



**PENENTUAN LOKASI *ROUTER* UNTUK OPTIMASI JARINGAN
KOMPUTER DENGAN MENGGUNAKAN METODE *DOMINATING SET*
(Studi Kasus: Universitas Jember)**

SKRIPSI

Oleh

IVAN DESTIOVIKO HARDJA

NIM 132410101022

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2018



**PENENTUAN LOKASI *ROUTER* UNTUK OPTIMASI JARINGAN
KOMPUTER DENGAN MENGGUNAKAN METODE *DOMINATING SET*
(Studi Kasus: Universitas Jember)**

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Studi Sistem Informasi (S1) dan mencapai gelar Sarjana
Komputer

Oleh

IVAN DESTIOVIKO HARDJA

NIM 132410101022

**PROGRAM STUDI SISTEM INFORMASI
UNIVERSITAS JEMBER**

2018

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT, yang telah memberikan nikmat sehat dan akal sehingga saya bisa menyelesaikan skripsi ini;
2. Orang tua saya ibu Lidia Sri Waningsih dan mbah Djamiati yang telah mendidik saya hingga sekarang. Mereka adalah pahlawan bagi saya. Sejak baru lahir hingga sekarang telah memberikan banyak limpahan kasih sayang, doa, dan motivasi serta dukungan yang luar biasa;
3. Kakak saya Frida Oktavika Hardja yang selalu membantu dan menasehati saya ketika salah;
4. Saudara-saudara saya keluarga besar saya yang tidak pernah berhenti memberikan dukungan dan doa;
5. Sahabat-sahabat saya yang di kontrakan Gurpong, Intention 2013, Kadal BBM
6. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

Jika kita tak bisa mengajak orang lain ke surga, setidaknya jangan membantu mereka untuk masuk neraka.



PERNYATAAN

Saya yang bertanda tangan dibawah ini:

Nama : Ivan Destioviko Hardja

NIM : 132410101022

Menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Penentuan Lokasi *Router* Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode *Dominating Set* (Studi Kasus: Universitas Jember)”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 8 Juni 2018

Yang menyatakan,

Ivan Destioviko Hardja

NIM 132410101022

SKRIPSI

**PENENTUAN LOKASI *ROUTER* UNTUK OPTIMASI JARINGAN
KOMPUTER DENGAN MENGGUNAKAN METODE *DOMINATING SET*
(Studi Kasus: Universitas Jember)**

oleh:

Ivan Destioviko Hardja

NIM 132410101022

Pembimbing

Dosen Pembimbing Utama : Prof. Drs. Slamin, M.Comp.Sc., Ph.D.

Dosen Pembimbing Pendamping : Nova El Maidah, S.Si.,M.Cs

PENGESAHAN PEMBIMBING

Skripsi berjudul “Penentuan Lokasi *Router* Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode *Dominating Set* (Studi kasus Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal :Jum’at, 8 Juni 2018

tempat :Program Studi Sistem Informasi Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D.
NIP. 1967042011992011001

Nova El Maidah, S.Si.,M.Cs
NIP. 198411012015042001

PENGESAHAN PENGUJI

Skripsi berjudul “Penentuan Lokasi *Router* Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode *Dominating Set* (Studi kasus Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 8 Juni 2018

tempat : Program Studi Sistem Informasi Universitas Jember.

Tim Penguji:

Penguji I,

Penguji II,

Nelly Oktavia A, S.Si., MT.
NIP. 1984102420009122008

Windy Eka Yulia Retnani S.Kom.,MT.
NIP. 196811131994121001

Mengesahkan
Pejabat Dekan,

Prof. Drs. Slamini, M.Comp.Sc.,Ph.D.
NIP 196704201992011001

RINGKASAN

Internet merupakan suatu jaringan komputer raksasa yang saling terhubung dan dapat berinteraksi. Hal ini terjadi karena adanya perkembangan teknologi jaringan yang sangat cepat, sehingga dalam beberapa tahun saja jumlah pengguna komputer yang tergabung dalam internet telah berlipat ganda. Karena itu jaringan komputer bukan suatu yang baru saat ini dan hampir di setiap perusahaan, instansi, maupun perguruan tinggi terdapat jaringan komputer.

Penentuan letak *Router* di perguruan tinggi Universitas Jember harus dilakukan secara optimal dengan meletakkan pada tempat yang strategis agar dapat meminimumkan jumlah *Router* menggunakan Teori Graf.

Dalam penelitian ini Teori Graf digunakan untuk mengetahui titik pendominasi pada graf *Router* di Universitas Jember. *Dominating set* diterapkan pada pencarian titik lokasi *Router* di Universitas Jember untuk mengetahui *Router* mana yang mendominasi *Router* lain. Pencarian titik pendominasi menggunakan algoritma *Greedy*. Algoritma *greedy* adalah salah satu algoritma yang sering dipakai dalam ilmu matematika dan komputer. Algoritma *Greedy* merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya. Nilai maksimum sementara ini dikenal dengan istilah *local maximum*. Algoritma *Greedy* biasanya memberikan solusi yang mendekati nilai optimum dalam waktu yang cukup cepat.

Penelitian ini bertujuan untuk optimasi dalam penentuan lokasi *Router* yang berada pada Universitas Jember dengan menggunakan *dominating set*. Algoritma pencarian yang digunakan untuk memecahkan masalah tersebut adalah algoritma *Greedy*.

PRAKATA

Puji syukur kehadirat Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Penentuan Lokasi Router Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode Dominating Set (Studi Kasus: Universitas Jember)”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terimakasih kepada:

1. Orang tua saya ibu Lidia Sriwaningsih dan nenek Djamiati yang telah mendidik saya hingga sekarang. Mereka adalah pahlawan bagi saya. Selama ini telah memberikan banyak limpahan kasih sayang, doa, dan motivasi serta dukungan yang luar biasa selama hidup saya;
2. Prof. Drs. Slamin, M.Comp., Ph.D., selaku Dosen Pembimbing Utama dan Nova El Maidah, S.Si.,M.Cs selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran dan perhatian dalam penulisan skripsi;
3. Dr. Saiful Bukhori, S.T., M.Kom.selaku Dosen Pembimbing Akademik yang telah membimbing selama penulis menjadi mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Kakak saya Frida Oktavika Hardja yang selama ini menjadi motivasi terbesar saya;
6. Sahabat seperjuanganku selama masa kuliah, Rois Ma'ruf, Sofyan Saury, Iqbal Maulana, Rizal Ilhafa, Muhammad Ismawan Zain, Nur hidayat, Ibnu firmansyah, Arief Setiawan.
7. Keluarga besar INTENTION semua mahasiswa Program Studi Sistem Informasi angkatan 2013 yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
8. Semua pihak yang tidak bisa disebutkan satu-persatu.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 8 Juni 2018

Penulis



DAFTAR ISI

| | |
|--------------------------------------|------|
| SKRIPSI..... | ii |
| PERSEMBAHAN..... | iii |
| MOTO..... | iv |
| PERNYATAAN..... | v |
| SKRIPSI..... | vi |
| PENGESAHAN PEMBIMBING..... | vii |
| PENGESAHAN PENGUJI..... | viii |
| RINGKASAN..... | ix |
| PRAKATA..... | x |
| DAFTAR ISI..... | xii |
| DAFTAR GAMBAR..... | |
| DAFTAR TABEL..... | |
| DAFTAR LAMPIRAN..... | i |
| BAB. 1 PENDAHULUAN..... | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Masalah..... | 2 |
| 1.3 Tujuan..... | 3 |
| 1.4 Manfaat..... | 3 |
| 1.5 Batasan Masalah..... | 3 |
| 1.6 Sistematika Penulisan..... | 3 |
| BAB. 2 TINJAUAN PUSTAKA..... | 13 |
| 2.1 Penelitian terdahulu..... | 13 |
| 2.2 Dominating Set..... | 18 |
| 2.3 Algoritma <i>Greedy</i> | 19 |

| | | |
|--------|---|----|
| BAB. 3 | METODOLOGI PENELITIAN | 21 |
| 3.1. | Tempat dan Waktu Penelitian | 21 |
| 3.2. | Tahap Pengumpulan Data..... | 21 |
| 3.2.1 | Studi Literatur | 21 |
| 3.2.2 | Studi Lapangan..... | 21 |
| 3.2.3 | Wawancara..... | 21 |
| 3.3 | Tahap Analisis Data | 22 |
| 3.4 | Tahap Pengembangan Sistem..... | 23 |
| 3.4.1 | Analisis Kebutuhan Sistem Analisis Kelayakan Investasi..... | 23 |
| 3.4.2 | Desain sistem analisis kelayakan investasi..... | 23 |
| 3.4.3 | Implementasi..... | 24 |
| 3.4.4 | Testing | 24 |
| 3.4.6 | Perawatan (<i>Maintenance</i>) | 25 |
| 3.5 | Gambaran Sistem | 25 |
| BAB. 4 | PENGEMBANGAN SISTEM..... | 18 |
| 4.1 | Analisis Kebutuhan | 18 |
| 4.1.1 | Kebutuhan Fungsional | 18 |
| 4.1.2 | Kebutuhan Non-Fungsional | 18 |
| 4.2 | Desain Sistem | 19 |
| 4.2.1 | <i>Business Process</i> | 19 |
| 4.2.2 | <i>Use Case Diagram</i> | 20 |
| 4.2.3 | Definisi Aktor | 20 |
| 4.2.4 | Definisi <i>Use Case</i> | 21 |
| 4.2.5 | <i>Scenario</i> | 22 |
| 4.2.6 | <i>Sequance Diagram</i> | 25 |

| | | |
|--------|---|----|
| 4.2.7 | <i>Activity Diagram</i> | 28 |
| 4.2.8 | <i>Class Diagram</i> | 29 |
| 4.2.9 | <i>Entity Relationship Diagram (ERD)</i> | 30 |
| 4.3 | Penulisan Kode Program..... | 30 |
| 4.3.1 | Kode Program <i>Login</i> | 31 |
| 4.3.2 | Kode Program Pengelolaan Data <i>Router</i> | 31 |
| 4.3.3 | Kode Program Pengelolaan Data <i>Point Neighbor</i> | 31 |
| 4.3.4 | Kode Program Melihat Hasil Perhitungan..... | 31 |
| 4.3.5 | Kode Program Pengelolaan Data <i>User</i> | 32 |
| 4.3.6 | Kode Program Melihat Data <i>Router</i> | 32 |
| 4.4 | Pengujian Sistem..... | 32 |
| 4.4.1 | Pengujian <i>White Box</i> | 32 |
| 4.4.2 | Pengujian <i>Black Box</i> | 35 |
| BAB. 5 | HASIL DAN PEMBAHASAN..... | 35 |
| 5.1 | Tampilan Sistem..... | 35 |
| 5.1.1 | Tampilan halaman <i>Login</i> | 35 |
| 5.1.2 | Tampilan halaman <i>Dashboard</i> | 36 |
| 5.1.3 | Tampilan halaman <i>Router Location</i> | 36 |
| 5.1.4 | Tampilan halaman <i>Point Neighbor</i> | 37 |
| 5.1.5 | Tampilan halaman <i>Analysis</i> | 38 |
| 5.1.6 | Tampilan halaman <i>User Management</i> | 40 |
| 5.2 | Metode Sistem..... | 40 |
| 5.2.1 | Penentuan <i>Vertex</i> | 41 |
| 5.2.2 | Penentuan Titik Dominator..... | 42 |
| 5.2.3 | Penentuan Titik Sisa..... | 43 |

| | | |
|--------|---|----|
| 5.2.4 | Penentuan Titik Sisa Dominator | 44 |
| 5.2.5 | Optimasi perhitungan | 44 |
| 5.3 | Hasil Perhitungan Sistem | 45 |
| BAB. 6 | PENUTUP | 48 |
| 6.1 | Kesimpulan..... | 48 |
| 6.2 | Saran | 49 |
| | DAFTAR PUSTAKA | 50 |
| | LAMPIRAN..... | 52 |
| A. | Skenario..... | 52 |
| A.1 | Skenario Login..... | 52 |
| A.2 | Skenario Mengelola Data <i>User</i> | 54 |
| A.3 | Skenario Mengelola Data <i>Router</i> | 57 |
| A.4 | Skenario melihat Data <i>Router</i> | 63 |
| A.5 | Skenario Mengelola <i>Point Neighbor</i> | 64 |
| A.6 | Skenario Melihat hasil perhitungan <i>dominating set</i> dan <i>algoritma greedy</i> | 68 |
| B. | Activity Diagram | 73 |
| B.1 | <i>Activity Diagram Login</i> | 73 |
| B.2 | <i>Activity Diagram Mengelola Data User</i> | 75 |
| B.3 | <i>Activity Diagram Mengelola Data Router</i> | 76 |
| B.4 | <i>Activity Diagram Melihat Data Router</i> | 78 |
| B.5 | <i>Activity Diagram Mengelola Point Neighbor</i> | 79 |
| B.6 | <i>Activity Diagram</i> melihat hasil Perhitungan <i>Dominating Set</i> dan <i>algoritma greedy</i> | 81 |
| C. | Sequance Diagram..... | 84 |

