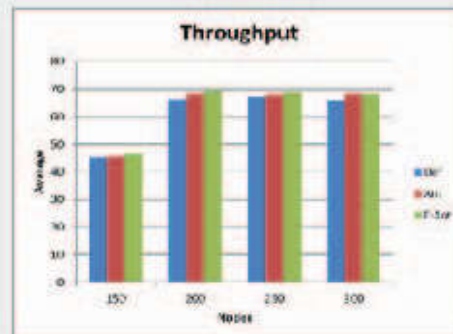
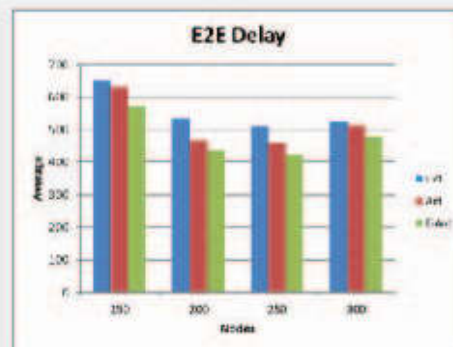
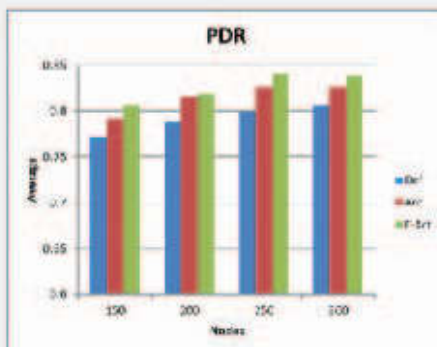


LATAR BELAKANG

Ad-Hoc On-Demand Distance Vector (AODV) merupakan sebuah protokol pemilihan rute (*routing protocol*) reaktif yang dapat digunakan dalam pemilihan rute untuk *packet forwarding* pada *Vehicular Ad-Hoc Network* (VANET). Mobilitas yang tinggi dan energi yang terbatas pada jaringan VANET menyebabkan tingkat kestabilan rute rendah dan rentan rusak / terputus. Pada *routing protocol* AODV, semua *node* melakukan *broadcast* paket *Route Request* (RREQ) ke seluruh *node* yang bisa dijangkau, sehingga menghabiskan energi dan membutuhkan waktu lama untuk menemukan rute. Untuk membatasi *broadcast* paket RREQ, telah diujikan sebelumnya metode modifikasi pada AODV dengan memanfaatkan algoritma *Ant Colony Optimization* yang mengambil nilai kekuatan sinyal dan energi residu *node* tetangga sebagai parameter fungsi heuristiknya. Pada Tugas Akhir ini akan dilakukan modifikasi pada fungsi heuristik *Ant Colony Optimization* dengan menambahkan parameter baru yakni kecepatan dan jarak *node* tetangga relatif dengan *node* tujuan.

HASIL SKENARIO REAL (JALAN DR. SOETOMO, SURABAYA)



SARAN

- Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang akurat.
- Perhitungan pula arah gerak *node* apakah mendekati atau menjauhi *node* tujuan.

METODE

Modifikasi pada Tugas Akhir ini ditujukan untuk mengoptimalkan *route discovery mechanism* pada AODV untuk meningkatkan *Quality of Service* (QoS) jaringan VANET. Ant Colony Optimization menggunakan *heuristic function pheromone counter* (PC) yang nilainya akan digunakan untuk mempertimbangkan pengambilan rute. Fungsi *pheromone counter*, pada metode sebelumnya mengambil parameter kekuatan sinyal yang diterima, energi residu pada *node* tetangga, *congestion* pada *node* tetangga dan *hop count* pada *node* tetangga.

Pada Tugas Akhir ini akan ditambahkan parameter pada fungsi ini yakni kecepatan *node* tetangga dan jarak *node* tetangga relatif dengan *destination node*. Saat simulasi dimulai, *source node* akan mempertimbangkan kekuatan sinyal *node* tetangga (*node* perantara) untuk mengirimkan RREQ. *Node* perantara bertugas untuk meneruskan RREQ ke *node* sekitarnya dengan memanfaatkan fungsi *pheromone counter* (PC) untuk memilih *node* tetangga yang akan dikirim RREQ. RREQ akan dikirimkan ke *node* tetangga yang memiliki nilai *pheromone* terbesar begitu seterusnya hingga RREQ diterima oleh *destination node*. Ketika RREQ sampai pada *destination node*, *destination node* akan mengirimkan RREP kepada *source node* melalui rute dengan nilai *pheromone* tertinggi yang merupakan rute paling stabil dan optimal.

KESIMPULAN

Modifikasi *routing protocol* AODV dengan menggunakan algoritma *Ant Colony Optimization* yang dimutakhirkan telah berhasil membatasi jumlah *forwarding node* yang bertugas untuk *rebroadcast* paket RREQ dilihat dari metrik *Forwarded Route Request* (RREQ F).

Penggunaan algoritma *Ant Colony Optimization* yang dimutakhirkan terhadap performa *routing protocol* AODV menunjukkan adanya perubahan yang cukup signifikan pada skenario real dengan rata-rata kenaikan PDR sebesar 4,38%, penurunan *End-to-End Delay* sebesar 17,79%, rata-rata penurunan *Routing Overhead* sebesar 28,77%, rata-rata penurunan *Forwarded Route Request* sebesar 28,4% dan rata-rata kenaikan *Throughput* sebesar 3,64%.



PENGEMBANGAN AODV ON-DEMAND DISTANCE VECTOR (AODV) DENGAN KONSEP POWER AND DELAY AWARE DI LINGKUNGAN DENGAN ENERGI YANG HETEROGEN PADA MOBILE AD-HOC NETWORK (MANET)

Oleh:
Ahmad Hanan
NRP 05111440007005

Dosen Pembimbing:
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc
NIP 198410162008121002

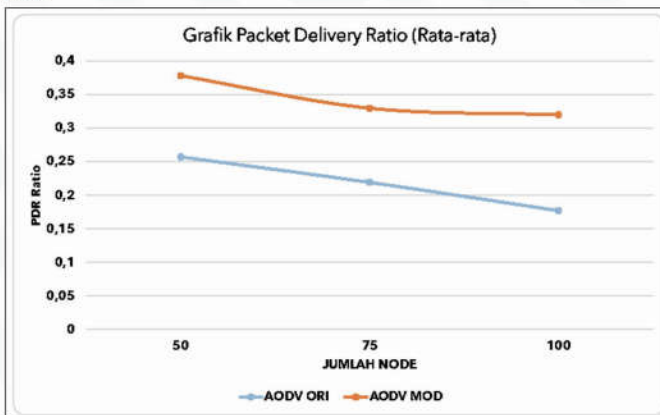
Latar Belakang

Mobile Ad-hoc Network (MANET) atau jaringan Mesh adalah sebuah jaringan yang terdiri dari banyak node bergerak yang saling berkomunikasi satu sama lain. Jaringan MANET memiliki karakteristik berupa topologi yang bersifat dinamis dan terbatasnya energi baterai yang dimiliki oleh node bergerak. Ada beberapa masalah yang ditimbulkan karena habisnya energi baterai suatu node yang bisa menyebabkan rusaknya rute pada jaringan MANET. Masalah-masalah itu di antaranya adalah hilangnya paket; node akan menginisialisasi *route discovery* atau pencarian rute ulang, sehingga menyebabkan banyaknya konsumsi *bandwidth*, meningkatkan *delay*, dan mengurangi *throughput*.

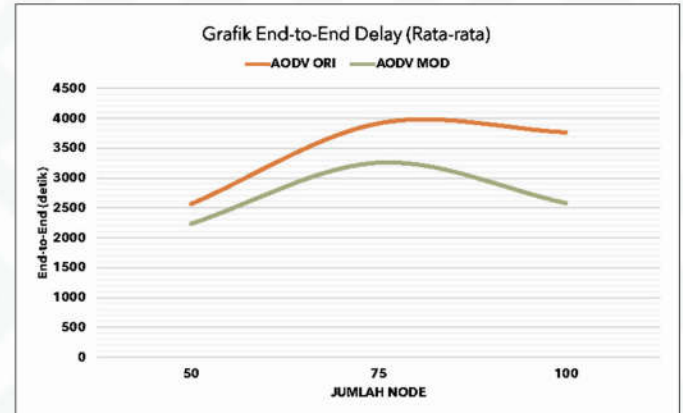
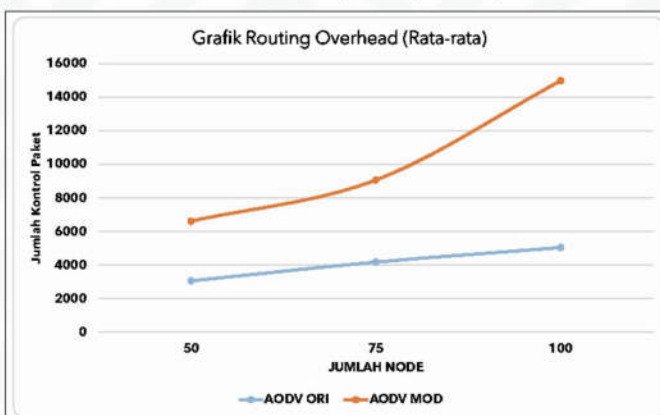
Metode

Dalam penelitian ini dilakukan studi kinerja jaringan MANET dengan menggunakan energi yang heterogen dengan menggunakan *routing protocol* baru bernama *Power and Delay-aware Multi-path Routing Protocol* (PDMRP) yang memperbaiki kekurangan dari *Stable Path Routing Protocol based on Power Awareness* (SPR), dan *Modified Ad-hoc On-Demand Distance Vector Routing Protocol* (MAODV) dalam segi *throughput*, *end-to-end delay*, dan *loss rate*.

