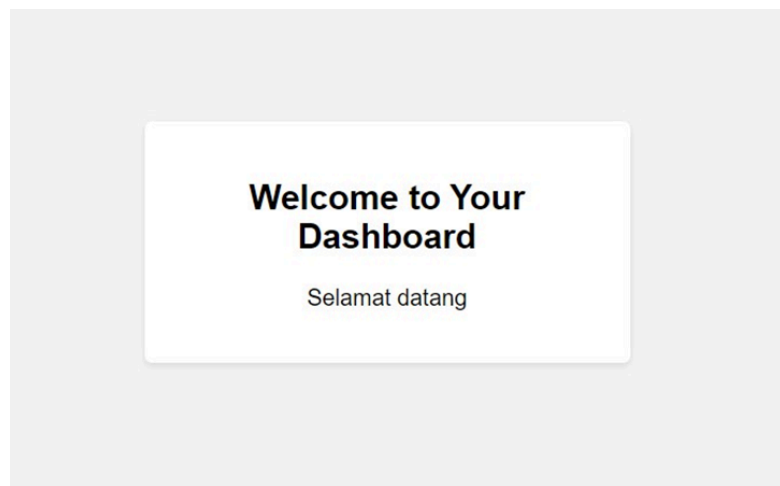
Login

Don't have an account? [Register](#)

Register

Already have an account? [Login](#)



10. Setelah itu salin folder file yang sudah kita bikin tadi
11. Buka Buka **folder XAMPP** di drive C komputer kamu:
`C:\xampp`
12. Masuk ke **folder htdocs**:
`C:\xampp\htdocs`
13. Di folder `htdocs` itu, **salin folder proyek kamu** (yang berisi file `login.html`, `register.html`, `styles.css`, `login.php`, `register.php`, `dashboard.html`).
14. Setelah itu buka laman `http://localhost/namafolderkalian/login.html`
15. Setelah berhasil terkoneksi coba melakukan register dan jika berhasil regis akan muncul tulisan “Registrasi berhasil.”
16. Setelah regis berhasil, coba melakukan login dan jika berhasil akan menuju menu `dashboard.html`



Tahap Pasca Pratikum

1. Berdasarkan percobaan yang telah kalian buat, analisa dan jelaskan proses bagaimana cara koneksi dari XAMPP ke Website yang sudah dibuat
2. Variasikan program yang sudah kalian buat dengan memasukkan form identitas diri kalian pada dashboard website yang terkoneksi dengan XAMPP dan tampilkan hasil dari form identitas diri pada website yang kalian buat.

Lampiran Permit to Work

| | | | | | |
|---|--|--|--|---|--|
|  | | <p align="center">PERMIT TO WORK SAFETY INSTRUMENTED SYSTEM LABORATORY</p> | |  | |
| Doc: P2 | | Rev: 00 | | 12/12/2025 | |

| | | | | | |
|-------------------------|-----------------------|------------------|--------------------|--|--|
| Pemohon | Nama NRP | | | | |
| Lokasi | Laboratorium | | | | |
| Deskripsi Pekerjaan | Judul Praktikum | | | | |
| Masa Berlaku Izin Kerja | Tanggal: hh – bb - tt | Mulai: 00.00 WIB | Selesai: 00.00 WIB | | |
| Alat dan Bahan | 1. 2. 3. | 4. 5. 6. | 7. 8. 9. | | |

| | | | |
|--|---|--|-----------|
| Checklist terkait faktor perizinan dan keselamatan kerja | | YES | NO |
| Apakah pekerjaan ini telah disetujui oleh Dosen Pengampu Mata Kuliah? | | | |
| Apakah izin pekerjaan ini telah disetujui oleh Kepala Laboratorium? | | | |
| Apakah sudah dibuat JSA (<i>Job Safety Analysis</i>)? | | | |
| Apakah APD yang sesuai telah dipersiapkan? | | | |
| <ul style="list-style-type: none"> Safety Helmet Safety Glasses Safety Gloves | <ul style="list-style-type: none"> Safety Shoes Mask Earmuff | <ul style="list-style-type: none"> Lainnya, sebutkan: | |
| Apakah area pekerjaan telah bebas dari material yang mudah terbakar? | | | |
| Apakah area pekerjaan telah bebas dari bahan yang mudah meledak? | | | |
| Apakah peralatan pemadam kebakaran telah tersedia? | | | |
| Apakah peralatan P3K telah tersedia? | | | |
| Apakah prosedur keadaan darurat telah dipahami? | | | |
| Apakah semua peralatan/perlengkapan telah diperiksa? | | | |

Saya telah memahami dan berkomitmen tentang pekerjaan yang akan saya kerjakan, dan akan melaksanakan pekerjaan sesuai prosedur dengan memperhatikan faktor keselamatan dan kesehatan kerja.

| | | |
|-----------------|----------------------|-----------------------|
| Catatan: | Diajukan oleh | Disetujui oleh |
| | | |
| | Nama Pemohon | Nama Laboran |

Note:

- Izin ini **harus disetujui** sebelum melakukan pekerjaan di laboratorium.
- Setiap pelanggaran terhadap SOP laboratorium dapat menyebabkan izin kerja dicabut.
- Jika terjadi insiden, segera laporkan kepada petugas laboratorium.

Lampiran JSA (Job Safety Analysis)

| FORM JSA PRAKTIKUM LAB SIS | | | | | |
|--------------------------------|--|---|---|--|--------------------------------------|
| LAB Safety Instrumented System | TANGGAL | NO JSA | | REV NO | DISIAPKAN |
| | 25 Februari 2025 | | | 1 - | |
| JOB SAFETY ANALYSIS | NAMA PROYEK | | | | DIREVIEW |
| | Praktikum Lab SIS | | | | |
| NAMA PEKERJAAN | LOKASI PEKERJAAN | PENGAWAS PEKERJAAN | | DISETUJUI | |
| | | | | | |
| | | | | | |
| | | | | | |
| NO | Urutan dari Langkah-langkah Pokok Pekerjaan Sequence Of Basic Job Steps | Bahaya-Bahaya Potensial Potential Hazard | Resiko/ Hazard Risk | Tindakan atau Prosedur yang Direkomendasikan Recommended Action or Procedure | Penanggung Jawab Person In Charge |
| 1 | Menyalakan Komputer | Tersengat Arus Listrik | - Serious or fatal injury due to electric shock | 1. Memperbaiki kabel yang terkelupas 2. Menggunakan casing CPU berbahan plastik 3. Menggunakan pengatur tegangan listrik 4. Menggunakan alas kaki (sandal/sepatu) 5. Tangan harus dalam kondisi kering | |
| 2 | Mengoperasikan Komputer | Kebakaran | - Fire or damage to electrical equipment | 1. Menggunakan pengatur tegangan listrik 2. Menyediakan APAR (Dry Chemical) yang mudah dijangkau | |

Untuk template PTW dan JSA dapat diunduh pada <https://intip.in/5ANc>