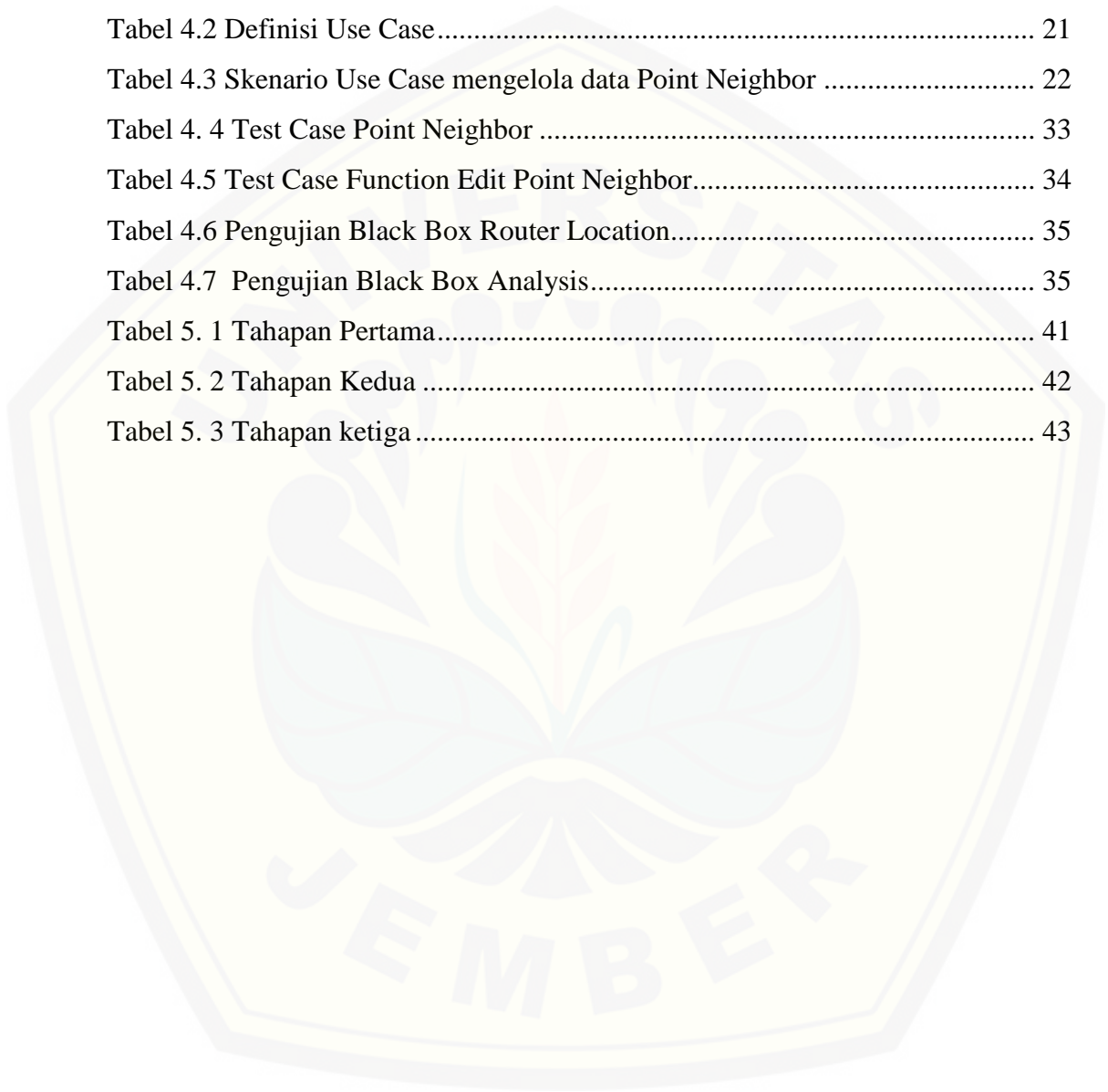
| | | |
|-----|--|-----|
| C.1 | <i>Sequance Digram Login</i> | 84 |
| C.2 | <i>Sequance Digram Mengelola Data User</i> | 85 |
| C.3 | <i>Sequance Digram Mengelola Data Router</i> | 87 |
| C.4 | <i>Sequance Digram Melihat Data Router</i> | 89 |
| C.5 | <i>Sequance Digram Mengelola Point Neighbor</i> | 89 |
| C.6 | <i>Sequance Digram Melihat hasil perhitungan</i> | 92 |
| D. | Kode Program..... | 93 |
| D.1 | Kode Program <i>login</i> | 93 |
| D.2 | Pengolahan Data <i>Router</i> | 94 |
| D.3 | Pengelolaan data <i>point neighbor</i> | 96 |
| D.4 | Melihat hasil Perhitungan | 98 |
| D.5 | Pengelolaan data <i>user</i> | 100 |
| D.6 | Melihat data <i>Router</i> | 101 |

DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2. 1 Graf G dengan Pedominsinya (1,3,5) (Iswadi, 2011) | 18 |
| Gambar 3. 1 Diagram Alir Proses Perhitungan Algoritma Greedy dan Dominating set | 22 |
| Gambar 3. 2 Model Waterfall (Sommerville, 2011) | 23 |
| Gambar 4. 1 Business Proses..... | 19 |
| Gambar 4. 2 Use Case Diagram | 20 |
| Gambar 4. 3 Sequence Mengelola data user | 27 |
| Gambar 4. 4 Activity Diagram Melihat Hasil Perhitungan..... | 29 |
| Gambar 4. 5 Class Diagram | 30 |
| Gambar 4. 6 Entity Relational Diagram..... | 30 |
| Gambar 4. 7 Tambah Point Neighbor | 33 |
| Gambar 4. 8 Edit Point Neighbor..... | 34 |
| Gambar 5. 1 Halaman Login..... | 35 |
| Gambar 5. 2 Halaman Dashboard | 36 |
| Gambar 5. 3 Halaman <i>Router</i> Location | 36 |
| Gambar 5. 4 Tabel <i>Router</i> Location..... | 37 |
| Gambar 5. 5 Halaman Point Neighbor..... | 37 |
| Gambar 5. 6 Tabel Point Neighbor | 38 |
| Gambar 5. 7 Tampilan halaman analysis | 38 |
| Gambar 5. 8 Tampilan halaman analysis 2 | 39 |
| Gambar 5. 9 Tampilan halaman analysis 3 | 39 |
| Gambar 5. 10 Tampilan halaman analysis 4 | 40 |
| Gambar 5. 11 Halaman User Management | 40 |
| Gambar 5. 12 Proses Tahap 5 | 45 |

DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Perbandingan Penelitian..... | 17 |
| Tabel 4.1 Definisi Aktor | 20 |
| Tabel 4.2 Definisi Use Case..... | 21 |
| Tabel 4.3 Skenario Use Case mengelola data Point Neighbor | 22 |
| Tabel 4. 4 Test Case Point Neighbor | 33 |
| Tabel 4.5 Test Case Function Edit Point Neighbor..... | 34 |
| Tabel 4.6 Pengujian Black Box Router Location..... | 35 |
| Tabel 4.7 Pengujian Black Box Analysis..... | 35 |
| Tabel 5. 1 Tahapan Pertama..... | 41 |
| Tabel 5. 2 Tahapan Kedua | 42 |
| Tabel 5. 3 Tahapan ketiga | 43 |



DAFTAR LAMPIRAN

| | |
|---|----|
| LAMPIRAN A. <i>USE CASE</i> SKENARIO | 47 |
| Tabel A. 1 Skenario Login Super Admin..... | 52 |
| Tabel A. 2 Skenario Login Admin Graf..... | 53 |
| Tabel A. 3 Skenario Login Admin Router | 53 |
| Tabel A. 4 Skenario Pengelolaan Data User | 54 |
| Tabel A. 5 Skenario Pengelolaan Data Router Super Admin | 57 |
| Tabel A. 6 Skenario Pengelolaan Data Router Admin Router..... | 60 |
| Tabel A. 7 Skenario Melihat Data Router Admin Graf | 63 |
| Tabel A. 8 Skenario Pengelolaan data Point Neighbor Super Admin | 64 |
| Tabel A. 9 Skenario Pengelolaan data Point Neighbor Admin Graf | 66 |
| Tabel A. 10 Skenario Hasil Perhitungan Super Admin | 68 |
| Tabel A. 11 Skenario Hasil Perhitungan Admin Graf | 70 |
| Tabel A. 12 Skenario Hasil Perhitungan Admin Router..... | 71 |
| LAMPIRAN B. <i>ACTIVITY DIAGRAM</i> | 68 |
| Gambar B. 1 Login Super Admin | 73 |
| Gambar B. 2 Login Admin Graf | 74 |
| Gambar B. 3 Login Admin Router..... | 74 |
| Gambar B. 4 Pengelolaan Data User..... | 75 |
| Gambar B. 5 Mengelola Data Router..... | 76 |
| Gambar B. 6 Mengelola Data Router admin Graf | 77 |
| Gambar B. 7 Melihat Data Router Admin Graf | 78 |
| Gambar B. 8 Pengelolaan Point Neighbor Super Admin..... | 79 |
| Gambar B. 9 Pengelolaan Point Neighbor Admin Router | 80 |
| Gambar B. 10 Melihat Hasil Perhitungan Super Admin | 81 |
| Gambar B. 11 Melihat Hasil Perhitungan Admin Graf..... | 82 |
| Gambar B. 12 Melihat Hasil Perhitungan Admin Router | 83 |
| LAMPIRAN C. <i>SEQUANCE DIAGRAM</i> | 79 |
| Gambar C. 1 Login Super Admin | 84 |

| | |
|--|-----|
| Gambar C. 2 Login Admin Graf | 84 |
| Gambar C. 3 Login Admin Router..... | 85 |
| Gambar C. 4 Pengelolaan Data User..... | 86 |
| Gambar C. 5 Pengelolaan Data Router Super Admin..... | 87 |
| Gambar C. 6 Pengelolaan Data Router Admin Router | 88 |
| Gambar C. 7 Melihat Data Router | 89 |
| Gambar C. 8 Pengelolaan Point Neighbor Super Admin..... | 90 |
| Gambar C. 9 Pengelolaan Point Neighbor Admin Graf..... | 91 |
| Gambar C. 10 Melihat Hasil Perhitungan Super Admin | 92 |
| Gambar C. 11 Melihat Hasil Perhitungan Admin Graf..... | 92 |
| Gambar C. 12 Melihat Hasil Perhitungan Admin Router | 93 |
| LAMPIRAN D. <i>Kode Program</i> | 89 |
| Gambar D. 1 clogin..... | 93 |
| Gambar D. 2 mlogin..... | 94 |
| Gambar D. 3 Pengelolaan data Router clocation | 95 |
| Gambar D. 4 pengelolaan data Router clocation | 95 |
| Gambar D. 5 pengelolaan data Router mlocation | 96 |
| Gambar D. 6 pengelolaan data point neighbor cAdjacency..... | 97 |
| Gambar D. 7 pengelolaan data point neighbor cAdjacency..... | 97 |
| Gambar D. 8 pengelolaan data point neighbor mAdjacency | 98 |
| Gambar D. 9 Melihat hasil perhitungan cmetode | 99 |
| Gambar D. 10 melihat hasil perhitungan cmetode..... | 99 |
| Gambar D. 11 Melihat hasil perhitungan mmetode | 100 |
| Gambar D. 12 pengelolaan data user clogin | 100 |
| Gambar D. 13 Pengelolaan data user mlogin..... | 101 |
| Gambar D. 14 Melihat data Router | 101 |
| Gambar D. 15 Melihat data Router | 102 |

BAB. 1 PENDAHULUAN

Bab ini merupakan langkah awal dari penulisan tugas akhir ini. Bab ini berisi latar belakang, rumusan masalah, tujuan, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Internet merupakan suatu jaringan komputer raksasa yang saling terhubung dan dapat berinteraksi. Hal ini terjadi karena adanya perkembangan teknologi jaringan yang sangat cepat, sehingga dalam beberapa tahun saja jumlah pengguna komputer yang tergabung dalam internet telah berlipat ganda (Hasan dkk, 2016). Karena itu jaringan komputer bukan suatu yang baru saat ini dan hampir di setiap perusahaan, instansi, maupun perguruan tinggi terdapat jaringan komputer.

Pemanfaatan teknologi jaringan komputer sebagai media komunikasi data hingga saat ini semakin meningkat. Kebutuhan atas penggunaan bersama sumber daya yang ada dalam jaringan baik perangkat lunak maupun perangkat keras telah mengakibatkan timbulnya berbagai pengembangan teknologi jaringan itu sendiri. Seiring dengan semakin tingginya tingkat kebutuhan dan semakin banyaknya pengguna jaringan yang menginginkan suatu bentuk jaringan yang dapat memberikan hasil maksimal baik dari segi efisiensi maupun peningkatan keamanan jaringan itu sendiri. Salah satunya yaitu *hotspot* yang banyak digunakan saat ini. Karena pemakaiannya yang mudah dan tidak begitu membutuhkan banyak biaya dalam penggunaan media atau perangkatnya. *Hotspot* tidak lagi membutuhkan kabel terlalu banyak untuk dapat sharing data. Sebab *hotspot* mengandalkan media transmisi *wireless* (nirkabel atau tanpa kabel) yang menggunakan sinyal (putra, 2013). Sehingga pada saat ini hotspot banyak digemari oleh semua kalangan. *Hotspot* sendiri membutuhkan *hardware* yang bisa mendukung data sharing tanpa menggunakan kabel, yaitu *Router*. *Router* adalah sebuah alat yang mengirimkan paket data melalui sebuah jaringan atau Internet menuju tujuannya, melalui sebuah proses yang dikenal sebagai routing. *Router* dapat digunakan untuk menghubungkan banyak jaringan kecil ke sebuah jaringan yang lebih besar, yang

disebut dengan *Internetwork*, atau untuk membagi sebuah jaringan besar ke dalam beberapa *subnetwork* untuk meningkatkan kinerja dan juga mempermudah manajemennya.

Letak *Router* pada jaringan komputer di Universitas Jember membutuhkan strategi yang tepat untuk menentukan letak yang efektif dalam menempatkan *Router*. Dalam penentuan letak *Router* harus dilakukan dengan optimal, dengan meletakkan pada tempat yang strategis agar dapat meminimumkan jumlah *Router* yang ada dengan Teori Graf. Dalam hal ini lokasi Universitas Jember di representasikan kedalam bentuk graf dengan letak *Router* sebagai titiknya. Graf tersebut kemudian dianalisis menggunakan *Dominating Set* untuk mengetahui letak dan jumlah *Router* yang akan dipasang. *Router* digambarkan sebagai titik dan jarak antar *Router* sebagai garis.

Dominating Set diterapkan pada pencarian titik lokasi *Router* di Universitas Jember untuk mengetahui *Router* mana yang mendominasi *Router* lain. Pencarian titik pendominasi menggunakan algoritma *Greedy*. Algoritma *Greedy* adalah salah satu algoritma yang sering dipakai dalam ilmu matematika dan komputer. Algoritma *Greedy* merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya. Nilai maksimum sementara ini dikenal dengan istilah *local maximum*. Algoritma *Greedy* biasanya memberikan solusi yang mendekati nilai optimum dalam waktu yang cukup cepat.

Penelitian ini bertujuan untuk optimasi dalam penentuan lokasi *Router* yang berada pada Universitas Jember dengan menggunakan *Dominating Set*. Algoritma pencarian yang digunakan untuk memecahkan masalah tersebut adalah algoritma *Greedy*.