Hasil



Grafik Packet Delivery Route (Rata-rata)



Grafik End-to-End Delay (Rata-rata)

Simpulan

Kesimpulan yang dapat diambil dari penelitian ini adalah sebagai berikut ini:
Dampak dari modifikasi ini bisa dilihat dari peningkatan PDR dan RO, yakni dengan rincian: Pada node 50, PDR mengalami peningkatan sebesar 47% dan RO mengalami peningkatan hingga 116%. Pada node 75, PDR mengalami peningkatan sebesar 50% dan RO juga mengalami peningkatan sebesar 117%. Pada node 100, PDR mengalami peningkatan sebesar 80,6% sedangkan untuk RO juga mengalami peningkatan hampir dua kali lipatnya, yakni sebesar 197%.
Namun, modifikasi ini tidak berdampak pada kenaikan pada End-to-End (E2E). Di sini, E2E mengalami penurunan nilai. Untuk node 50, E2E mengalami penurunan sebesar 12,7%. Pada node 75 dan node 100 juga mengalami penurunan, masing-masing sebesar 16,8% dan 31,4%.

Saran

Saran yang dapat diberikan dari hasil uji coba dan evaluasi adalah sebagai berikut:

1. Perhitungan jumlah node tetangga yang lebih dinamis pada routing protocol AODV. Contohnya saat node tetangga menjauh dari jangkauan, maka ada mekanisme untuk memperbarui list simpanan node tetangga sehingga perhitungan jumlah node tetangga lebih akurat.
2. Menambahkan aspek lain untuk melakukan pengembangan penelitian ini sehingga hasilnya lebih akurat.





DISTANCE VECTOR (AODV) BERDASARKAN FAKTOR THROUGHPUT DAN HOP DI LINGKUNGAN VANETS

Oleh
Faturrahman Ma'ruf
NRP 05111540000027

Dosen Pembimbing I
Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP 198410162008121002

Dosen Pembimbing II
Ary Mazharuddin S., S.Kom., M.Comp.Sc.
NIP 198410162008121002

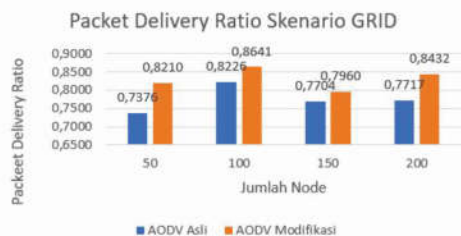
Latar Belakang

VANETs (*Vehicular Ad-Hoc Networks*) merupakan pengembangan dari MANETs (*Mobile Ad-Hoc Networks*) yang dikhususkan untuk komunikasi kendaraan jaringan nirkabel. Berdasarkan cara membangun rute, dapat dibedakan menjadi dua, yaitu *routing protocol proactive* dan *routing protocol reactive*. Contoh *routing protocol reactive* adalah *Adhoc On-Demand Distance Vector (AODV)* yang mekanismenya membuat rute ketika sumber membutuhkannya. Performa AODV menurun ketika jumlah *hop* tinggi dan *throughput* tinggi. *End-to-end delay* AODV untuk *throughput* tinggi lebih rendah dibandingkan AODV dengan *throughput* kecil. Oleh karena itu *throughput* dan *hop* harus dipertimbangkan dalam pencarian rute agar performa AODV lebih bagus.

Metode

Mengimplementasikan modifikasi AODV dengan menambahkan field *throughput* diinisialisasi nilainya sama dengan link *throughput* Source node melakukan broadcast paket RREQ dan menyimpan nilai *throughput* dari node tersebut. Kemudian menghitung nilai *throughput* setiap kali mengirimkan RREP atau RREP. Saat forward request, akan dibandingkan nilai *throughput* link dengan nilai *throughput* RREQ untuk menentukan nilai *throughput* RREQ dan menyimpan nilai *throughput* pada

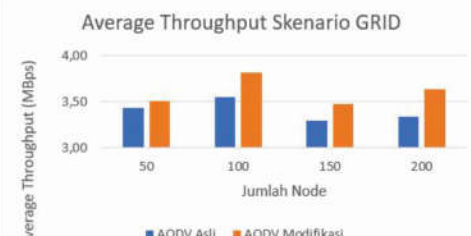
Hasil



PDR
Packet Delivery Ratio



E2E
End-to-End Delay



Average Throughput

Kesimpulan

- Pemilihan rute berdasarkan *throughput* dan *hop* bisa dilakukan dengan cara mengirim paket RREQ dengan *throughput* lebih besar jika node dengan sequence number yang sama menerima paket RREQ lebih dari 1 kali dan memilih *hop* yang lebih kecil jika nilai *throughput* sama dan mengirim paket RREP dengan *throughput* yang lebih besar jika source node menerima paket RREP lebih dari 1 kali dan memilih *hop* yang lebih kecil jika nilai *throughput* paket RREP sama dengan paket RREP sebelumnya.
- Dampak pemilihan rute berdasarkan *throughput* dan *hop* pada skenario grid adalah rata-rata peningkatan Packet Delivery Ratio(PDR) sebesar 7,23%, rata-rata penurunan End-to-End Delay sebesar 35,32%. Rata-rata peningkatan *throughput* adalah 5,93%.

Saran

- Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat, seperti lebih dari 10 skenario untuk tiap jumlah nodenya.
- Lebih banyak jumlah node yang dilakukan dalam uji coba seperti minimal 250 node untuk mengetahui nilai optimal dari AODV yang telah dimodifikasi.



DQUEUE (DQAODV) PADA LINGKUNGAN VANETS

Oleh
Muhammad Budiana Eka Faruqi
NRP 05111540000084

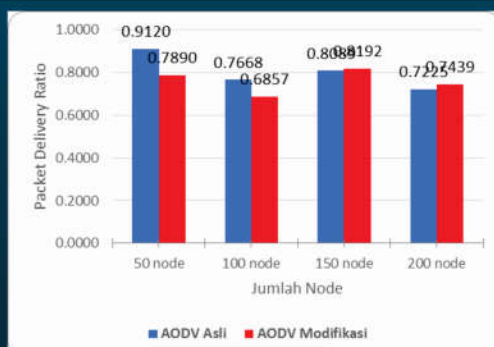
Dosen Pembimbing I
Dr. Eng. Radityo Anggoro, S.Kom., M.Sc.
NIP 198410162008121002

Dosen Pembimbing II
Wahyu Suadi, S.Kom., M.M., M.Kom.
NIP 197110302002121001

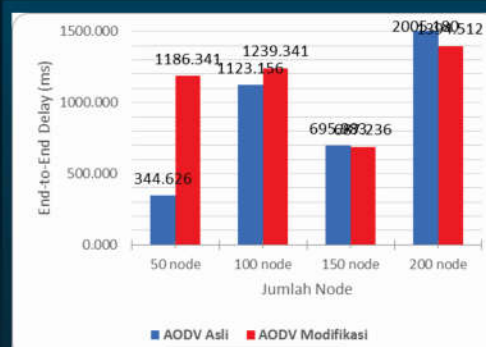
Latar Belakang

VANETs (*Vehicular Ad-Hoc Networks*) merupakan pengembangan dari MANETs (*Mobile Ad-Hoc Networks*) yang dikhususkan untuk komunikasi kendaraan jaringan nirkabel. Berdasarkan cara membangun rute, dapat dibedakan menjadi dua, yaitu *routing protocol* proaktif dan *routing protocol* reaktif. Contoh untuk *routing protocol* reaktif adalah *Adhoc On-Demand Distance Vector* (AODV) yang mekanismenya membangun rute ketika ada permintaan pengiriman paket. Pemilihan rute yang stabil saat proses pencarian rute diperlukan untuk memperpanjang waktu penggunaan. Ketika *node* menemukan kerusakan dalam rute aktif pesan *error* akan dihasilkan yang akan berpengaruh pada lamanya memilih rute stabil yang mengakibatkan *delay* dan paket yang seharusnya terkirim berkurang.

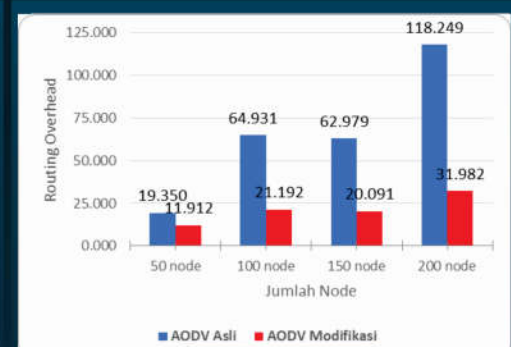
Hasil



Grafik PDR
Packet Delivery Ratio



Grafik Delay
End-to-end Delay



Grafik RO
Routing Overhead

Metode

Pada DqAODV, broadcast ID dari node lain yang mengirimkan paket RREQ akan disimpan dalam sebuah list di node penerima. Nantinya jika terjadi kerusakan, DqAODV akan menyediakan node cadangan yang sudah disimpan di node tempat kerusakan terjadi.

Tahap Perancangan Modifikasi:

- Mengubah mekanisme *route discovery* pada header RREQ.
- Setiap *node* yang menerima paket RREQ akan melakukan penyimpanan IP dari *node* pengirim.
- Menyediakan IP *node* cadangan jika terjadi kerusakan untuk melakukan *reunicast* RREP.

Kesimpulan

- Dengan menggunakan metode DqAODV, AODV modifikasi berhasil meminimalisir kegagalan saat RREP mencari rute kembali dilihat dari rata-rata Packet Delivery Ratio turun menjadi 0.68% pada skenario grid dan turun menjadi 4.96% pada skenario real namun stabil seiring dengan bertambah padatnya node.
- Dengan penggunaan DqAODV pada lingkungan VANET berhasil meningkatkan performa AODV dilihat rata-rata Routing Overhead yang turun menjadi 57.66% pada skenario grid dan turun menjadi 61.71% pada skenario real.

Saran

- Menambahkan aspek lain pada dqAODV seperti mengurangi terjadinya kerusakan link agar dqAODV dapat bekerja lebih cepat.
- Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat.

IMPLEMENTASI OTOMATISASI PENGAMBILAN WAJAH UNTUK FACE RECOGNITION DENGAN MENGGUNAKAN KEDIPAN MATA BERBASIS ANDROID

Dosen Pembimbing I :

Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.

19841016 200812 1 002

John Harison

051116400000003

Dosen Pembimbing II :

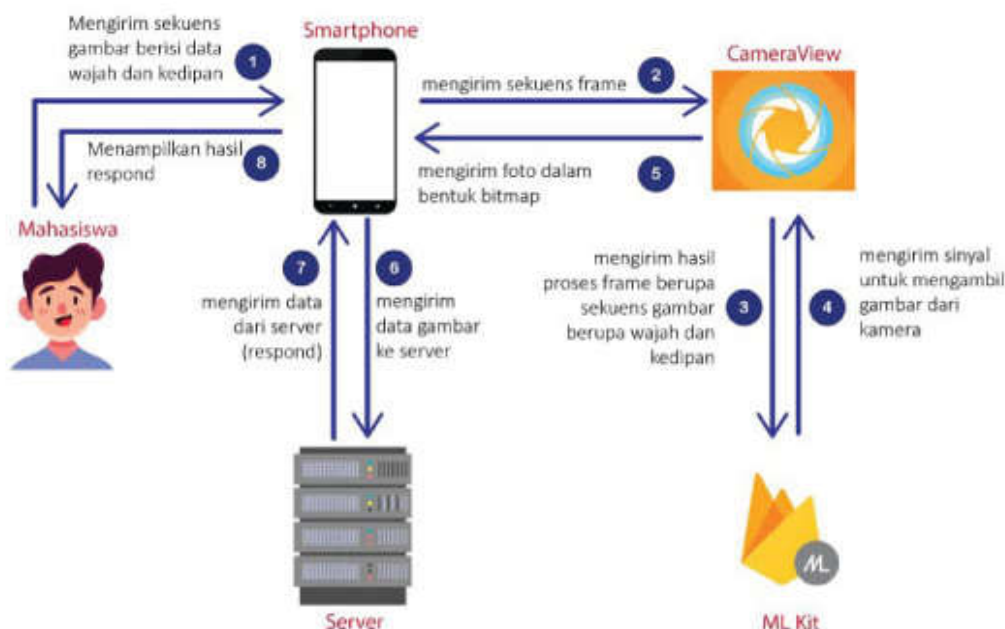
Dwi Sunaryono, S.Kom., M.Kom.

197205281997021001

LATAR BELAKANG

Sistem kehadiran mahasiswa merupakan proses pencatatan kehadiran mahasiswa di kelas. Sistem kehadiran ini dapat menjadi bukti bahwa seorang mahasiswa telah berada di dalam kelas. Di Jurusan Teknik Informatika ITS, sistem kehadiran mahasiswa dilakukan secara manual yaitu melakukan tanda tangan pada kertas daftar kehadiran kelas. Hal ini menimbulkan celah kelemahan yaitu kemungkinan terjadinya kecurangan dalam pencatatan kehadiran. Kecurangan ini sering disebut titip absen. Titip absen merupakan kejadian di mana mahasiswa tercatat hadir pada kertas daftar kehadiran kelas tetapi kenyataannya mahasiswa tersebut tidak hadir di kelas. Dengan adanya celah kelemahan ini, mahasiswa lain dapat melakukan tanda tangan untuk mahasiswa lain. Oleh karena itu, diperlukan suatu sistem yang dapat digunakan untuk mengatasi kecurangan dalam pencatatan kehadiran mahasiswa di dalam kelas.

METODE



SARAN

Saran terkait Tugas Akhir ini yang diharapkan bisa membuat Tugas Akhir ini menjadi lebih baik adalah memperbaiki User Interface untuk mempermudah pengguna dalam mencari menu.



KESIMPULAN

Berdasarkan pengerjaan Tugas Akhir ini dan hasil yang didapatkan pada tahap uji coba aplikasi, penulis mengambil beberapa kesimpulan sebagai berikut.

1. Implementasi otomatisasi pengambilan wajah dengan kedipan mata berhasil diimplementasikan

PENGEMBANGAN KEMAMPUAN SKALA OTOMATIS UNTUK INFRASTRUKTUR CLOUD-NATIVE BERBASIS KUBERNETES MENGGUNAKAN CUSTOM METRIC DAN SIDECAR PATTERN

Oleh:
Aguel Satria Wijaya
NRP 05111640000056

Dosen Pembimbing I
Royyana Muslim Ijtihad, S.Kom., M.Kom., Ph.D.ie
NIP 197708242006041001

Dosen Pembimbing II
Rizky Januar Akbar, S.Kom., M.Eng.
NIP 198701032014041001

LATAR BELAKANG

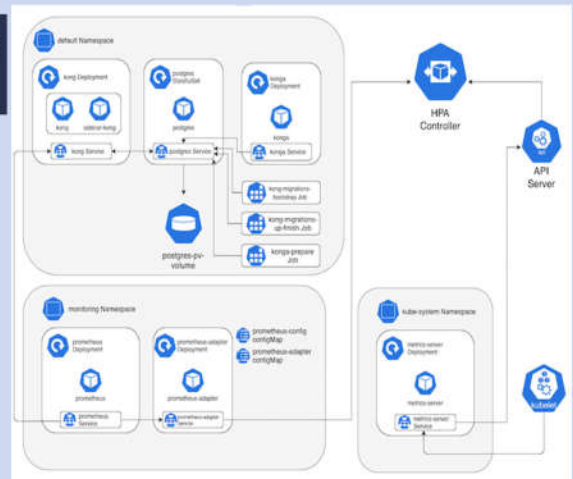
API gateway pada myITS merupakan sebuah aplikasi pusat untuk mengarahkan pengguna ke aplikasi yang dituju. Sehingga tingkat akses pada API gateway ini kemungkinan sangat tinggi. Jika hanya ada satu layanan saja maka aplikasi API gateway bisa kewalahan dalam melayani permintaan dari pengguna jadi diperlukan infrastruktur kluster yang adaptif sehingga bisa menyesuaikan jumlah layanan yang ada terhadap beban layanan pada waktu sebenarnya.