1.2 Rumusan Masalah

Berdasarkan permasalahan yang telah dijabarkan, maka dapat dirumuskan masalah dalam penelitian ini yaitu sebagai berikut:

1. Bagaimana pengimplementasikan algoritma *Greedy* untuk menentukan *Dominating Set* pada lokasi penempatan *Router* Universitas Jember?

2. Bagaimana merancang dan membangun sebuah sistem informasi yang mampu menentukan lokasi *Router* Universitas Jember?

1.3 Tujuan

Berikut merupakan tujuan yang ingin dicapai dalam penelitian ini.

1. Mengimplementasikan algoritma *Greedy* untuk penentuan *Dominating Set* lokasi *Router* Universitas Jember.
2. Merancang dan membangun sebuah sistem informasi yang mampu menentukan lokasi penempatan *Router* Universitas Jember.

1.4 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Hasil penelitian dapat digunakan sebagai cara untuk memberikan masukan untuk penempatan lokasi *Router* Universitas Jember.
2. Hasil penelitian ini dapat digunakan sebagai dasar bagi penelitian yang selanjutnya untuk pengembangan sistem informasi yang lebih baik.

1.5 Batasan Masalah

Beberapa hal yang membatasi penelitian ini adalah:

1. Sistem informasi dibangun berbasis *Website*.
2. Sistem informasi ini hanya meliputi penempatan *Router* pada bangunan lantai satu
3. Sistem informasi ini digunakan untuk menentukan *Dominating Set* pada graf *Router* Universitas Jember.
4. Proses Penentuan menggunakan metode *Dominating Set* serta Algoritma *Greedy*.

1.6 Sistematika Penulisan

Sistematika penulisan dalam penyusunan tugas akhir ini adalah sebagai berikut

1. Pendahuluan
Bab ini terdiri atas latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika penulisan.
2. Tinjauan Pustaka

Bab ini berisi tentang kajian materi dan informasi apa saja yang digunakan dalam penelitian ini..

3. Metodologi Penelitian

Bab ini berisi menguraikan tentang metodologi penelitian apa yang dilakukan selama penelitian. Dimulai dari tahap pencarian masalah hingga pengujian sistem analisis kelayakan investasi yang akan dibuat.

4. Analisa Kebutuhan dan Perancangan Sistem

Bab ini menganalisa proses bisnis setelah sistem informasi diimplementasikan. Kemudian menguraikan tentang perancangan desain sistem dan pengembangan sistem secara keseluruhan.

5. Hasil dan Pembahasan

Bab ini menjelaskan hasil dan pembahasan dari penelitian yang telah dilakukan. Dengan menampilkan hasil penelitian dan hasil percobaan perhitungan sistem.

6. Penutup

Bab ini berisi kesimpulan dari penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB. 2 TINJAUAN PUSTAKA

Bab ini berisi pembahasan mengenai tinjauan pustaka, definisi, serta penjelasan mengenai teori-teori yang berhubungan dengan permasalahan yang akan dibahas dalam penelitian.

2.1 Penelitian terdahulu

Penelitian terdahulu yang dilakukan dengan menerapkan teori *Dominating Set* yaitu dengan judul “Penerapan Teori Dominating Set dalam Instalasi Client Hub Untuk Jaringan Intranet di Universitas Jember” dilakukan oleh Ika Hesti Agustin S.Si., M.Si dari Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Jember pada tahun 2014. Pada penelitian tersebut dijelaskan bagaimana cara meminimumkan jumlah Hub dalam sebuah jaringan Intranet di Universitas Jember menggunakan pendekatan matematis Teori Graf *Dominating Set*. Penelitian tersebut menghasilkan beberapa kemungkinan untuk instalasi meletakkan *Client Hub* untuk jaringan intranet. Penelitian tersebut juga menghasilkan beberapa teorema dalam menentukan Domination Number pada graf-graf khusus (Agustin, 2014). Dari Penelitian dengan judul Penerapan Teori Dominating Set Dalam Instalasi Client Hub Untuk Jaringan Intranet Di Universitas Jember bisa di jadikan referensi dalam penelitian yang sudah dilakukan, karena dari penelitian tersebut bertujuan untuk menentukan titik mana yang harus di didirikan *Client Hub* Intranet dengan metode *Dominating Set*. Penelitian yang telah dilakukan menggunakan metode *Dominating Set* dengan algoritma *Greedy*. Penggunaan algoritma tersebut diharapkan menghasilkan penentuan lokasi yang lebih baik, karena algoritma *Greedy* memberikan pilihan terbaik yang dapat diperoleh saat itu juga.

Penelitian selanjutnya berjudul Implementasi *Dominating Set* untuk Sistem Penentuan Letak Panel Surya Sebagai Sumber Energi Terbarukan Pada Jaringan Lampu Lalu Lintas. Penelitian ini bertujuan untuk membuat perancangan dan pembangunan sistem informasi penentu letak panel surya sebagai sumber energi terbarukan pada jaringan lampu lalu lintas. Penelitian tersebut teori graf digunakan sebagai pendekatan matematis yang bisa digunakan untuk meminimumkan jumlah

panel surya. Penentuan titik pendominasi menggunakan Algoritma *Greedy*. Algoritma *Greedy* akan mencari penyelesaian pertahap sampai menghasilkan solusi optimum. Hasil yang diperoleh dari *algoritma Greedy* digunakan sebagai inputan *Dominating Set* dan kemudian dihitung nilai keoptimalannya berdasarkan perhitungan *Dominating Set*. Hasil dari penelitian ini adalah beberapa kemungkinan untuk penentuan letak panel surya. Hasil tersebut dapat digunakan sebagai bahan pertimbangan dalam penentuan letak panel surya pada jaringan lampu lalu lintas di wilayah kota Banyuwangi (Fambudii, 2017). Penelitian dengan judul Implementasi *Dominating Set* untuk Sistem Penentuan Letak Panel Surya Sebagai Sumber Energi Terbarukan Pada Jaringan Lampu Lalu Lintas sangat membantu pihak yang berwenang untuk mengurangi kesalahan dalam penempatan panel surya yang akan didirikan pada wilayah kota Banyuwangi dan juga membuat pengeluaran biaya untuk mendirikan panel surya menjadi efisien.

Penelitian selanjutnya berjudul Sistem Informasi Penentuan Tata Letak Mobil Listrik Menggunakan Himpunan Dominasi Pada Graf (Simulasi Agro Techno Park Jubung Universitas Jember). Penelitian tersebut bertujuan untuk penentuan tata letak mobil listrik yang tepat agar dapat meminimumkan jumlah mobil listrik yang digunakan dan menempatkan dengan strategis. Pendekatan matematis yang digunakan pada penelitian tersebut adalah dengan Teori Graf untuk meminimumkan mobil listrik yang dibutuhkan serta menentukan letak dengan strategis dengan Teori Himpunan Dominasi. Sistem tersebut menggunakan himpunan dominasi pada graf dalam pencarian titik kandidatnya dengan menggunakan *Algoritma Greedy*, adapun beberapa tahapan yang dilakukan. Tahap pertama mengurutkan *vertex* berdasarkan jumlah *edge* terbanyak, selanjutnya memilih *vertex* dari jumlah derajat yang tinggi, tahapan ketiga memanggil *vertex* sisa yang belum menemukan himpunan dominasi serta melihat sisa derajat dari *vertex*, tahap keempat memanggil *vertex* sisa dan menentukan himpunan dominasi pada *vertex* sisa, tahapan yang terakhir menggabungkan semua himpunan dominasi dimana semua titik sudah menemukan himpunan dominasi (Levitasari, 2016). Penelitian dengan judul Sistem Informasi Penentuan Tata Letak Mobil Listrik

Menggunakan Himpunan Dominasi Pada Graf (Simulasi Agro Techno Park Jubung Universitas Jember). Penelitian ini dapat memberikan masukan serta terapan metode yang dapat digunakan dalam penelitian yang sudah dilakukan, karena dalam penelitian tersebut sama dalam penggunaan metode yaitu *Dominating Set* dan algoritma *Greedy* dan dimana lokasi pada penelitian ini dilakukan di Agro Techno Park Jubung Universitas Jember yang luas lokasinya tidak terlalu luas sama dengan lokasi yang sudah dilakukan pada Universitas Jember, diharapkan menghasilkan hasil yang optimal.

Penelitian selanjutnya berjudul Implementasi Algoritma *Greedy* Dalam Menentukan Rute Trayek Angkutan Kota (Angkot) Tanjung Pinang. Penelitian tersebut dilakukan oleh (Singgih Dkk, 2014). Jurusan Informatika, Fakultas Teknik, Universitas Maritim Raja Ali Haji (UMRAH), menjelaskan bahwa kurang optimalnya pembagian jalur trayek di Kota Tanjung Pinang tentunya sangat menyulitkan masyarakat untuk sampai ke tujuan. Jalur trayek yang ada sekarang bersifat random atau keliling, semua penumpang mempunyai kemungkinan untuk naik angkot karena angkot tersebut tidak mempunyai trayek yang pasti. Untuk masalah yang sedang terjadi, maka penulis mencoba menerapkan algoritma *Greedy* untuk menyelesaikan permasalahan itu. Algoritma *Greedy* adalah berupa metode yang paling populer untuk memecahkan masalah persoalan optimasi. Persoalan Optimasi itu berupa persoalan mencari solusi optimum. Penelitian tersebut bertujuan menentukan jalur trayek yang lebih optimal di Kota Tanjung Pinang menggunakan algoritma *Greedy*, sehingga dapat lebih membantu pengguna dalam menentukan jalur trayeknya. Penggunaan metode algoritma *Greedy* akan membantu memberikan solusi rute trayek serta memberikan nilai perbedaan jarak tempuh pada semua rute trayeknya, itu akan memberikan solusi terhadap penentuan rute trayek angkutan kota (Angkot) yang akan di lewati oleh supir angkot. Penelitian dengan judul berjudul Implementasi Algoritma *Greedy* Dalam Menentukan Rute Trayek Angkutan Kota (Angkot) Tanjung Pinang, dapat memberikan masukan serta terapan metode yang dapat digunakan dalam penelitian yang sudah dilaksanakan, karena dalam penelitian tersebut sama dalam penggunaan metode yaitu *Dominating Set* dan algoritma *Greedy*. Tetapi dalam penelitian ini

bertujuan untuk memberikan solusi terhadap penentuan rute trayek angkutan kota berbeda dengan penelitian yang sudah dilaksanakan, penelitian yang akan datang lebih bertujuan dalam lokasi penempatan lokasi *Router* pada Universitas Jember.

Penelitian selanjutnya berjudul Penggunaan Algoritma *Greedy* Dalam Penentuan Rute Wisata yang dilakukan oleh Renusa Andra Prayogo bertujuan untuk dapat menemukan solusi permasalahan penentuan rute wisata, dan menentukan apakah dengan algoritma tersebut dapat menghasilkan rute yang tidak terlalu jauh dari solusi optimal. Tujuan dari algoritma *Greedy* pada permasalahan ini adalah untuk menentukan rute yang terbaik dari suatu titik ke titik lain dengan mempertimbangkan jarak tempuh dan tempat wisata yang dikunjungi. Pada algoritma ini, untuk mendapatkan solusi akan dilakukan dua kali perhitungan *greedy*, yaitu *Greedy* berdasarkan jarak terdekat, dan *Greedy* berdasarkan tempat wisata. Pada perhitungan *Greedy* berdasarkan tempat wisata, selain diperhitungkan jaraknya juga diperhitungkan juga “ranking” dari tempat wisata tersebut, sehingga setiap tempat wisata memiliki “bobot” masing-masing yang akan digunakan pada fungsi seleksi pada algoritma *Greedy*. Solusi umum dari *Greedy* ini akan ditentukan dari hasil kedua algoritma tersebut (Prayogo, 2014). Penelitian dengan judul Penggunaan Algoritma *Greedy* Dalam Penentuan Rute Wisata, dapat memberikan masukan serta terapan metode yang dapat digunakan dalam penelitian selanjutnya, karena dalam penelitian tersebut sama dalam penggunaan metode yaitu *Dominating Set* dan algoritma *Greedy*. Sama dengan penelitian yang mengangkat topik tentang rute angkot, penelitian ini berfokus pada pencarian rute untuk menuju tempat wisata yang berada di beberapa kota. Berdasarkan penelitian tersebut dapat memberikan masukan serta terapan metode yang dapat digunakan dalam penelitian yang akan datang. Menggunakan metode *Dominating Set* dan Algoritma *Greedy* dapat memberikan solusi optimum untuk penempatan lokasi *Router* pada Universitas Jember. Perbedaan dan persamaan penelitian terdahulu dengan penelitian yang telah dilakukan seperti ditunjukkan pada Tabel 2.1

Tabel 2.1 Perbandingan Penelitian

| No | Peneliti | Objek Penelitian | Metode | Variabel | Tujuan |
|----|-----------------------|--|--|---|---|
| 1. | Agustin, 2014 | Instalasi Client Hub Untuk Jaringan Intranet Di Universitas jember | <i>Dominating Set</i> | jumlah user yang akses dalam waktu bersamaan dan jarak antar ruangan yang satu dengan ruangan yang lain. | Meminimalkan jumlah Hub dalam sebuah Jaringan intranet dengan menggunakan teori Dominating Set dan mengembangkan teori Dominating set |
| 2. | Fambudi, 2017 | Penentuan Letak Panel Surya Di Kota Banyuwangi | <i>Dominating Set (Algoritma Greedy)</i> | Jarak antar lampu lalu lintas di persimpangan jalan | Meminimalkan panel surya yang akan ditempatkan pada jaringan lampu lalu lintas |
| 3. | Levitasari, 2016 | Penentuan tata letak mobil listrik(Simulasi Agrotechno park jubung Universitas Jember) | <i>Dominating Set (Algoritma Greedy)</i> | Jalan Sebagai titik, dan persimpangan sebagai sisi | Meminimalkan mobil listrik yang di tempatkan di agrotechnopark Universitas Jember |
| 4. | Bettiza, Suswaini, | Penentuan Rute Trayek Angkutan Kota(Angkot) Tanjung Pinang | Algoritma <i>Greedy</i> | Titik keramaian, dan Jarak untuk setiap titik keramaian. | Menentukan jalur trayek angkutan kota yang lebih optimal |
| 5. | Prayogo, 2014 | Penggunaan Algoritma <i>Greedy</i> Dalam Penentuan Rute Wisata | Algoritma <i>Greedy</i> | Kota-kota yang berhubungan, jarak dari kota tersebut ke kota yang bersebelahan, daftar objek wisata, Rating dari objek wisata, nama objek wisata, Status kota memiliki objek wisata atau tidak. | Menemukan solusi algoritma untuk permasalahan penentuan rute wisata dan rute yang tidak terlalu jauh dari solusi optimal |
| 6. | Hardja, 2018 | Penentuan Lokasi <i>Router</i> Pada Universitas Jember | <i>Dominating Set(Algoritma Greedy)</i> | Jarak antar bangunan yang akan bangun <i>Router</i> | Meminimalkan jumlah router yang akan ditempatkan pada Universitas Jember |

2.2 Dominating Set

Himpunan pendominasi ialah suatu himpunan bagian V' dari himpunan titik (G) dimana titik-titik yang tidak berada pada V' terhubung langsung dengan minimal satu titik pada V' . Ukuran terkecil dari himpunan pendominasi disebut bilangan dominasi (*domination number*) dilambangkan dengan $\gamma(G)$ (Alanko, Dkk, 2011).