Custom metric pada layanan API gateway, diperlukan adanya suatu container tambahan yang mendukung container utama Kong gateway yang berfungsi untuk mengambil nilai koneksi aktif dan menyediakannya untuk digunakan sebagai salah satu parameter skala tanpa mengganggu fungsi container utama yaitu sebagai API gateway. Pola pemasangan container tambahan pada container utama ini bisa disebut dengan sidecar pattern.

METODE

1. Membuat sidecar container untuk mengambil nilai metrics pada container utama dan dipasang pada application.
2. Memasang prometheus untuk mendapatkan nilai metrics dari sidecar container.
3. Memasanga metric server agar bisa membaca metric CPU dan memori.
4. Memasang prometheus adapter agar nilai metric terdaftar dalam API kubernetes.
5. Memasang scaler dengan menggunakan HPA dengan menggunakan metrics yang telah dibuat.

ARSITEKTUR SISTEM



HASIL UJI COBA

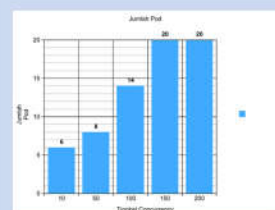
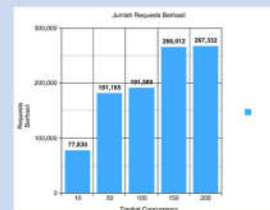
1. Sidecar Pattern berjalan sebagai mana seharusnya
2. Keenam komponen pada pembuatan kluster berjalan sebagaimana seharusnya

No	Tingkat Concurrent	Total Requests	Failed Requests	Persentase
1	1000	49876	1	0,002%
2	2000	46624	3	0,006%
3	3000	47869	1	0,002%
4	4000	44755	230	0,05%
5	5000	37311	189	0,51%
5	6000	49247	3415	6,93%
5	7000	35822	6644	18,55%

Kekuatan Pod Mengakses Admin API

No	Tingkat Concurrent	Total Requests	Failed Requests	Persentase
1	1000	2711	57	1,10%
2	1100	2675	149	5,57%
3	1200	2764	86	3,21%
4	1300	2749	180	6,55%
5	1400	2674	245	9,16%
6	1500	2656	263	9,90%
7	1600	3071	437	14,23%

Kekuatan Pod Mengakses Service Test



KESIMPULAN

1. Sistem dapat melakukan skala otomatis sesuai dengan metric yang telah ditentukan.
2. Sidecar pattern bekerja sesuai dengan fungsinya.
3. Custom metric yang telah dibuat dapat terbaca oleh HPA controller.
4. Kemampuan skala berjalan sebagaimana seharusnya.

SARAN

1. Kemampuan skala otomatis pada tugas akhir ini lebih difokuskan pada tipe deployment yang stateless sehingga tipe deployment dengan state masih belum terimplementasi dengan baik.
2. Proses pembaharuan image kong yang baru perlu disertai pre-pulling secara manual di setiap node agar image kong baru ada di penyimpanan lokal di node.



LATAR BELAKANG

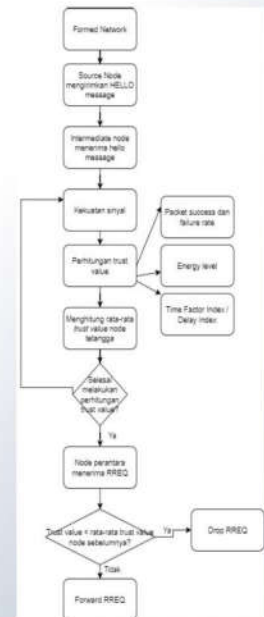
VANETs (vehicular ad hoc network) merupakan pengembangan dari MANETs (mobile ad hoc network) yang tidak terpengaruh pada infrastruktur jaringan. Berdasarkan cara membangun rute, routing protocol ad hoc dibagi menjadi dua, yakni proactive routing dan reactive routing protocol. AODV (ad hoc on-demand distance vector) termasuk ke dalam reactive routing protocol, yang mana hanya membangun rute ketika dibutuhkan. Pencarian rute menjadi suatu mekanisme yang penting untuk mendukung mobilitas di VANET. Pemilihan rute yang stabil saat proses pencarian rute sangat diperlukan untuk memperpanjang waktu penggunaan rute. Salah satu cara dapat dilakukan adalah dengan memilih rute yang node-nya mempunyai trust value yang tinggi. Ketika semua node melakukan rebroadcast RREQ, hal itu akan membutuhkan waktu yang lama dan resource yang banyak. Hal itu dapat mempengaruhi pada kestabilan rute dan dapat mengakibatkan bertambahnya delay dan mengurangi jumlah paket yang terkirim.

METODE

Melakukan seleksi forwarding node yang berhak melakukan forwarding route request berdasarkan rata-rata nilai trustness node tetangga

TAHAP PERANCANGAN MODIFIKASI

- Melakukan perhitungan nilai trustness suatu node berdasarkan 4 faktor, yakni: packet success rate, packet failure rate, energy level, time factor index / delay index.
- Melakukan perhitungan rata-rata trust value node tetangga yang dimiliki oleh node itu.
- Seleksi node yang berhak meneruskan paket RREQ. Node yang berhak meneruskan paket RREQ yakni node yang mempunyai trust value yang lebih tinggi dari rata-rata trust value node tetangga node sumber.



HASIL MODIFIKASI



Packet Delivery Ratio

End-to-end Delay



Routing Overhead

KESIMPULAN

- Penghitungan nilai trustness suatu node dihitung berdasarkan 4 faktor, yakni packet success rate, packet failure rate, energy level, dan time factor index / delay index.
- Rata – rata kenaikan PDR sebesar 15,39%
Rata – rata penurunan E2E Delay sebesar 31,76%,
Routing overhead meningkat 6-9 kali lipat

SARAN

- Menambahkan aspek lain untuk menghitung nilai trustness suatu node sebagai dasar untuk pembatasan forwarding node yang meneruskan paket RREQ seperti tingkat kepadatan node, kecepatan, arah, dan lain-lain.
- Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat.



IMPLEMENTASI CI/CD PIPELINE DENGAN JENKINS PADA LINGKUNGAN VIRTUAL MACHINE MENGUNAKAN ANSIBLE

Azkiatunnisa Rahma Fajriyati
05111640000106

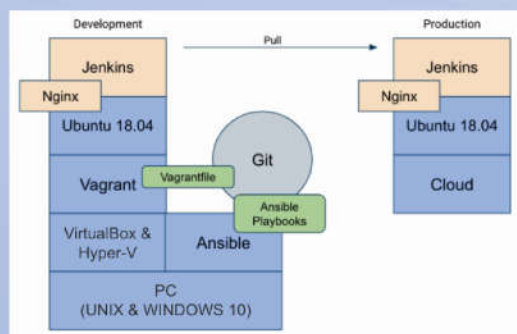
Dosen Pembimbing I
Royyana Muslim Ijtihadie, S.Kom, M.Kom, Ph.D
197708242006041001

Dosen Pembimbing II
Rizky Januar Akbar, S.Kom, M.Eng
198701032014041001

LATAR BELAKANG

Dengan banyaknya kebutuhan aplikasi pada myITS, dibutuhkan proses yang cepat dalam pengembangannya. Saat ini, pengembangan aplikasi di DPTSI cukup lambat karena harus melakukan pengaturan lingkungan manual di komputer lokal pengembang. Proses pengembangannya terdapat beberapa penyesuaian karena lingkungan pengembangan berbeda dengan produksi. Dari masalah itu, proses pengembangan ini difasilitasi *Virtual Machine* yang dijalankan dengan *tools* Vagrant untuk mempersingkat fase pengembangan. Dengan *Virtual Machine* dan Vagrant, proses pengembangannya lebih cepat dibandingkan dengan pengembang membuat lingkungan pengembangan sendiri di komputer lokalnya. Oleh karena itu, dirancahlah suatu alur pengembangan aplikasi yang mudah dibangun kembali pada komputer pengembang atau bisa disebut *reproducible* di *Virtual Machine* dengan bantuan Vagrant. Penggunaan *Virtual Machine* pada sistem ini dikarenakan lingkungan pengembangan awal di DPTSI sudah menggunakan *Virtual Machine*, sehingga penyesuaianannya lebih mudah. Kemudian terdapat skrip Ansible untuk melakukan tahapan pengembangan aplikasi, serta mengintegrasikan Git dengan Jenkins menggunakan *webhook* untuk melakukan proses *Continuous Integration-Continuous Delivery (CI/CD)*. Proses *CI/CD* sendiri berarti integrasi dan pengiriman yang berkelanjutan yang dilakukan secara otomatis.