(Haynes, 1998) menyatakan mengenai batas atas dari domination number adalah banyaknya titik di graf. Ketika paling sedikit satu titik yang dibutuhkan untuk *domination* di graf, maka $1 \leq \gamma(G) \leq n$ untuk setiap graf yang berordo n .

Persamaan 1 merupakan teorema yang diambil dari Miftahur R, et.al: dalam “Kajian Himpunan Dominasi pada Graf” untuk mengetahui keoptimalan dalam menentukan dominating set pada sembarang graf.

$$\left\lceil \frac{p}{1+\Delta(G)} \right\rceil \leq \gamma(G) \leq p - \Delta(G) \quad \dots(1)$$

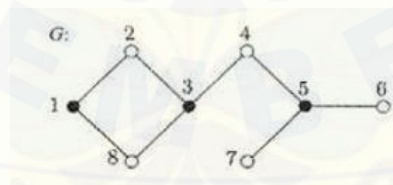
Keterangan:

p : jumlah titik-titik yang ada pada sebarang graf

$\Delta(G)$: derajat terbesar pada sebarang graf

$\gamma(G)$: jumlah perkiraan *Dominating Set*

Contoh sebuah Graf G dengan himpunan pendominasinya dapat dilihat di Gambar



Gambar 2. 1 Graf G dengan Pendominasinya (1,3,5) (Iswadi, 2011)

2.3 Algoritma Greedy

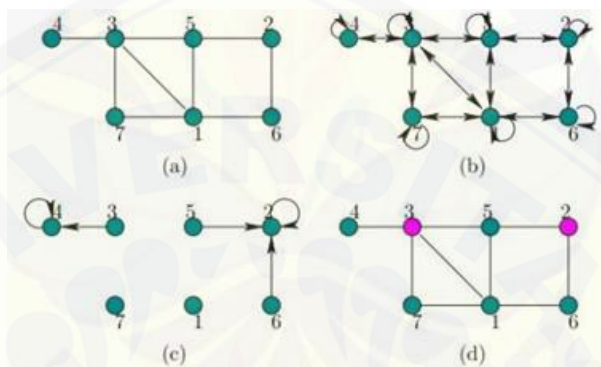
Algoritma *Greedy* merupakan jenis algoritma yang menggunakan pendekatan penyelesaian masalah dengan mencari nilai maksimum sementara pada setiap langkahnya. Nilai maksimum sementara ini dikenal dengan istilah local maximum. Pada kebanyakan kasus, algoritma *Greedy* tidak akan menghasilkan solusi paling optimal, begitupun algoritma *Greedy* biasanya memberikan solusi yang mendekati nilai optimum dalam waktu yang cukup cepat. Karena sifat *Greedy* tersebut, maka algoritma *Greedy* sering kali dianggap sebagai algoritma berkarakteristik “*short sight*” dan “*non-recoverable*”. Oleh karena itu pula, algoritma *Greedy* baiknya digunakan pada permasalahan yang tidak mementingkan solusi optimum, dan cocok untuk masalah-masalah sederhana (Prayogo, 2014).

Komponen algoritma *Greedy* terdiri dari (ardiyansyah, et al., 2010):

- a. Himpunan Kandidat C
Merupakan himpunan yang berisi elemen pembentuk solusi.
- b. Himpunan Solusi
Himpunan yang berisi elemen solusi pemecahan masalah.
- c. Fungsi Seleksi
Fungsi yang memilih kandidat yang paling memungkinkan dari himpunan kandidat untuk dimasukkan ke dalam himpunan solusi agar solusi optimal terbentuk. Kandidat yang sudah terpilih pada suatu langkah tidak akan dipertimbangkan pada langkah selanjutnya.
- d. Fungsi Kelayakan
Fungsi yang memeriksa apakah suatu kandidat yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut, bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.
- e. Fungsi Obyektif
Fungsi yang memaksimalkan atau meminimalkan nilai solusi.

Algoritma *Greedy* sederhana digunakan untuk menemukan dominating set yang seminimal mungkin dari graf tidak berarah G dengan p titik dan q sisi. Dapat ditunjukkan dengan batas atas $\gamma(G) \leq p + 1 - \sqrt{2q + 1}$, dimana $\gamma(G)$ adalah

kardinaliti dari dominating set yang ditemukan dari algoritma *Greedy*. Misalkan $V = \{1,2,3, \dots, p\}$, dan mendefinisikan $S = \emptyset$. *Greedy* menambahkan sisi ke S di setiap langkah, sampai D merupakan sebuah *dominating set*. Sebuah titik j dikatakan tercakup jika $j \in S$ atau sebarang tetangga dari titik j ada di S maka titik yang tidak dicover dikatakan tidak tercakup (Alfarisi, 2014).



Gambar 2.1 Algoritma Greedy (Alfarisi, 2016)

BAB. 3 METODOLOGI PENELITIAN

Bab ini menjelaskan mengenai tempat dan waktu penelitian, tahap penelitian, proses pengumpulan data, serta metode yang digunakan pada penelitian ini untuk menyelesaikan pembuatan sistem Penentuan Lokasi *Router* Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode *Dominating Set*.

3.1. Tempat dan Waktu Penelitian

Penelitian dilakukan di Kampus Universitas Jember. Waktu penelitian dilakukan selama 3 bulan dimulai pada bulan Januari 2018 sampai Maret 2018.

3.2. Tahap Pengumpulan Data

Tahap ini melakukan teknik pengumpulan data yang dibutuhkan dalam penelitian ini, dilakukan aktifitas sebagai berikut:

3.2.1 Studi Literatur

Tahap ini akan dilakukan studi literatur untuk menambah wawasan dalam melakukan penelitian. Hal ini dilakukan untuk memperoleh landasan teori dan untuk menentukan variable apa saja yang diukur kemudian menganalisis penelitian sebelumnya yang memiliki hubungan dengan penelitian ini.

3.2.2 Studi Lapangan

Penelitian ini studi lapang dilakukan di wilayah Universitas Jember. Studi lapang ini dilakukan untuk mengumpulkan data-data yang sesuai dengan keadaan lapang, mulai dari letak *Router* sampai jarak antar *Router*. Studi lapang juga dilakukan untuk melakukan *tracking* lahan Universitas Jember.

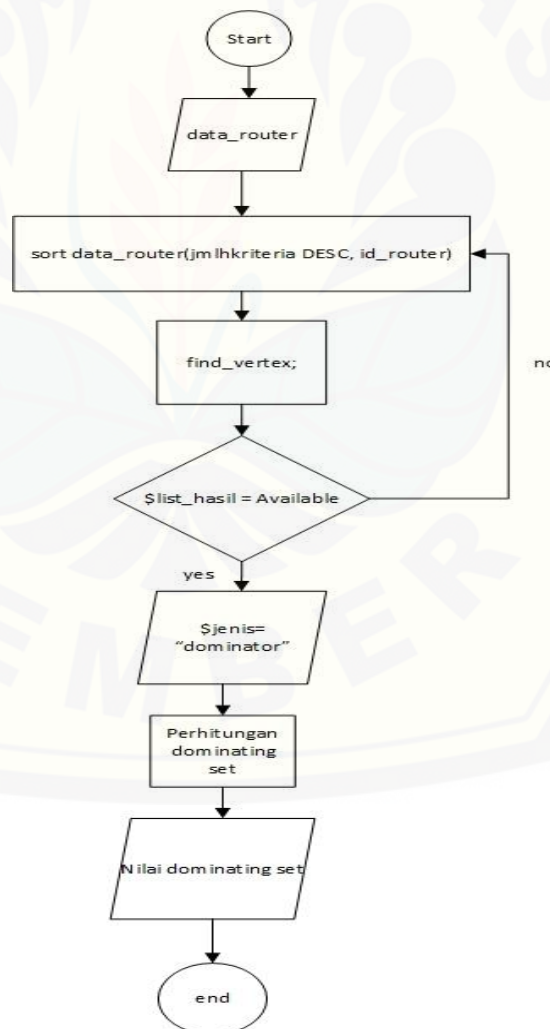
3.2.3 Wawancara

Wawancara adalah satu dari beberapa cara untuk mengumpulkan informasi, wawancara ini digunakan untuk mengumpulkan informasi mengenai *Router* Universitas Jember. Wawancara dilakukan secara langsung dengan petugas jaringan Universitas Jember. Wawancara ini menghasilkan data lokasi *Router*, data

lokasi perangkat *Unifi*, serta data pendukung lainnya yang berada di Universitas Jember tepatnya berada pada UPT Teknologi dan Komunikasi Universitas Jember.

3.3 Tahap Analisis Data

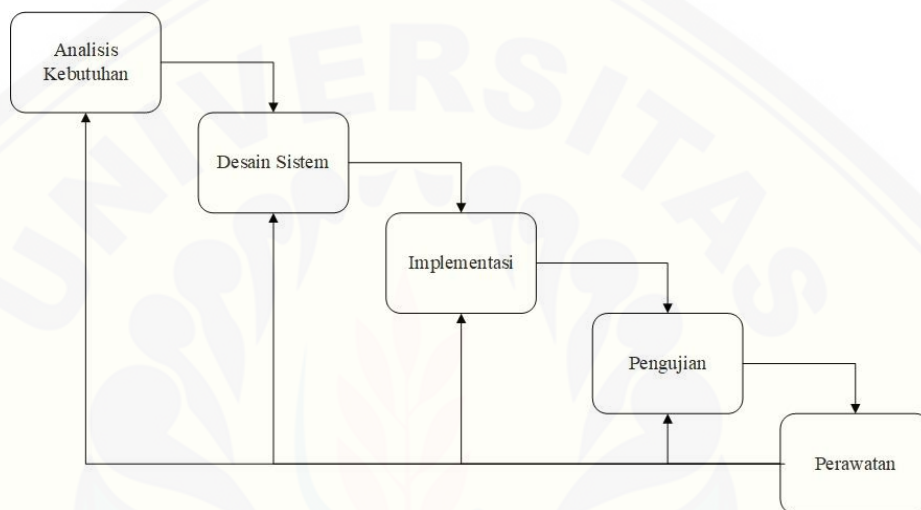
Tahap analisis data dimulai dengan melakukan analisis dari data yang telah didapatkan. Data yang digunakan dalam penelitian ini adalah data lokasi *Router* yang akan dibangun di Universitas Jember dan peta wilayah Universitas Jember yang diperoleh dari proses *Tracking* menggunakan aplikasi *Geo Tracker*. Hasil *tracking* kemudian diolah menggunakan *GPS Track Editor*. Dari data tersebut kemudian dilakukan perhitungan secara matematis menggunakan *Dominating Set* dan algoritma *Greedy*. Diagram alir dari sistem informasi penentu lokasi *Router* dapat dilihat pada Gambar 3.1



Gambar 3.1 Diagram Alir Proses Perhitungan Algoritma *Greedy* dan *Dominating set*

3.4 Tahap Pengembangan Sistem

Teknik pengembangan penelitian sistem ini menggunakan model *Waterfall*. Proses pengembangan dijalankan bertahap dari tahap sebelumnya dan berjalan berurutan. Secara umum tahapan-tahapan di dalam model *waterfall* terbagi atas 5 tahapan yaitu: analisis kebutuhan, desain sistem, implementasi, pengujian dan perawatan yang tercakup pada diagram alir gambar 3.2



Gambar 3. 2 Model Waterfall (Sommerville, 2011)

3.4.1 Analisis Kebutuhan Sistem Analisis Kelayakan Investasi

Tahapan analisis kebutuhan merupakan tahapan pertama dalam model pengembangan *waterfall*, dimana pada tahap ini adalah untuk menentukan permasalahan inti dari penelitian. Metode perhitungan menggunakan *Dominating Set*.

3.4.2 Desain sistem analisis kelayakan investasi

Tahap desain dalam model *waterfall* merupakan fase yang berfokus pada struktur desain arsitektur perangkat lunak yang akan dibangun dan detail algoritma yang digunakan. Nantinya pada tahapan ini akan menghasilkan sebuah dokumen yang disebut *software requirement*. Untuk perancangan sistem disini menggunakan konsep perancangan berorientasi objek, dimana dalam tahapan desainnya menggunakan UML (*Unified Modelling Language*) diantaranya yaitu :

1. *Business Process*

2. *Use Case Diagram*
3. *Scenario*
4. *Activity Diagram*
5. *Sequence Diagram*
6. *Class Diagram*
7. *Entity Relationship Diagram*

3.4.3 Implementasi

Pada tahap ini seluruh rangkaian desain yang sudah dibuat akan diimplementasikan ke dalam pembuatan sistem dalam bentuk baris pengkodean program. Sistem yang akan dibangun disini berbasis *website* sehingga bahasa pemrograman yang digunakan adalah *HTML (Hyper Text Markup Language)*, *CSS (Cascading Style Sheets)* dan *PHP (Hypertext Preprocessor)*. Untuk manajemen basis data dalam pengembangan sistem ini menggunakan *DBMS My SQL*.

3.4.4 Testing

Pada tahapan ini implementasi yang sudah dilakukan dalam pembangunan sistem akan diuji apakah sistem yang dibangun dapat berjalan dengan baik sesuai dengan yang diharapkan atau tidak. pada tahapan ini juga digunakan untuk mengetahui kegunaan dari sistem yang dibangun. Untuk pengujian aplikasi ini menggunakan.

1. *White Box*

White Box Testing adalah salah satu cara untuk menguji suatu aplikasi atau software dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak. Kalau modul yang telah dan sudah di hasilkan berupa *output* yang tidak sesuai dengan yang di harapkan maka akan di *compile* ulang dan di cek kembali kode-kode tersebut hingga mencapai sesuai dengan yang di harapkan.

2. *Black Box*

Black box digunakan untuk menemukan kesalahan dalam beberapa kategori yaitu fungsi tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur

data atau akses database eksternal, kesalahan kinerja atau perilaku, kesalahan inisialisasi dan terminasi (Wahyuningrum & Januarita, 2015).

3.4.6 Perawatan (*Maintenance*)

Ini merupakan tahapan terakhir dari model pengembangan *waterfall*, dimana pada tahap ini sistem yang sudah dibangun diharuskan adanya pemeliharaan yang bertujuan untuk memperbaiki kesalahan yang tidak ditemukan dari tahapan-tahapan yang sebelumnya. Bukan hanya dari sisi perbaikan kesalahan, tujuan lain dari tahapan *maintenance* ini adalah untuk peningkatan kinerja dari sistem yang akan digunakan sebagai kebutuhan baru.

3.5 Gambaran Sistem

Sistem informasi yang dibangun merupakan sistem informasi untuk penentuan *dominating set* atau himpunan pendominasi pada graf dengan studi kasus *Router* di Universitas Jember. Sistem informasi ini berbasis *Web* dan menggunakan *Database MySql*. Sistem Informasi ini dibangun untuk mengetahui titik pendominasi pada graf *Router* di Universitas Jember. Titik yang digunakan adalah lokasi *Router* dan garisnya jarak antar titik *Router*. Aktor yang terlibat dalam sistem ini yaitu super administrator yang bertugas manajemen data dan semua fitur yang terdapat dalam sistem. Administrator graf memiliki tugas dalam pengelolaan titik-titik graf yang dibutuhkan. Kemudian aktor selanjutnya adalah administrator *Router* bertugas menentukan titik lokasi *Router*.

BAB. 4 PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan sistem informasi penentu lokasi *Router* di Universitas Jember. Tahap pengembangan sistem dilaksanakan berdasarkan model *waterfall*, dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, pembuatan desain sistem, penulisan kode program, dan pengujian sistem.

4.1 Analisis Kebutuhan

Analisis kebutuhan sistem merupakan tahapan awal untuk menentukan kebutuhan sistem. Kebutuhan sistem yang diperlukan yaitu kebutuhan fungsional dan non-fungsional.

4.1.1 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem mampu difungsikan sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem itu sendiri. Kebutuhan fungsional dari sistem informasi penentu lokasi *Router* di Universitas Jember yaitu:

1. Sistem dapat mengelola data *User* (*create, read, update, delete*).
2. Sistem dapat mengelola data lokasi *Router* (*create, read, update, delete*).
3. Sistem dapat mengelola data titik tetangga atau *Point Neighbor* (*create, read, update, delete*).
4. Sistem dapat menampilkan hasil penentuan letak *Router* berdasarkan perhitungan *Dominating Set*.

4.1.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan fitur-fitur yang dimiliki untuk mendukung sistem dalam memenuhi fungsionalitasnya untuk dapat memenuhi kebutuhan dari pengguna. Kebutuhan non-fungsional dari sistem informasi penentu *dominating set* pada sistem informasi penentu lokasi *Router* di Universitas Jember yaitu:

1. Sistem memiliki batasan hak akses pengguna dengan menggunakan *Username*

dan *password*.

2. Tampilan dan bahasa komunikasi sistem mudah dimengerti oleh pengguna untuk memberikan kemudahan dalam pengoprasian.
3. Sistem dapat dijalankan dengan *browser* yang berbeda.

4.2 Desain Sistem

Desain sistem yang dibuat meliputi *Business Process*, *Usecase Diagram*, *Scenario*, *Sequence Diagram*, *Activity Diagram*, *Class Diagram*, dan *Entity Relationship Diagram (ERD)*.

4.2.1 Business Process

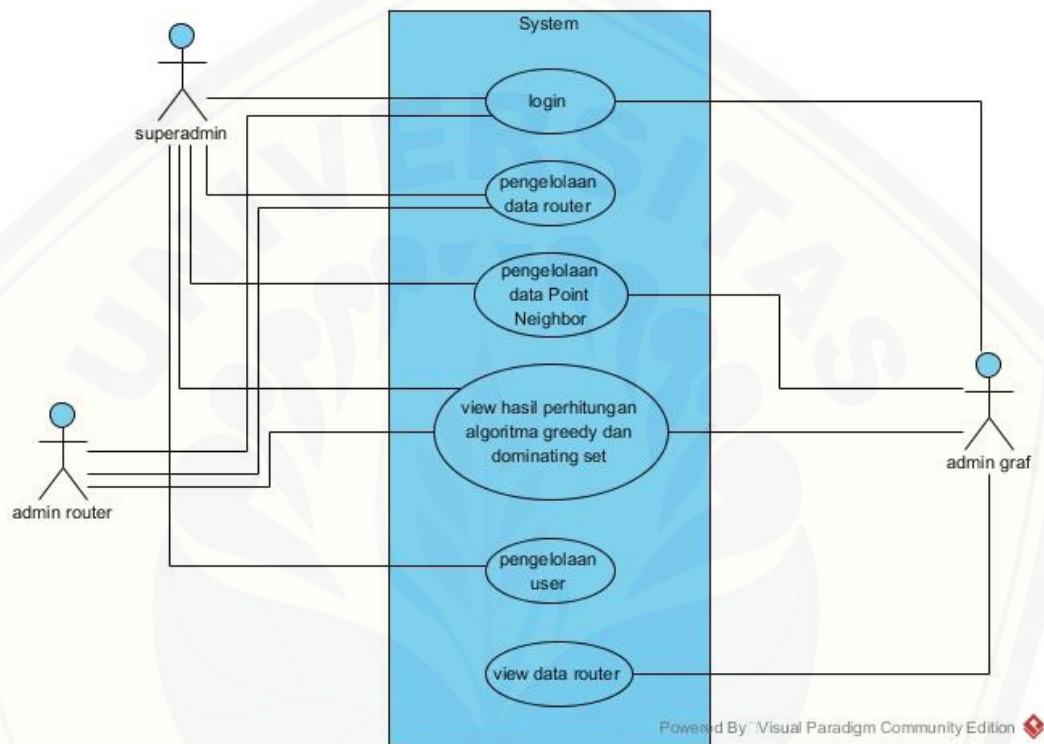
Merupakan kumpulan dari aktivitas yang tersusun dan saling berhubungan sehingga dapat menghasilkan suatu produk. *Business Process* sendiri terdiri dari beberapa bagian, data yang menjadi masukan sistem (*Input*), data masukan akan diolah melalui sistem sehingga menghasilkan data keluaran (*Output*), media yang digunakan (*Uses*), serta tujuan yang ingin dicapai oleh sistem (*Goal*). *Business Process* pada sistem ini dapat dilihat pada gambar 4.1.



Gambar 4.1 *Business Process*

4.2.2 Use Case Diagram

Use Case Diagram merupakan gambaran fungsional dari sistem yang menunjukkan fitur-fitur yang disediakan oleh sistem serta aktor yang dapat mengakses fitur tersebut pada sistem aplikasi penentu *Router*. *Use Case Diagram* pada sistem ini dapat dilihat pada gambar 4.2.



Gambar 4.2 Use Case Diagram

4.2.3 Definisi Aktor

Definisi aktor merupakan aktor-aktor yang dapat mengakses sistem aplikasi penentu *Router*. Terdapat 3 aktor seperti yang dijelaskan pada tabel 4.1.

Tabel 4.1 Definisi Aktor

| No | Aktor | Deskripsi |
|----|-------------|---|
| 1. | Super Admin | Aktor Super Admin pada sistem ini memiliki hak akses penuh dalam sistem diantaranya mengelola data user, data lokasi <i>Router</i> , data <i>point neighbor</i> , dan |

| | | |
|----|---------------------|---|
| | | dapat melihat hasil perhitungan dari sistem. |
| 2. | Admin Graf | Aktor Admin Graf pada sistem ini memiliki hak akses dalam mengelola data <i>Point Neighbor</i> , dapat melihat titik lokasi <i>Router</i> , dan dapat melihat hasil perhitungan dari sistem |
| 3. | Admin <i>Router</i> | Aktor Admin Graf pada sistem ini memiliki hak akses dalam mengelola data lokasi <i>Router</i> , dan dapat melihat hasil perhitungan dari sistem |

4.2.4 Definisi *Use Case*

Definisi *use case* yaitu penjelasan dari masing masing *Use Case* atau fitur-fitur yang terdapat pada penentuan lokasi *Router* untuk optimasi jaringan komputer dengan menggunakan metode *Dominating Set* yang akan dibangun. Terdapat 6 *Use Case* seperti yang dijelaskan pada Tabel 4.2

Tabel 4.2 Definisi *Use Case*

| No | <i>Use Case</i> | Deskripsi |
|----|--------------------------------------|---|
| 1. | <i>Login</i> | Menggambarkan proses autentifikasi yang digunakan untuk masuk ke sistem. |
| 2. | Mengelola Data <i>Router</i> | Menggambarkan proses untuk menambah, mengedit, menghapus serta melihat data lokasi <i>Router</i> |
| 3. | Mengelola Data <i>Point Neighbor</i> | Menggambarkan proses untuk menambah, mengedit, menghapus serta melihat data <i>Point Neighbor</i> . |
| 4. | Melihat Hasil Perhitungan | Menggambarkan proses untuk melihat Hasil Perhitungan Dari Sistem |

| | | |
|----|----------------------------|---|
| 5. | Mengelola Data <i>User</i> | Menggambarkan proses untuk menambah, mengubah, dan melihat data <i>User</i> . |
| 6. | Melihat Data <i>Router</i> | Menggambarkan proses untuk melihat data <i>Router</i> |

4.2.5 Scenario

Scenario digunakan untuk menjelaskan alur sistem sesuai dengan kebutuhan fungsional yang terdapat pada *Use Case Diagram* seperti pada gambar 4.2.

1. Skenario *Login*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario login dijelaskan pada lampiran A.

2. Skenario Mengelola Data *Router*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario Mengelola Data *Router* dijelaskan pada lampiran A.

3. Skenario Mengelola Data *Point Neighbor*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario Mengelola Data *Point Neighbor* dijelaskan pada tabel 4.3.

Tabel 4.3 Skenario *Use Case* mengelola data *Point Neighbor*

| | |
|--------------------------|---|
| Nama Use Case | Pengolahan data <i>Point Neighbor</i> |
| Aktor | Super admin |
| Deskripsi singkat | Super admin akan mengelola data <i>Point Neighbor</i> |
| Prekondisi | Super Admin berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>Point Neighbor</i> pada menu <i>Point neighbor</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>Point Neighbor</i> berhasil ditambah • Data <i>Point Neighbor</i> berhasil dihapus |

| | |
|--|--|
| | <ul style="list-style-type: none"> Data <i>Point Neighbor</i> berhasil diubah |
| Flow event | |
| Skenario Normal : Input <i>Point Neighbor</i> | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>Point Neighbor</i> | |
| | 2. Menampilkan halaman <i>Point Neighbor</i> meliputi peta dengan data <i>Point Neighbor</i> yang telah diinputkan beserta tabel data <i>Point Neighbor</i> dengan atribut <ul style="list-style-type: none"> - Id list - Id Router - Titik tetangga - Action Tombol <i>Edit</i> dan <i>Delete</i> |
| 3. Klik titik pada peta | |
| | 4. Menampilkan informasi detail <i>Point Neighbor</i> |
| 5. Klik <i>Add</i> | |
| | 6. Menampilkan halaman <i>Point Neighbor</i> meliputi peta dan form tambah <i>Point Neighbor</i> Form input <i>Router</i> meliputi: <ul style="list-style-type: none"> - Id Router - Titik tetangga - Latitude - longitude Tombol <i>Save</i> , <i>Reset</i> , |
| 7. Mengisi form <i>Point Neighbor</i> | |
| 8. Klik <i>Save</i> | |
| | 9. Menampilkan halaman <i>Point Neighbor</i> meliputi peta dengan data <i>Point Neighbor</i> yang telah diinputkan beserta tabel data <i>Point Neighbor</i> dengan atribut <ul style="list-style-type: none"> - Id list - Id Router - Titik tetangga - Action Tombol <i>Edit</i> dan <i>Delete</i> |
| Skenario alternatif : Form data tidak lengkap | |

| Aksi aktor | Reaksi sistem |
|---|--|
| 8a. Klik <i>Save</i> | |
| | 9a. Menampilkan pesan “ data tidak lengkap ” |
| Skenario normal : Edit <i>Point Neighbor</i> | |
| Aksi aktor | Reaksi sistem |
| 3. Klik titik pada peta | |
| | 1. Menampilkan halaman form edit <i>Point Neighbor</i> -Id list -Id <i>Router</i> -Titik tetangga -Action Tombol <i>Save</i> , <i>Reset</i> |
| 5. Mengisi form <i>Edit</i> | |
| 6. Klik tombol <i>Save</i> | |
| | 8. Menampilkan halaman <i>Point Neighbor</i> |
| Skenario alternatif : data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 6a. Klik <i>Save</i> | |
| | 7a. Menampilkan pesan ”data tidak lengkap” |
| Skenario normal : Delete <i>Point Neighbor</i> | |
| Aksi aktor | Reaksi sistem |
| 5. klik <i>Delete</i> | |
| | 4. Menampilkan pesan “apakah yakin ingin menghapus data” |
| 6. klik ya | |
| | 8. Menampilkan halaman <i>Point Neighbor</i> |

4. Skenario Melihat Hasil Perhitungan

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario Melihat Hasil Perhitungan pada lampiran A.

5. Skenario Mengelola Data *User*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario mengelola *User* dijelaskan pada lampiran A.

6. Skenario Melihat Data *Router*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif. Skenario melihat data *Router* dijelaskan pada lampiran A.