ARSITEKTUR & METODE



Tahapan - tahapan implementasi yang dilakukan yaitu :

1. Pembuatan Vagrant box *Virtual Machine*
2. Pembuatan skrip Ansible untuk proses otomatisasi pengembangan
3. Pembuatan pipeline di Jenkins untuk proses *CI/CD*

Teknologi Pendukung:



HASIL

Deskripsi	Durasi	Hasil
Menjalankan vagrant box	-	OK
Menginstal ansible dan python	-	OK
Menginstal aptitude dan membuat user baru	2,03 menit	OK
Menginstal requirement, nginx, mysql, composer, php, membuat folder konfigurasi	2,62 menit	OK
Melakukan clone dari repositori github, menjalankan composer, membuat konfigurasi .env, menyesuaikan permission file dan directory, melakukan migrasi database	1,33 menit	OK
Pipeline berjalan	4 s	OK

Hasil Uji Coba pada Sistem Operasi UNIX

Deskripsi	Durasi	Hasil
Menjalankan vagrant box	-	OK
Menginstal ansible dan python	-	OK
Menginstal aptitude dan membuat user baru	3,78 menit	OK
Menginstal requirement, nginx, mysql, composer, php, membuat folder konfigurasi	5,02 menit	OK
Melakukan clone dari repositori github, menjalankan composer, membuat konfigurasi .env, menyesuaikan permission file dan directory, melakukan migrasi database	1,72 menit	OK
Pipeline berjalan	4 s	OK

Hasil Uji Coba pada Sistem Operasi Windows

KESIMPULAN & SARAN

1. Telah dibuat implementasi otomatisasi lingkungan pengembangan aplikasi "URL Shortener" berbasis *Virtual Machine* dengan Vagrant box.
2. Proses pengembangan dengan skema automation lebih memudahkan pengguna karena aplikasi jadi *reproducible* dengan bantuan Ansible di lingkungan *Virtual Machine*
3. Transisi antara lingkungan pengembangan dengan lingkungan produksi dapat tercapai dengan adanya *CI/CD Pipeline* dengan Jenkins.

Terdapat pula beberapa saran, yaitu

1. Pengembangan sistem ini dapat dilakukan dengan menggunakan platform yang lebih fleksibel di banyak lingkungan.
2. Pada tugas akhir ini, ansible hanya menjalankan skrip pada satu VM saja. Pengembangan sistem ini dapat dilakukan dengan membuat skrip ansible yang bisa mengembangkan banyak VM sekaligus.
3. Implementasi sistem ini dapat dikembangkan dengan

IMPLEMENTASI PROTOKOL REVERSE AD-HOC ON-DEMAND DISTANCE VECTOR (AODV) DENGAN MEMPERTIMBANGKAN FAKTOR KONSUMSI ENERGI PADA TAHAP ROUTING DI VANETS



Departemen Teknik Informatika
Fakultas Teknologi Elektro dan Informatika Cerdas
Institut Teknologi Sepuluh Nopember (ITS)
Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

DASAR TEORI

Vehicular Ad-hoc Networks (VANETs) adalah salah satu jaringan nirkabel yang memiliki node berupa kendaraan (*vehicle*). Tingkat mobilitas node yang sangat tinggi pada VANETs berperan penting dalam proses pengiriman data sehingga diperlukan juga informasi seperti jarak antar node, kecepatan, dan delay transmisi setiap node.

Ad-hoc On-demand Distance Vector (AODV) merupakan protokol routing yang termasuk dalam klasifikasi protokol routing reaktif, sebuah protokol yang hanya akan membuat rute pada saat dibutuhkan. Perilaku reaktif pada AODV tersebut dinilai sangat cocok untuk diimplementasikan pada VANETs. Salah satu tahap dalam AODV adalah *route discovery* yang terjadi ketika akan dilakukan pengiriman pesan pada jaringan, namun belum terdapat rute dari node sumber ke node tujuan. Pada *route discovery* terdapat dua pesan kontrol yaitu berupa *Route Request (RREQ)* dan *Route Reply (RREP)*.

PERMASALAHAN

Mekanisme pengiriman paket RREP oleh AODV yang hanya bergantung pada satu

1. rute (*unicast*) meningkatkan kemungkinan terjadinya *RREP packet loss* yang dikarenakan mobilitas node yang tinggi pada VANETs memungkinkan node penyusun rute tersebut bergerak keluar dari jangkauan transmisi.
2. Tidak adanya proses seleksi *forwarding node* pada proses pengiriman paket RREQ memungkinkan terbentuknya rute rentan putus yang disebabkan oleh salah satu atau lebih node penyusunnya kehabisan daya ditengah proses pengiriman data.

SOLUSI

1. Mengubah mekanisme pengiriman paket RREP dari *unicast* menjadi *broadcast* sehingga paket RREP akan dikirimkan melalui beberapa rute untuk mengurangi kemungkinan terjadinya *RREP packet loss*
2. Menambahkan filter energi pada proses *forwarding* paket RREQ untuk mengeliminasi node yang tidak memiliki energi cukup sehingga rute yang akan terbentuk tidak akan putus karena node penyusunnya kehabisan daya.

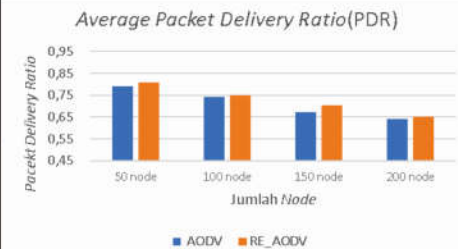
KESIMPULAN

1. Jumlah *Routing Overhead (RO)* atau jumlah pengiriman paket kontrol yang sangat meningkat pada AODV termodifikasi disebabkan oleh adanya perubahan mekanisme pengiriman RREP dari *unicast* menjadi *broadcast* sehingga menyebabkan pengiriman paket RREP secara masif.
2. AODV termodifikasi memiliki kinerja yang lebih baik daripada AODV asli dilihat dari peningkatan PDR dan *throughput*, serta penurunan E2E dan konsumsi energi

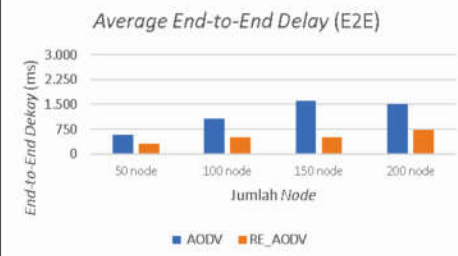
SARAN

1. Perlu dilakukan lebih banyak lagi uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat.
2. Menggunakan *threshold* dinamis agar dapat beradaptasi sesuai dengan kondisi lingkungan.
3. Menambahkan filter energi tidak hanya pada saat pengiriman *broadcast* paket RREQ, namun juga pada saat pengiriman *broadcast* paket RREP untuk mereduksi jumlah *routing overhead*.

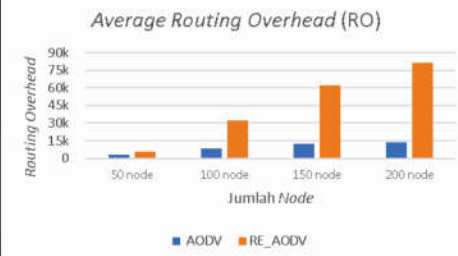
HASIL SIMULASI



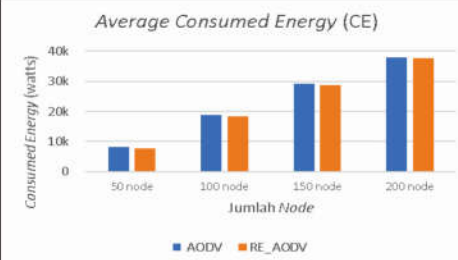
Peningkatan PDR sebesar 2,42%



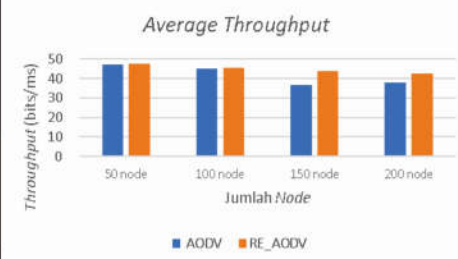
Penurunan E2E sebesar 56,1%



Peningkatan RO sebesar 314,59%



Penurunan CE sebesar 2,68%



Peningkatan throughput sebesar 8,66%



Christopher Andrew - 05111640000107

Dosen Pembimbing I : Dr. Eng. Raditya Anggoro, S.Kom, M.Sc.