4.2.6 *Sequance Diagram*

Sequance Diagram adalah diagram yang digunakan untuk menggambarkan interaksi antara objek dan sistem yang disusun pada sebuah urutan dan rangkaian waktu pada Penentuan Lokasi *Router* Untuk Optimasi Jaringan Komputer Dengan Menggunakan Metode *Dominating Set*.

1. *Sequance Diagram Login*

Penggambaran *Sequence Diagram Login* digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

2. *Sequance Diagram Mengelola Data Router*

Penggambaran *Sequence Diagram* mengelola data *Router* digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

3. *Sequance Diagram Mengelola Data Point Neighbor*

Penggambaran *Sequence Diagram* mengelola data *Point Neighbor* digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

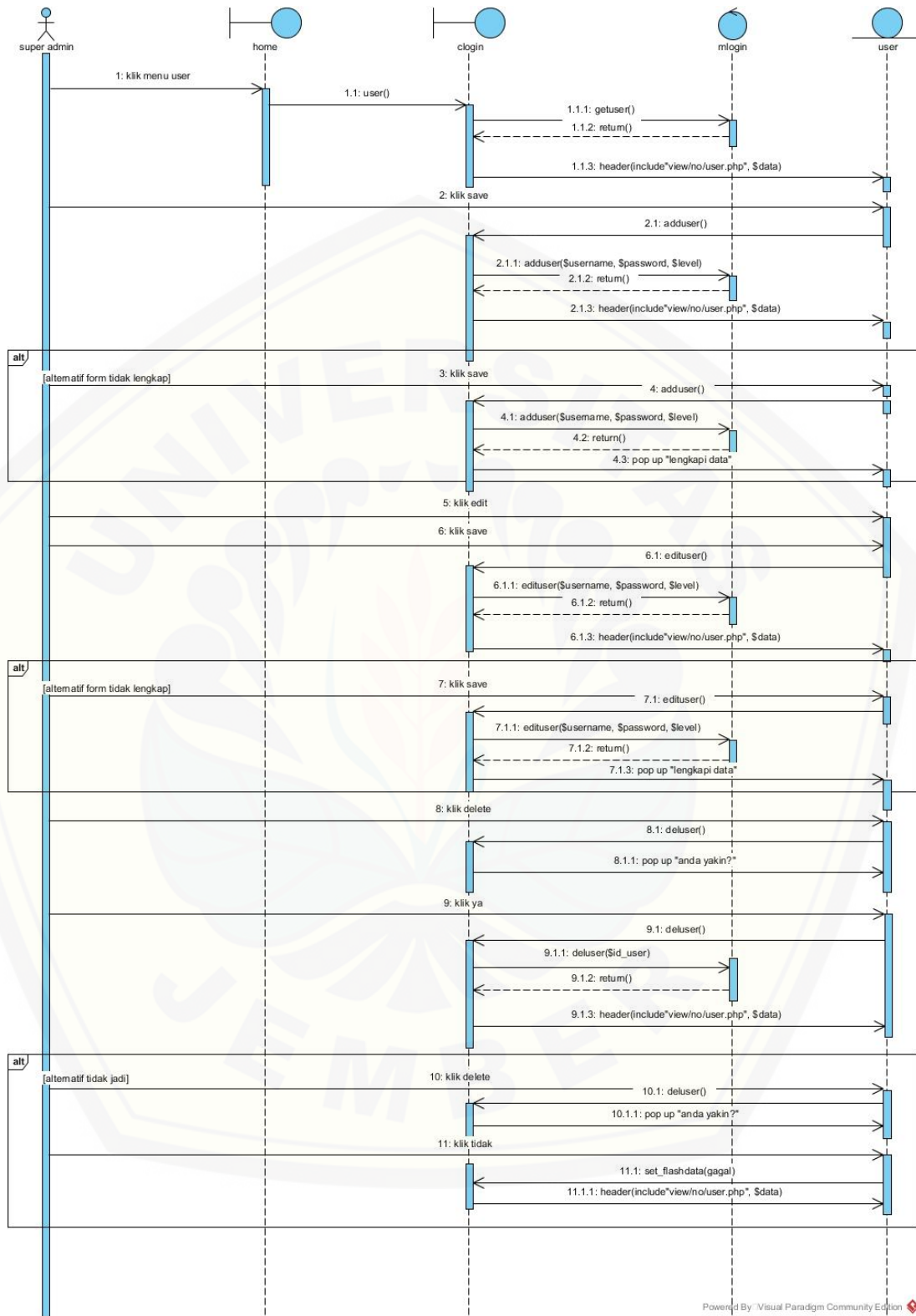
4. *Sequance Diagram Melihat Hasil Perhitungan*

Penggambaran *Sequence Diagram* melihat hasil perhitungan digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

5. *Sequance Diagram Mengelola Data User*

Penggambaran *Sequence Diagram* mengelola data *User* digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3.





Powered By Visual Paradigm Community Edition

Gambar 4.3 Sequence Mengelola data User

6. *Sequence Diagram* Melihat Data Router

Penggambaran *Sequence Diagram* melihat Data Router digunakan untuk menjelaskan fungsi dan *Method* yang akan dibuat seperti yang ditunjukkan pada lampiran C.

4.2.7 Activity Diagram

Activity Diagram menggambarkan alur aktivitas dalam sistem penentuan lokasi Router untuk optimasi jaringan komputer dengan yang akan dibangun.

1. *Activity Diagram Login*

Activity Diagram Login dapat dilihat pada lampiran B.

2. *Activity Diagram* Mengelola Data Router

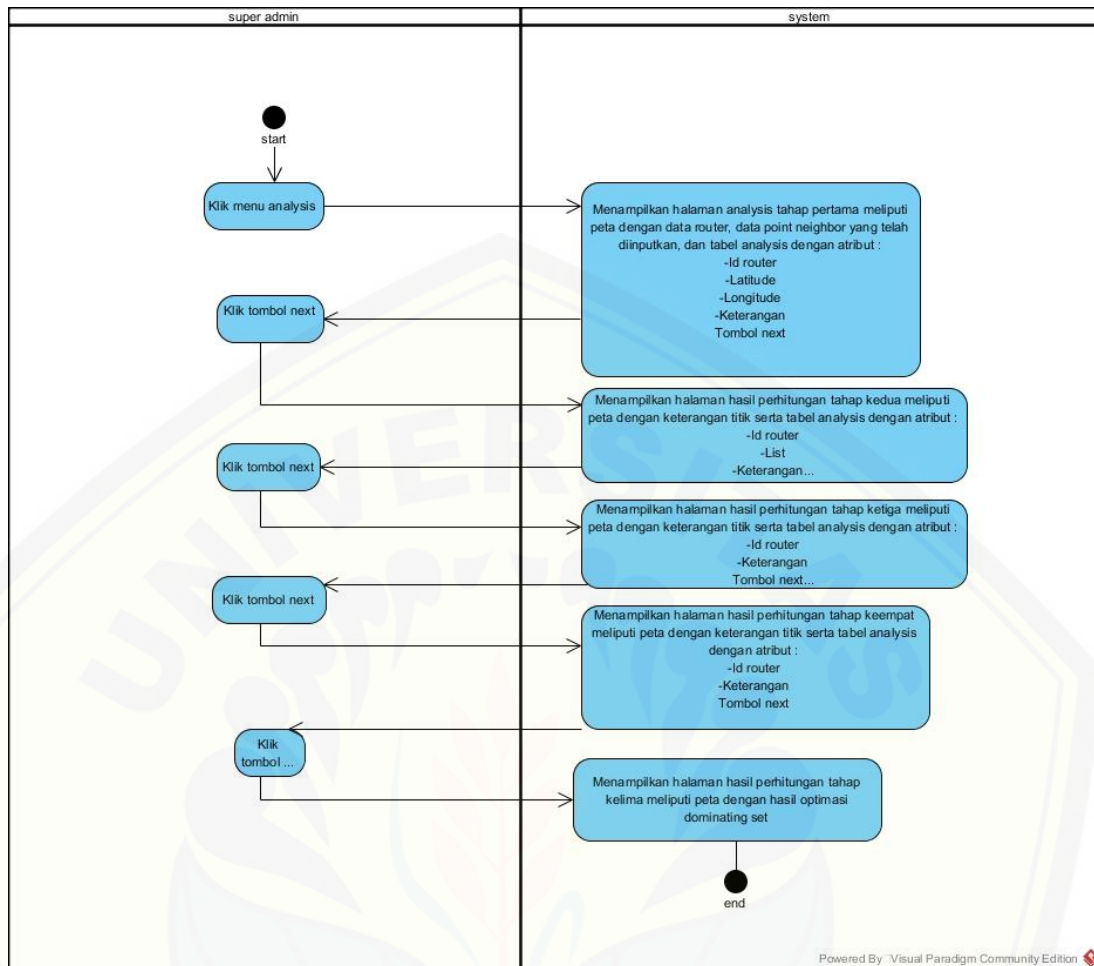
Activity Diagram mengelola data Router dapat dilihat pada lampiran B.

3. *Activity Diagram* Mengelola Data Point Neighbor

Activity Diagram mengelola data Point Neighbor dapat dilihat pada lampiran B.

4. *Activity Diagram* Melihat Hasil Perhitungan

Activity Diagram melihat hasil perhitungan dapat dilihat pada Gambar 4.4.



Gambar 4.4 *Activity Diagram* Melihat Hasil Perhitungan

5. *Activity Diagram* Mengelola Data User

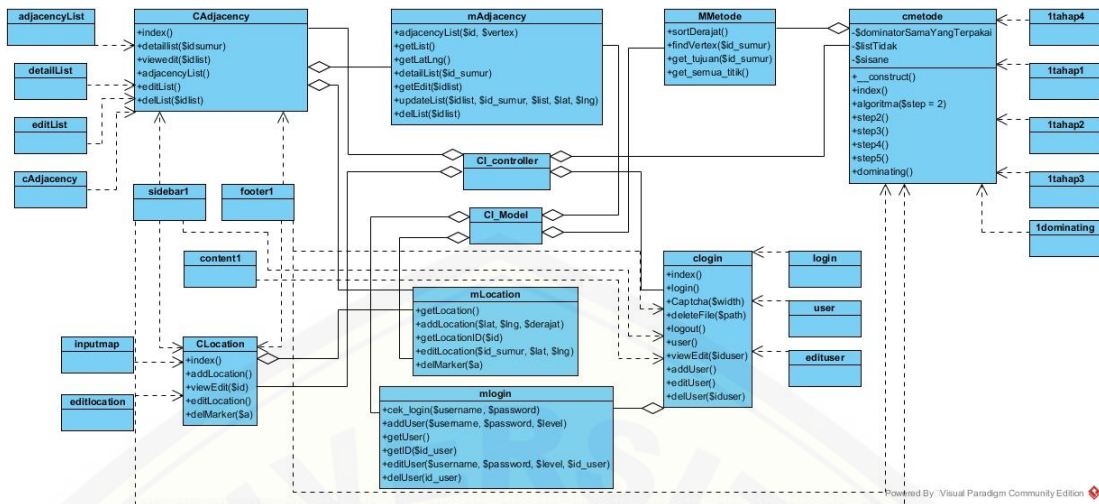
Activity Diagram mengelola data user dapat dilihat pada lampiran B.

6. *Activity Diagram* Mengelola Data Router

Activity Diagram mengelola data Router dapat dilihat pada lampiran B.

4.2.8 *Class Diagram*

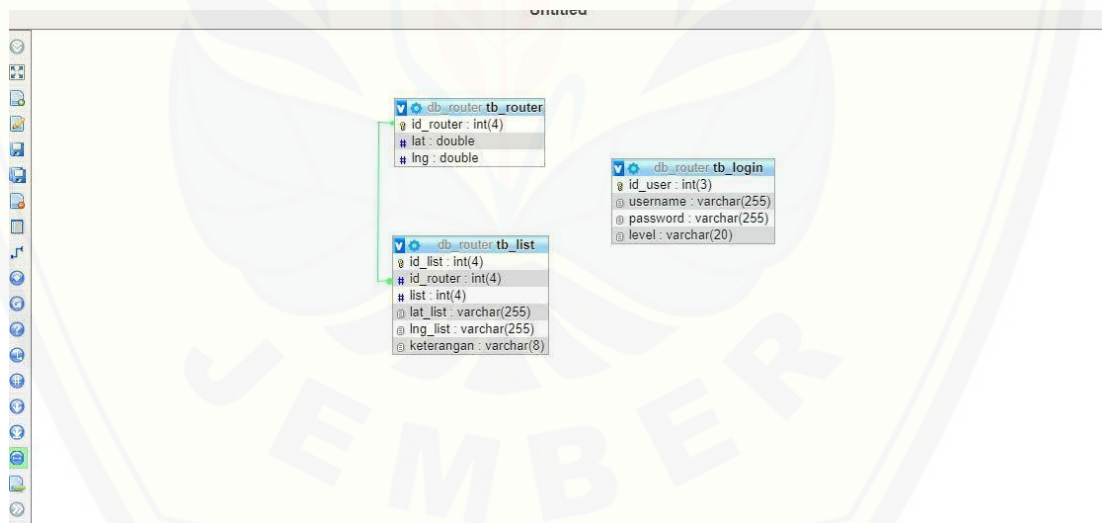
Class Diagram menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem. *Class Diagram* Sistem Penentuan Lokasi Router dapat dilihat pada Gambar 4.5



Gambar 4.5 Class Diagram

4.2.9 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD Sistem Sistem Penentuan Lokasi Router dapat dilihat pada gambar 4.6



Gambar 4.6 Entity Relational Diagram

4.3 Penulisan Kode Program

Tahap selanjutnya setelah proses desain sistem merupakan tahap penulisan kode program atau tahap implementasi dengan cara pengkodean atau coding dengan mengacu kepada desain sistem yang telah dibuat. Pengkodean dikerjakan menggunakan bahasa *Page Hyper Text Pre-Processor* (PHP). Manajemen database

dikerjakan dengan *Database Management System MySQL*. Proses koding dilakukan untuk mendapatkan sistem yang dapat dioperasikan oleh pengguna.

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Penulisan kode program untuk setiap fitur seperti yang telah digambarkan pada *Use Case Diagram* Gambar 4.2 ditulis dalam 3 bagian kelas, meliputi kelas *View*, *Controller*, dan model.

4.3.1 Kode Program *Login*

Kode program *login* terletak pada kelas *login* dan *clogin*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses yang dilakukan oleh sistem ini, pengguna harus memasukkan *Username* dan *Password* yang tepat agar proses dilanjutkan menuju step selanjutnya, karena jika tidak kode program akan menampilkan pesan *Username* atau *Password*.

4.3.2 Kode Program Pengelolaan Data *Router*

Kode program pengelolaan data proyek terletak pada kelas *clocation*, *mlocation*, *inputmap*, *editlocation*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses ini memiliki tugas untuk peletakan titik *Router* pada peta Universitas Jember, yang nantinya dibutuhkan sebagai data pada *Analysis*.

4.3.3 Kode Program Pengelolaan Data *Point Neighbor*

Kode program pengelolaan data *Point Neighbor* terletak pada kelas *cAdjacency*, *mAdjacency*, *AdjacencyList*, *detailList*, *editList*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses ini bertujuan untuk membuat titik tetangga *Router* yang ada pada peta Universitas Jember, Bertujuan sebagai salah satu data untuk proses *Analysis*.

4.3.4 Kode Program Melihat Hasil Perhitungan

Kode program pengelolaan hasil perhitungan terletak pada kelas *cmetode*, *mMetode*, *1tahap1*, *1tahap2*, *1tahap3*, *1tahap4*, *1dominating*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses ini merupakan pengolahan data *Router* dan *Point Neighbor* yang nantinya akan menghasilkan *Output* berupa titik pendominator, titik sisa, titik lokasi router, dan optimasi perhitungan.

4.3.5 Kode Program Pengelolaan Data *User*

Kode program melihat data manfaat terletak pada kelas *clogin*, *mlogin*, *user*, *edituser*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses ini memiliki tujuan untuk mengolah data *User* seperti, membuat *User* baru, ataupun Menghapus data *User*

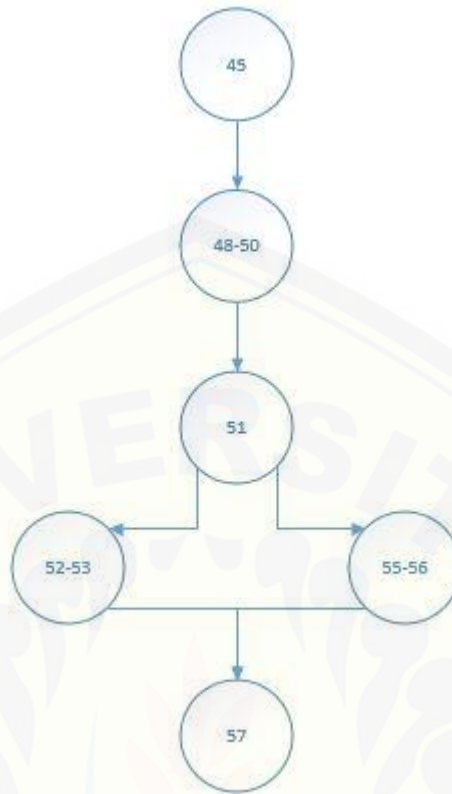
4.3.6 Kode Program Melihat Data *Router*

Kode program melihat data *Router* terletak pada kelas *clocation*, *mlocation*. Penulisan kode program ini dapat dilihat pada lampiran D. Proses ini memiliki tujuan untuk menunjukkan data *Router* bagi admin graf.

4.4 Pengujian Sistem

4.4.1 Pengujian White Box

Pengujian *White Box* pada Sistem analisis investasi sistem informasi ini dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian *Test Case*. Diagram alir pengujian sistem dapat dilihat pada gambar 4.7 dan 4.8. Pengujian komplektisitas siklomatik pada fitur mengelola data proyek dapat dilihat pada tabel 4.7 dan tabel 4.8.



Gambar 4.7 Tambah *Point Neighbor*

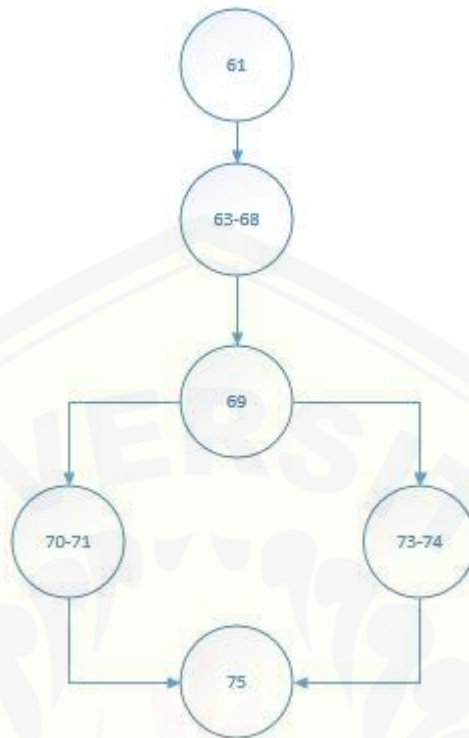
$$CC = E - N + 2 = 6 - 6 + 2 = 2$$

E= garis pada setiap cc

N= lingkaran pada setiap cc

Tabel 4. 4 *Test Case Point Neighbor*

| | |
|-------------------------------|---|
| <i>Test Case 1</i> | Menyimpan data ke <i>Database</i> |
| Target yang diharapkan | menyimpan data ke <i>Database</i> |
| Hasil Pengujian | Benar |
| <i>Path</i> | 45-48-50-51-52-53-57 |
| | |
| <i>Tes Case 2</i> | Menyimpan data ke <i>Database</i> |
| Target yang diharapkan | Tidak menyimpan data ke <i>Database</i> |
| Hasil Pengujian | Benar |
| <i>Path</i> | 45-48-50-51-55-56-57 |



Gambar 4.8 Edit Point Neighbor

$$CC = E - N + 2 = 20 - 14 + 2 = 8$$

P= cabang pada setiap cc

Tabel 4.5 Test Case Function Edit Point Neighbor

| | |
|-------------------------------|----------------------|
| Test Case 1 | Mengedit data tabel |
| Target yang diharapkan | Berhasil mengedit |
| Hasil Pengujian | Benar |
| Path | 61-63-68-69-70-71-75 |
| | |
| Tes Case 2 | Mengedit data tabel |
| Target yang diharapkan | Gagal mengedit |
| Hasil Pengujian | Benar |
| Path | 61-63-68-69-73-74-75 |

4.4.2 Pengujian *Black Box*

Pengujian *Black Box* merupakan pengujian yang dilakukan oleh pengguna dengan menjalankan program secara langsung dan menganalisa *input* dan *output* yang dihasilkan aplikasi. Pengujian black box pada fitur *Router Location* dapat dilihat pada tabel 4.6 dan untuk fitur analisis dapat dilihat pada tabel 4.7.

Tabel 4.6 Pengujian *Black Box Router Location*

| No | Fitur | Aksi | Hasil | Kesimpulan | |
|----|---------------------------------------|-----------------------|---|------------|-------|
| | | | | Berhasil | Tidak |
| 1. | Menambahkan data lokasi <i>Router</i> | Menekan tombol simpan | Menyimpan pada <i>database</i> | V | |
| 2. | Mengembalikan data seperti semula | Menekan tombol reset | Menampilkan data <i>Router</i> seperti semula | V | |

Tabel 4.7 Pengujian *Black Box Analysis*

| No | Fitur | Aksi | Hasil | Kesimpulan | |
|----|--------------------------------|---------------------|-------------------------------|------------|-------|
| | | | | Berhasil | Tidak |
| 1. | Melihat data hasil perhitungan | Menekan tombol next | Menampilkan hasil perhitungan | V | |

BAB. 6 PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut dapat digunakan sebagai acuan pada penelitian selanjutnya.

6.1 Kesimpulan

Kesimpulan dari penelitian yang dilakukan adalah:

1. Pengimplementasian algoritma *Greedy* untuk menentukan *Dominating Set* pada lokasi penempatan *Router* Universitas Jember adalah dengan menentukan *Vertex* terlebih dahulu pada peta universitas jember oleh admin *Router* selanjutnya menentukan titik dominator dari *Vertex* berdasarkan titik tetangganya, apabila list titik tetangganya belum didominasi oleh titik lain maka titik itu menjadi dominator, proses ketiga adalah penentuan titik sisa, proses ini mencari titik yang belum didominasi oleh titik dominator, tahapan selanjutnya mencari titik dominasi dari titik yang berada pada tahapan penentuan titik sisa lalu pada tahapan terakhir dilakukan perhitungan dominating set, Dari perhitungan tersebut di hasilkan $4 \leq 8 \leq 20$ dimana 4 merupakan jumlah minimum *Dominating Set*, 8 merupakan *Dominating set* yang dihasilkan dari perhitungan algoritma *Greedy*, dan 20 adalah jumlah maksimum *Dominating Set*. Disimpulkan bahwa hasil pencarian titik menggunakan algoritma *Greedy* telah optimal, karena jumlah *Dominating Set* yang ditemukan dari perhitungan algoritma *Greedy* mendekati jumlah minimum *Dominating Set*.
2. Penerapan algoritma *Greedy* untuk menentukan *Dominating Set* pada lokasi penempatan *Router* Universitas Jember dibangun dengan *Software Development Life Cycle* model *Waterfall*. Model *waterfall* digunakan karena kebutuhan sistem telah didefinisikan di awal. Tahap pengembangan dilakukan dengan tahap analisis kebutuhan, desain, penulisan kode program, pengujian dan pemeliharaan. Analisis kebutuhan dilakukan dengan cara mencari kebutuhan fungsional dan non fungsional yang dibutuhkan untuk

membangun sistem. Tahap desain sistem dilakukan dengan melakukan perancangan sistem dengan membuat beberapa diagram yang akan digunakan sebagai acuan dalam penulisan kode program. Selanjutnya dilakukan penulisan kode program dengan menggunakan bahasa pemrograman *PHP*, *framework CodeIgniter*, dan database manajemen *MySQL*. Tahap pengujian dilakukan setelah penulisan kode program yakni dengan melakukan pengujian *black box* dan *white box*. Lalu pengujian perhitungan manual dengan mencari keakuratan perhitungan.

6.2 Saran

Beberapa saran berikut diharapkan dapat memberikan perbaikan dalam penelitian selanjutnya, yaitu:

1. Pengembangan lebih lanjut diharapkan memiliki lingkup proyek yang lebih luas daripada lingkup Universitas Jember.
2. Membuat sistem berbasis android agar lebih mempermudah penggunaan secara mobile.
3. Diperlukannya pengembangan sistem informasi dengan mempertimbangkan faktor seperti jarak, luas, dan lahan.
4. Diperlukannya penggunaan algoritma lain sebagai pembanding dari hasil yang diperoleh dari penelitian ini.

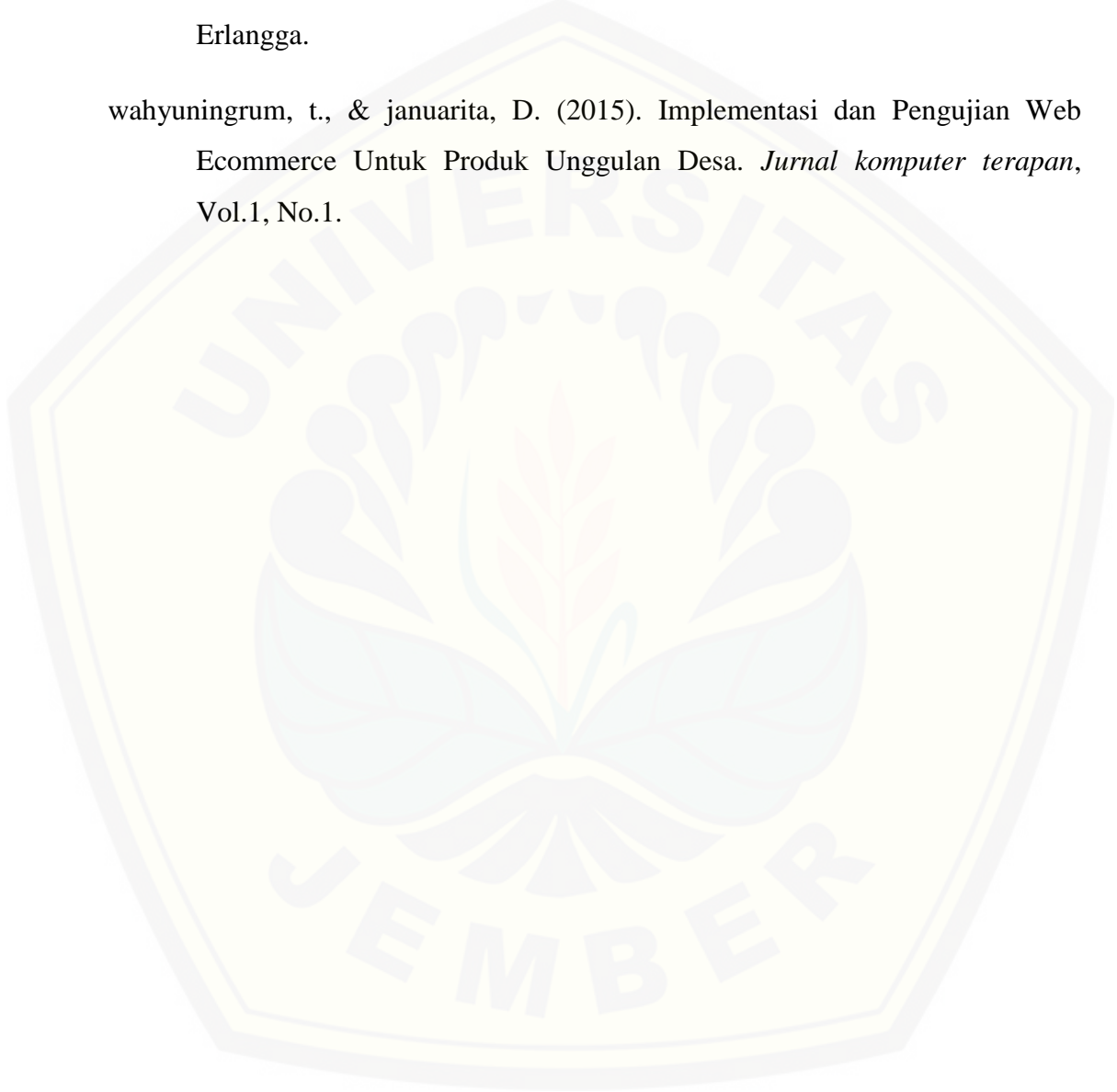
DAFTAR PUSTAKA

- Agustin, i. h. (2014). penerapan teori dominating set dalam instalasi client hub untuk jaringan intranet. 16.
- alanko, Crevals, Issopoussu, Ostergard, & Petterson. (2011). Computing the dominating number of grid graph. *the electronic journal*.
- Alfarisi, W. (2016). pencarian jalur terpendek pengiriman barang menggunakan algoritma A. *jurnal riset komputer*, 6.
- ardiyansyah, efendi, f. s., syaifullah, pinto, m., pujianto, & tempake, h. s. (2010). implementasi algoritma greedy untuk melakukan graph coloring. 8.
- fambudii, B. z. (2017). Implementasi Dominating Set untuk Sistem Penentuan Letak Panel Surya Sebagai sumber energi terbarukan pada jaringan lampu lalu lintas. *jurnal ilmiah mahasiswa*, 6.
- Iswadi, H. (2011). Batas Atas Bilangan Dominasi Lokasi Metrik Dari Graf Hasil Operasi Korona.
- Levitasari, A. (2016). simulasi sistem informasi penentuan tata letak mobil listrik dengan menggunakan himpunan dominasi pada graf (studi kasus agro techno park jubung universitas jember). *jurnal ilmiah mahasiswa*.
- Muhammad, & hasan, I. (2016). Analisa Dan Pengembangan Jaringan Wireless Berbasis Mikrotik Router Os V.5.20 Di Sekolah Dasar Negeri 24 Palu. *Jurnal Elektronik Sistem Informasi dan Komputer*, 19.
- Prayogo, R. A. (2014). Penggunaan Algoritma Greedy Dalam Penentuan Rute Wisata. 6.
- putra, i. e. (2013). Perancangan Jaringan Hotspot Berbasis Mikrotik Router Os 3.3.0. 5.