IMPLEMENTASI K-MEANS CLUSTERING DENGAN METODE LINK COEFFICIENT PADA AODV UNTUK MENINGKATKAN STABILITAS RUTE DALAM MANETS

Akmal Darari Rafif B
05111640000148

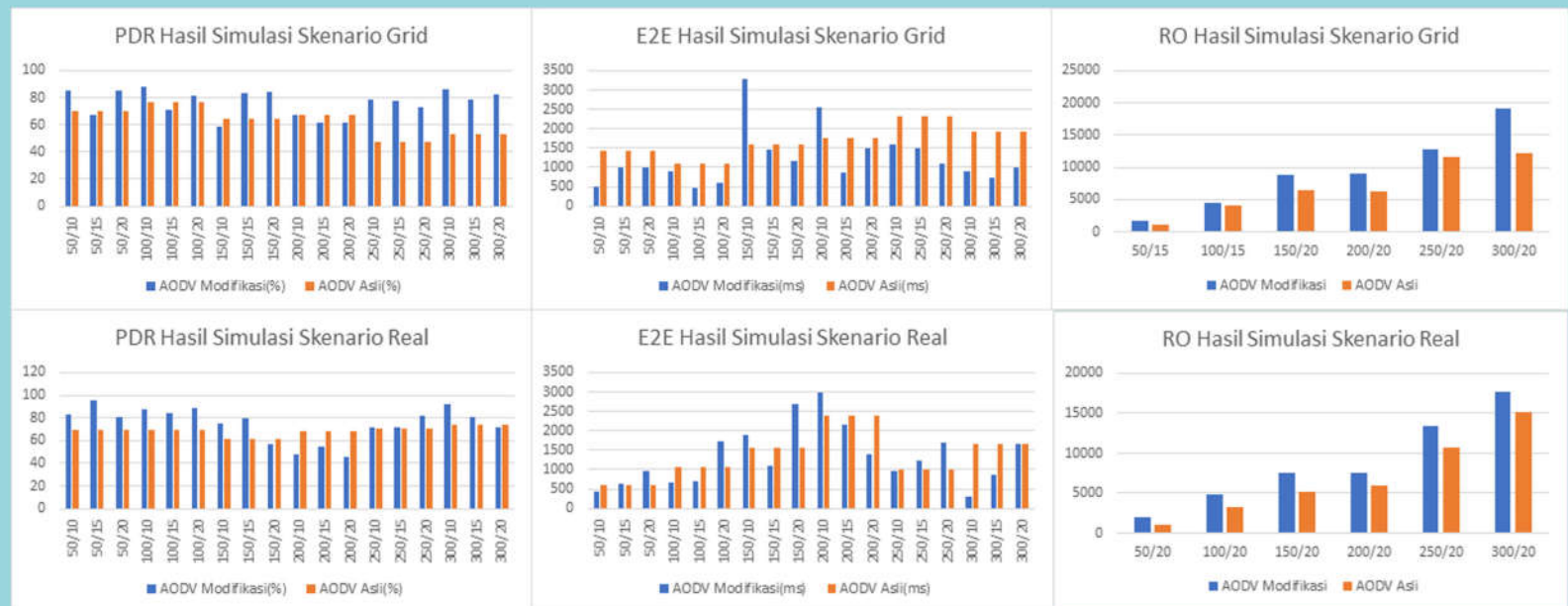
Dosen Pembimbing 1:
Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.
(NIP. 198410162008121002)

Dosen Pembimbing 2:
Ary Mazharuddin Shidiqi., S.Kom., M.Comp.Sc., Ph.D.
(NIP. 198106202005011003)

LATAR BELAKANG:

Mobile Ad-hoc Network (MANET) adalah jaringan nirkabel yang terdiri dari kumpulan mobile node yang bersifat dinamis dan spontan, dan dapat diaplikasikan dimana pun tanpa menggunakan jaringan infrastruktur yang telah ada. Dewasa ini, MANETs menjadi topik hangat para peneliti dan pengembang. Namun, karena pergerakan MANETs yang dinamis akan membuat rentan dalam serangan sekaligus juga pemilihan jalur juga hal yang vital dalam jaringan ini. Karena itu, dibutuhkan stabilitas dalam pemilihan rute dalam MANETs. Pada tugas akhir ini akan dibahas sebuah metode yang dapat menstabilkan pemilihan rute yang dipilih pada saat berkomunikasi, yaitu dengan penerapan metode clustering. Routing ini bergantung pada tingkat energi dan perpindahan node pada suatu jaringan. Perkiraan biaya energi dihitung berdasarkan energi yang digunakan pada suatu node dalam penerapan K-Means Clustering yang diikuti oleh AODV dengan faktor Coefficient Link dengan perhitungan kecepatan antar node supaya dapat meningkatkan stabilitas pada saat berkomunikasi dengan menggunakan MANETs.

HASIL MODIFIKASI:



KESIMPULAN:

- Penerapan K-Means Clustering telah berhasil mengurangi jumlah forwarding node secara signifikan dan setelah dikombinasikan dengan Coefficient Link secara umum menghasilkan hasil yang baik dari sisi Packet Delivery Ratio, dan End-to-End Delay. Namun kombinasi ini juga menyebabkan Routing Overhead meningkat.
- Pengaruh implementasi K-Means Clustering dan Coefficient Link terhadap performa AODV pada skenario grid adalah rata-rata kenaikan PDR sebesar 21,69%, rata-rata penurunan E2E sebesar 67,6%, dan rata-rata kenaikan RO sebesar 241,95%
- Pengaruh implementasi K-Means Clustering dan Coefficient Link terhadap performa AODV pada skenario real adalah rata-rata kenaikan PDR sebesar 13,15%, rata-rata penurunan E2E sebesar 49,3%, rata-rata kenaikan RO sebesar 201,316%

SARAN:

- Melakukan uji coba dengan variasi yang lebih variatif untuk mendapatkan hasil yang lebih akurat.
- Menerapkan metode lain pada K-Means Clustering agar memperoleh jumlah cluster yang optimal.
- Menerapkan sebuah metode untuk menstabilkan hasil E2E pada simulasi skenario Real supaya hasilnya tidak fluktuatif dan grafiknya cenderung membaik seperti pada simulasi skenario Grid.



Achmad Hanif Pradipta
NRP 05111640000154

Dosen Pembimbing
Dr. Eng. Radityo Anggoro, S. Kom., M.Sc.
Ir. F.X. Arunanto, M.Sc.

LATAR BELAKANG

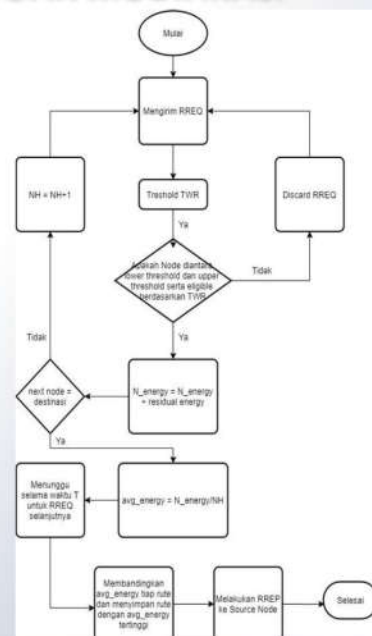
Mobile Adhoc Network (MANET) merupakan teknologi jaringan yang tersusun dari kumpulan ad-hoc node yang berbeda-beda atau sejenis dan setiap node pada MANET terhubung dengan node lain menggunakan koneksi wireless. Permasalahan yang ada pada MANET adalah topologi jaringan yang sering berubah atau bersifat dinamis. Oleh karena itu protokol routing sangat diperlukan untuk menentukan rute atau jalur pada jaringan MANET. Dalam Mobile Adhoc Network (MANET) selama proses routing, konsumsi energi tetap menjadi hambatan karena node seluler memiliki baterai yang terbatas. Salah satu cara dapat dilakukan adalah dengan melakukan penghitungan energi rata-rata dan jalur energi dengan berdasarkan pembagian zona yang bertujuan untuk mengurangi pengiriman paket secara terus-menerus dan mengurangi jumlah hop di route discovery serta melakukan pemilihan rute Route Reply (RREP) berdasarkan rata-rata energi tertinggi yang diterima.

METODE

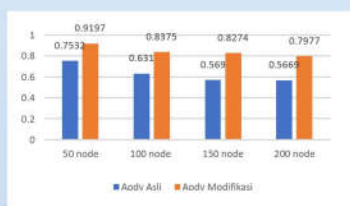
Melakukan metode selection based berdasarkan energy path selection pada MANET

TAHAP PERANCANGAN MODIFIKASI

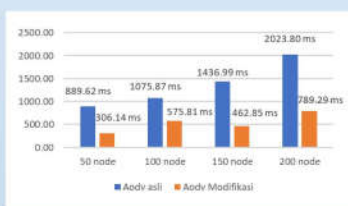
- Melakukan perhitungan nilai Total Weight of the Route (TWR)
- Melakukan pemilihan forwarding node berdasarkan penerimaan kekuatan sinyal
- Melakukan Pemilihan rute Route Reply (RREP) berdasarkan rata-rata residu energi tertinggi yang diterima dan melakukan RREP kembali sebagai backup rute apabila terjadi kegagalan saat proses route discovery untuk diteruskan menuju source node



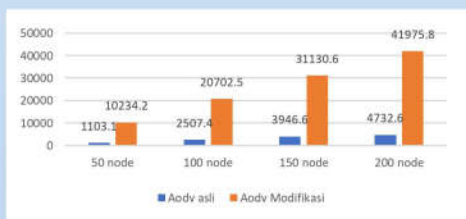
HASIL MODIFIKASI



Packet Delivery Ratio



End-to-end Delay



Routing Overhead

KESIMPULAN

- Dengan menambahkan modifikasi pemilihan rute RREP berdasarkan energy path selection serta penjumlahan nilai Total Weight of the Route (TWR) dan pemilihan forwarding node berdasarkan penerimaan kekuatan sinyal maka dapat meningkatkan efisiensi pada routing protocol AODV
- Rata-rata kenaikan PDR sebesar 35,24%
- Rata-rata penurunan E2E Delay sebesar 60,21%
- Routing overhead meningkat 6-9 kali lipat

SARAN

- Menambahkan aspek lain untuk mempercepat proses route discovery seperti arah, kecepatan, kepadatan node dan aspek lainnya.
- Lebih banyak uji coba yang dilakukan untuk mendapatkan hasil yang lebih akurat.

Teknologi Supporting





PENGEMBANGAN PERANGKAT LUNAK INFRASTRUKTUR CLOUD MENGGUNAKAN DOCKER DAN JENKINS

Oleh
Hilmi Raditya Prakoso
NRP 05111640000164

Dosen Pembimbing I
Royyana Muslim Ijtihadie, S.Kom., M.Kom., Ph.D.
NIP 197708242006041001

Dosen Pembimbing II
Rizky Januar Akbar, S.Kom., M.Eng.
NIP 194806191973011001

Latar Belakang

Deployment adalah kegiatan yang bertujuan untuk menyebarkan aplikasi yang telah dikerjakan oleh para pengembang. Penyebarannya dapat menggunakan berbagai macam strategi, salah satunya adalah strategi Shadow Deployment.

Dengan menggunakan metode strategi Shadow Deployment, diharapkan proses deployment dapat berjalan di pipeline dengan error dan downtime seminimal mungkin Serta meningkatkan efektivitas dalam proses deployment dalam lingkungan Kubernetes pada tahap production dengan mengimplementasikan pipeline sehingga deployment dapat dilakukan secara otomatis.

Rumusan Masalah & Tujuan

Rumusan Masalah dalam Tugas Akhir ini adalah :

- Bagaimana menerapkan Jenkins dan Kubernetes dalam proses Pipeline?
- Bagaimana melakukan pengalihan traffic request secara otomatis ke versi terbaru?
- Bagaimana membuat alur pipeline dengan downtime se-minimal mungkin?

Tujuan dalam Tugas Akhir ini adalah :

- Dapat mengintegrasikan Git, Docker, Helm, Jenkins, Kubernetes serta Istio
- Mengetahui mekanisme pengalihan traffic menggunakan service mesh Istio
- Membuat alur deployment secara automatic, sehingga proses lebih efisien

Hasil Uji Coba

No	Skenario Uji Fungsionalitas	Uji Coba Kong Gateway	Uji Coba Laravel
1	Tidak terdapat error pada aplikasi yang akan di deploy, sehingga proses deployment berjalan secara normal.	Sukses	Sukses
2	Terdapat error pada aplikasi yang akan di deploy, sehingga terjadi pembatalan pada deployment dan pipeline pada jenkins melakukan rollback ke aplikasi versi sebelumnya.	Kong tidak dapat berjalan apabila ada error, dikarenakan pada saat proses build image terdapat validation. Sehingga pengujian dengan case Terdapat error pada aplikasi yang akan di deploy pada studi kasus Kong Api Gateway tidak dilakukan.	Sukses

Uji Coba Fungsionalitas

No	Skenario Uji Fungsionalitas	Performa Down Time Kong Gateway	Performa Down Time Framework Laravel
1	Tidak terdapat error pada aplikasi yang akan di deploy, sehingga proses deployment berjalan secara normal.	4%	0%
2	Terdapat error pada aplikasi yang akan di deploy, sehingga terjadi pembatalan pada deployment dan pipeline pada jenkins melakukan rollback ke aplikasi versi sebelumnya.	Kong tidak dapat berjalan apabila ada error, dikarenakan pada saat proses build image terdapat validation. Sehingga pengujian dengan case Terdapat error pada aplikasi yang akan di deploy pada studi kasus Kong Api Gateway tidak dilakukan.	0%

Uji Coba Performa

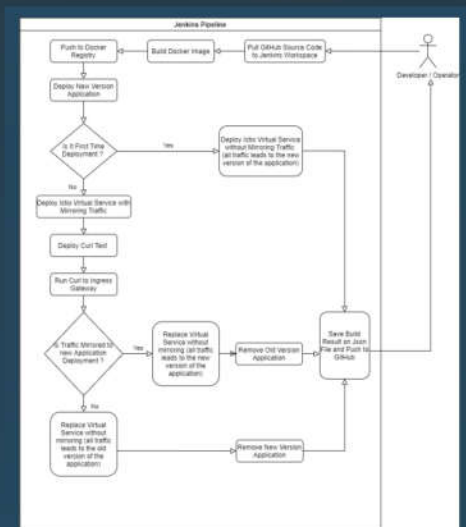
Kesimpulan

- Zero Downtime telah sukses dilakukan dengan uji kasus Framework Laravel.
- Pada uji kasus Kong memiliki error request sebesar 1% - 6%, sehingga Zero Downtime dapat dikatakan hampir sukses.
- Telah diimplementasikan strategi shadow deployment yang jauh lebih baik dibandingkan dengan blue/green atau rolling update karena shadow deployment terdapat fitur mirroring traffic yang berguna untuk melakukan pengecekan apakah aplikasi siap menerima traffic request dengan sukses atau tidak.

Saran

- Sistem dapat dikembangkan dengan melakukan optimasi zero downtime pada saat melakukan deployment kong
- Pada tugas akhir ini, pada bagian mirroring traffic yang bertujuan untuk menghindari error pada lingkungan produksi hanya menggunakan curl dan mengakses suatu endpoint pada suatu pod, Untuk tahap pengembangan kedepannya mirroring traffic dapat diintegrasikan dengan PHP Unit Test untuk studi kasus Framework Laravel.

Perancangan



Alur Pipeline

Oleh :
Ismail Syarief
05111640000168

Pembimbing I :
Royyana Muslim Ijtihadie S.Kom, M.Kom., Ph.D
197708242006041001

Pembimbing II :
Rizky Januar Akbar, S.Kom., M.Eng.
198701032014041001

Latar Belakang

Dengan berkembangnya teknologi di bidang cloud-native, platform myITS mengimplementasikan Kubernetes sebagai platform orkestrasinya, dengan kebutuhan aplikasi yang kian bertambah, mendatang akan ada aplikasi baru yang terus ditambahkan pada platform myITS. Demi meningkatkan efektivitas dalam integrasi aplikasi baru dengan lingkungan Kubernetes pada platform myITS, diperlukan sebuah pipeline CI/CD untuk menunjang kebutuhan tersebut. Suatu aplikasi perlu dikembangkan pada lingkungan yang serupa dengan lingkungan produksi, dengan ini perlu adanya sebuah standarisasi struktur atau scaffold dalam melakukan pengembangan dari suatu aplikasi. Kubernetes pada platform myITS melakukan orkestrasi terhadap kontainer docker, yang berarti seluruh aplikasi dijalankan di dalam satu atau lebih kontainer docker, suatu lingkungan pengembangan juga perlu bersifat reusable dan reproducible, maksud reusable itu sendiri adalah suatu lingkungan kerja dapat digunakan untuk mengembangkan aplikasi lain dan reproducible adalah suatu lingkungan kerja dapat dibuat kembali menjadi lingkungan pengembangan ataupun dibuat kembali pada lingkungan produksi, dengan itu kontainer docker akan digunakan sebagai scaffold pengembangan bagi programmer. Pipeline CI/CD akan dijalankan pada sebuah aplikasi bernama Jenkins, Jenkins akan mengeksekusi setiap tahapan dari pipeline CI/CD, Jenkins akan mengemas kode sumber pada scaffold pengembang menjadi image yang dapat dijalankan oleh kontainer docker pada kluster kubernetes myITS, lingkungan pengembang yang berjalan pada docker dengan lingkungan produksi yang berjalan pada Kubernetes dapat terintegrasi dengan baik dikarenakan kesamaan media yang dijalankan yaitu image, dengan kesamaan tersebut aplikasi yang dikemas dalam image dapat dijalankan pada docker dan Kubernetes meskipun docker dan Kubernetes berasal dari vendor yang berbeda. Maka tugas akhir ini bertujuan untuk merancang dan mengimplementasikan pipeline CI/CD serta merancang scaffold lingkungan pengembangan bagi programmer

Rumusan Masalah

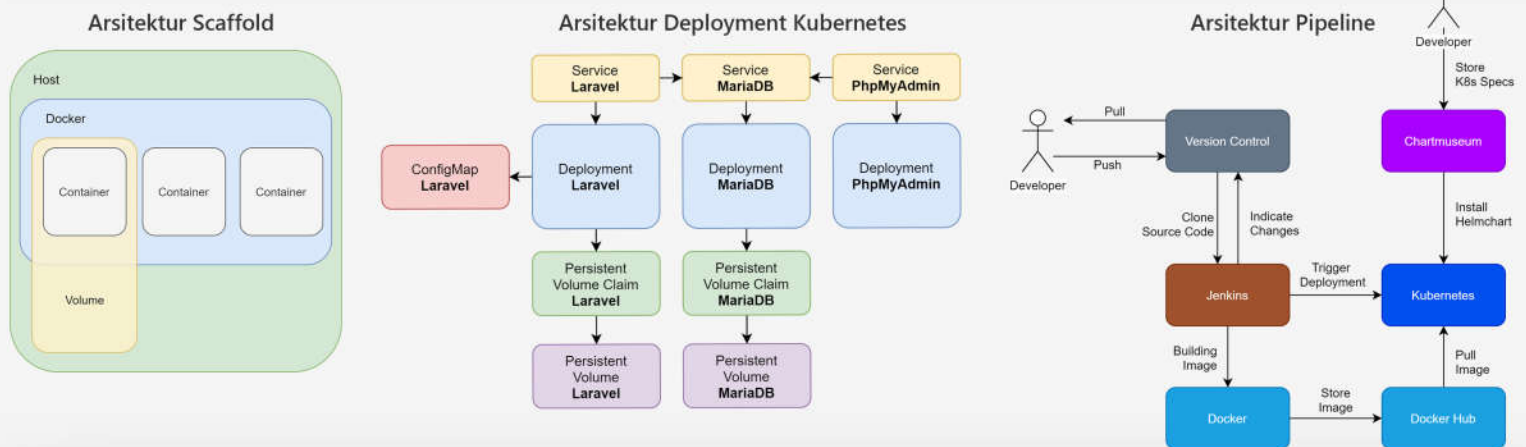
- Bagaimana menggunakan kontainer docker sebagai standar scaffolding pada lingkungan pengembang.
- Bagaimana menerapkan Jenkins dan Kubernetes dalam proses pipeline CI/CD
- Bagaimana mengintegrasikan lingkungan pengembang dengan lingkungan produksi?
- Bagaimana membuat lingkungan pengembangan yang reusable dan reproducible?

Deskripsi Umum Sistem

Proses pengembangan akan berjalan pada kontainer docker, hal tersebut bertujuan untuk menjaga standar suatu lingkungan pengembang, supaya tetap konsisten sesuai dengan yang ada pada lingkungan produksi. Jenkins berperan sebagai integrator utama dari kedua lingkungan. Proses berjalannya pipeline diawali dengan programmer melakukan commit terbaru pada repositori.

Webhook yang terhubung akan mendeteksi perubahan yang terjadi, selanjutnya Jenkins akan menjalankan perintah untuk melakukan pull dan build image terbaru, kemudian menyimpannya pada DockerHub, setelah itu Jenkins akan mengeksekusi perintah Helm untuk mengeksekusi Helmchart yang berupa konfigurasi dari deployment Kubernetes. Helm akan melakukan pemasangan atau perbaruan dari sebuah aplikasi mengikuti konfigurasi yang ada pada Helmchart.

Desain Sistem



Kesimpulan

- Telah diimplementasikan kontainer docker sebagai standar scaffold pada lingkungan pengembang.
- Telah diimplementasikan Jenkins untuk mengintegrasikan lingkungan kerja pengembang dengan menerapkan standarisasi scaffold kepada lingkungan produksi yang berjalan pada kluster kubernetes.
- Integrasi dapat tercapai dengan mengimplementasikan scaffold pengembangan sehingga aplikasi pada lingkungan pengembang dapat dengan mudah dijalankan pada lingkungan produksi, dikarenakan kesamaan media yang digunakan yaitu image.

Lingkungan pengembangan yang reusable dan reproducible dapat dibuat dengan mengimplementasikan kontainer docker yang didefinisikan dalam docker-compose.

Saran

- Sistem dapat dikembangkan dengan melakukan optimasi saat melakukan build image untuk mempersingkat durasi build pada pipeline.

Didukung Oleh:



DESIGN AND IMPLEMENTATION OF TRAVEL INFORMATION SYSTEM IN FIJI USING REST-API CONCEPT

By :
Nemesio R. Raitubu
NRP: 05111640007006

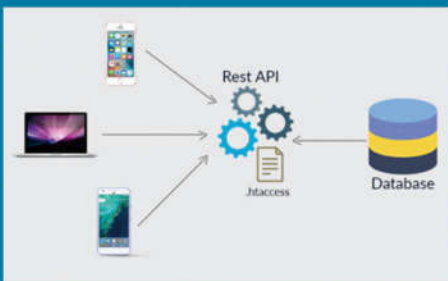
Thesis Supervisor I :
Radityo Anggoro, S.Kom., M.Sc
NIP: 19841016 2008121002

BACKGROUND

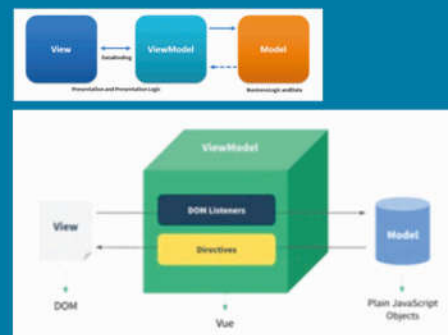
Nearly everyone goes on a vacation and a Travel information system would play a vital role in planning the perfect trip. The travel information system allows the user of the system access all the details such as hotels and flights. The main purpose is to help individual to manage customer, flights and hotels. The system can also be used for both professional and business trips. The proposed system maintains centralized repository to make necessary travel arrangements and to retrieve information easily.

With the rapid economic development in Fiji, the government has invested a lot of money into the tourism industry for the construction of many hotels to attract tourists to choose Fiji as their holiday destination. Therefore, Travel Information System in Fiji will be very useful for the tourists to use in booking flights and hotels.

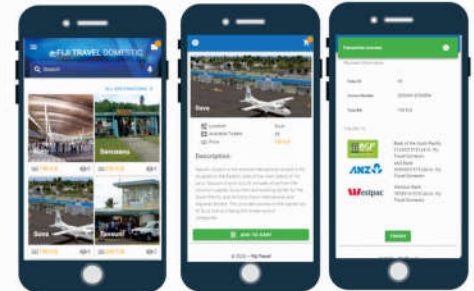
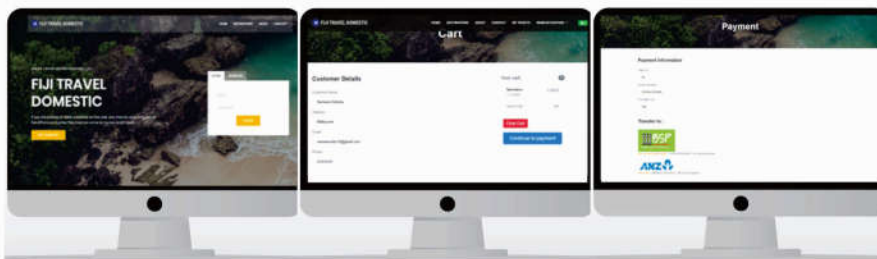
SYSTEM ARCHITECTURE



DESIGN PATTERN



SCREENSHOTS



CONCLUSIONS

1. It can be concluded that Travel Information System if Fiji conceptualization promises to provide a firmer base for Web applications development at the initial stages of design. It is possible to integrate this system approach with current methods of software development processes.
2. The travel Information System in Fiji is user-friendly and still allowing researchers for improvements in the future.

RECOMMENDATIONS

1. The future researchers would be able to complete the payment process by producing flight tickets and Itinerary.
2. Need to explore further needs in coding with relevant users for future system development.
3. And lastly, I would strongly recommend that future researchers to explore more on this project to develop a good International Information System.