singgih, Bettiza, M., & Suswaini, E. (2014). Implementasi Algoritma Greedy Dalam Menentukan Rute Trayek Angkutan Kota (Angkot) Tanjung Pinang. 9.

Sommerville, i. (2011). *software engineering (rekayasa perangkat lunak)*. jakarta: Erlangga.

wahyuningrum, t., & januarita, D. (2015). Implementasi dan Pengujian Web Ecommerce Untuk Produk Unggulan Desa. *Jurnal komputer terapan*, Vol.1, No.1.



LAMPIRAN

A. Skenario

A.1 Skenario Login

Penjelasan urutan *login* yang akan dilakukan oleh 3 aktor pada sistem yakni, Super Admin, admin graf, dan admin *Router*. Penjelasan urutan aksi aktor dan reaksis sistem pada skenario normal dan skenario alternatif dapat dilihat pada tabel A.1 sampai dengan tabel A.4.

Tabel A. 1 Skenario *Login Super Admin*

| | |
|--|--|
| Nama Use Case | <i>Login</i> |
| Aktor | Super Admin |
| Deskripsi Singkat | Super Admin Akan mengakses sistem |
| Prekondisi | Super admin berada pada halaman utama pengguna dan akan memasukkan <i>Username</i> dan <i>Password</i> |
| Prakondisi | Super admin berada pada halaman utama super admin |
| Flow Events | |
| Skenario Normal : <i>login</i> | |
| Aksi Aktor | Reaksi Sistem |
| 1. Membuka Halaman <i>Website</i> | |
| | 2. Menampilkan Halaman <i>Login</i> |
| 3. Menginputkan <i>Username</i> dan <i>Password</i> | |
| 4. Klik tombol <i>login</i> | |
| | 5. Mengecek <i>Username</i> dan <i>password</i> ke <i>database</i> |
| | 6. Menampilkan Halaman <i>home</i> |
| Skenario Alternatif : <i>Username</i> dan <i>Password</i> Salah | |
| Aksi Aktor | Reaksi Sistem |
| 4a. Klik Tombol <i>login</i> | |
| | 6a. Menampilkan pesan “ <i>username</i> atau <i>password</i> salah” |

Tabel A. 2 Skenario *Login Admin Graf*

| | |
|---|---|
| Nama Use Case | <i>Login</i> |
| Aktor | Admin Graf |
| Deskripsi Singkat | Admin Graf Akan mengakses sistem |
| Prekondisi | Admin Graf berada pada halaman utama pengguna dan akan memasukkan <i>Username</i> dan <i>Password</i> |
| Prakondisi | Admin Graf berada pada halaman utama Admin Graf |
| Flow Events | |
| Skenario Normal : <i>login</i> | |
| Aksi Aktor | Reaksi Sistem |
| 1. Membuka Halaman <i>Website</i> | 2. Menampilkan Halaman <i>Login</i> |
| 3. Menginputkan <i>Username</i> dan <i>Password</i> | |
| 4. Klik tombol <i>login</i> | |
| | 5. Mengecek <i>Username</i> dan <i>password</i> ke database |
| | 6. Menampilkan Halaman <i>home</i> |
| Skenario Alternatif : <i>Username dan Password Salah</i> | |
| Aksi Aktor | Reaksi Sistem |
| 4a. Klik Tombol <i>login</i> | |
| | 6a. Menampilkan pesan “ <i>username</i> atau <i>password</i> salah” |

Tabel A. 3 Skenario *Login Admin Router*

| | |
|--------------------------|---|
| Nama Use Case | <i>Login</i> |
| Aktor | Admin Router |
| Deskripsi Singkat | Admin Router Akan mengakses sistem |
| Prekondisi | Admin Router berada pada halaman utama pengguna dan akan memasukkan <i>Username</i> dan <i>Password</i> |

| | |
|---|---|
| Prakondisi | Admin <i>Router</i> berada pada halaman utama Admin <i>Router</i> |
| Flow Events | |
| Skenario Normal : <i>login</i> | |
| Aksi Aktor | Reaksi Sistem |
| 1. Membuka Halaman <i>Website</i> | |
| | 2. Menampilkan Halaman <i>Login</i> |
| 3. Menginputkan <i>Username (varchar),length(Password)</i> | |
| 4. Klik tombol <i>login</i> | |
| | 5. Mengecek <i>Username</i> dan <i>password</i> ke database |
| | 6. Menampilkan Halaman <i>home</i> |
| Skenario Alternatif : <i>Username dan Password Salah</i> | |
| Aksi Aktor | Reaksi Sistem |
| 4a. Klik Tombol <i>login</i> | |
| | 6a. Menampilkan pesan " <i>username</i> atau <i>password</i> salah" |

A.2 Skenario Mengelola Data *User*

Merupakan skenario yang menjelaskan alur pada pengelolaan data *user* yang dilakukan oleh Super Admin. Alur aksi aktor dan reaksi sistem dapat dilihat pada tabel A.4

Tabel A. 4 Skenario Pengelolaan Data *User*

| | |
|--------------------------|---|
| Nama Use Case | Pengelolaan data <i>user</i> |
| Aktor | Super admin |
| Deskripsi singkat | Super admin akan mengelola data <i>user</i> |
| Prekondisi | Super Admin berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>user</i> pada menu <i>User Management</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>user</i> Berhasil ditambahkan • Data <i>user</i> berhasil dirubah • Data <i>user</i> berhasil dihapus |

| Flow Events | |
|---|---|
| Skenario Normal : Input User | |
| Aksi Aktor | Reaksi Sistem |
| 1. Klik menu <i>user management</i> | |
| | 2. Menampilkan halaman <i>user management</i> berisi tabel data user dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>Username</i> - <i>Password</i> - <i>Level</i> - <i>Action</i> Tombol edit dan delete Form input user meliputi: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol save dan reset |
| 3. Mengisi form tambah <i>user</i> <ul style="list-style-type: none"> - <i>Username (varchar), length(255)</i> - <i>Password (varchar), length (255)</i> - <i>Level (varchar), length (20)</i> | |
| 4. Klik tombol <i>save</i> | |
| | 5. Menampilkan halaman <i>user management</i> berisi tabel data user dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>Username</i> - <i>Password</i> - <i>Level</i> - <i>Action</i> Tombol edit dan delete Form input user meliputi: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol <i>save</i> dan <i>reset</i> |
| Skenario Alternatif : Form data tidak lengkap | |
| Aksi Aktor | Reaksi sistem |
| 6a. Klik tombol <i>save</i> | |

| | |
|--|---|
| | 7a. Menampilkan pesan “lengkapi data” |
| Skenario Normal : Edit user | |
| Aksi Aktor | Reaksi Sistem |
| 3. Klik tombol <i>edit user</i> | |
| | 4. Menampilkan halaman form <i>edit user</i> meliputi: - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol <i>save</i> dan <i>reset</i> |
| 6. Mengisi form <i>edit user</i> - <i>Username (varchar), length(255)</i> - <i>Password (varchar), length (255)</i> - <i>Level (varchar), length (20)</i> | |
| 6 Klik tombol <i>save</i> | |
| | 7. Menampilkan halaman <i>user management</i> berisi tabel data user dengan atribut - <i>Id</i> - <i>Username</i> - <i>Password</i> - <i>Level</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form input user meliputi: - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol <i>save</i> dan <i>reset</i> |
| Skenario Alternatif : Form data tidak lengkap | |
| Aksi aktor | Reaksi Sistem |
| 6a. Klik tombol <i>save</i> | |
| | 7a. Menampilkan pesan “lengkapi data” |
| Skenario Normal Delete User | |
| Aksi aktor | Reaksi sistem |
| 3. Klik tombol <i>delete user</i> | |

| | |
|---|--|
| | 4. Menampilkan pesan “apakah yakin anda ingin menghapus data ini ?” |
| 5. Klik ya | |
| | 6 Menampilkan halaman <i>user management</i> berisi tabel data user dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>Username</i> - <i>Password</i> - <i>Level</i> - <i>Action</i> Tombol edit dan delete Form input user meliputi: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol <i>save</i> dan <i>reset</i> |
| Skenario Alternatif : Klik tidak | |
| Aksi aktor | Reaksi Sistem |
| 5a. Klik tombol tidak | |
| | 6a. Menampilkan halaman <i>user management</i> berisi tabel data user dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>Username</i> - <i>Password</i> - <i>Level</i> - <i>Action</i> Tombol edit dan delete Form input user meliputi: <ul style="list-style-type: none"> - <i>Username</i> - <i>Password</i> - <i>Level</i> Tombol <i>save</i> dan <i>reset</i> |

A.3 Skenario Mengelola Data Router

Merupakan skenario yang menjelaskan alur mengelola Data Router oleh Super Admin dan Admin Router . Alur aksi aktor dan reaksi sistem dapat dilihat pada tabel A5 dan A.6.

Tabel A. 5 Skenario Pengelolaan Data Router Super Admin

| | |
|---------------|------------------------|
| Nama Use Case | Pengolahan data Router |
|---------------|------------------------|

| | |
|--|--|
| Aktor | Super admin |
| Deskripsi singkat | Super admin akan mengelola data <i>Router</i> |
| Prekondisi | Super Admin berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>Router</i> pada menu <i>Router Location</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>Router</i> berhasil ditambah • Data <i>Router</i> berhasil dihapus • Data <i>Router</i> berhasil diubah |
| Flow event | |
| Skenario Normal : Input <i>Router</i> | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>Router location</i> | |
| | <p>2. Menampilkan halaman <i>Router location</i> meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data lokasi <i>Router</i> dengan atribut</p> <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> <p>Tombol <i>edit</i> dan <i>delete</i> Form input <i>Router</i> meliputi:</p> <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> <p>Tombol <i>save</i>, <i>reset</i>, dan <i>find</i>.</p> |
| 3. Klik lokasi pada peta | |
| | <p>4. Menampilkan titik baru pada peta lokasi yang dipilih serta otomatis mengisi <i>langitude</i> dan <i>latitude</i> lokasi <i>Router</i> pada form input <i>Router</i></p> |
| 5. Klik tombol <i>save</i> | |
| | <p>6. Menampilkan halaman <i>Router location</i> meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data lokasi <i>Router</i> dengan atribut</p> <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> <p>Tombol <i>edit</i> dan <i>delete</i></p> |

| | |
|--|--|
| | Form input <i>Router</i> meliputi: - <i>latitude</i> - <i>longtitude</i> Tombol <i>save</i> , <i>reset</i> , dan <i>find</i> . |
| Skenario alternatif : Form data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 5a. Klik tombol <i>save</i> | |
| | 6a. Menampilkan pesan “ klik lokasi pada peta” |
| Skenario alternatif : klik reset | |
| Aksi aktor | Reaksi sistem |
| 5a. Klik tombol reset | |
| | 6a. Menampilkan halaman <i>Router</i> location dan menghilangkan titik yang telah dibuat |
| Skenario normal : edit <i>Router</i> | |
| Aksi aktor | Reaksi sistem |
| 3a. Klik salah satu titik pada peta | |
| | 4a. Menampilkan pop up detail data <i>Router</i> |
| 5a. Klik <i>edit</i> | |
| | 6a. Menampilkan peta dengan titik yang bisa digeser |
| 7a. Geser titik kedaerah yang diinginkan | |
| 8a. Klik <i>save</i> | |
| | 9a. Menampilkan halaman <i>Router</i> location meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data lokasi <i>Router</i> dengan atribut - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form input <i>Router</i> meliputi: - <i>latitude</i> - <i>longtitude</i> Tombol <i>save</i> , <i>reset</i> , dan <i>find</i> . |
| Skenario alternatif : data tidak lengkap | |

| Aksi aktor | Reaksi sistem |
|---|---|
| 9a. Klik <i>save</i> | |
| | 10a. Menampilkan pesan “ please fill out this field |
| Skenario normal : <i>delete Router</i> | |
| Aksi aktor | Reaksi sistem |
| 3. klik pada salah satu titik | |
| | 4. Menampilkan pop up detail data <i>Router</i> |
| 5. klik <i>delete</i> | |
| | 6. Menampilkan halaman <i>Router</i> location meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data <i>lokasi Router</i> dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form <i>input Router</i> meliputi: <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> Tombol <i>save</i> , <i>reset</i> , dan <i>find</i> . |

Tabel A. 6 Skenario Pengelolaan Data *Router* Admin *Router*

| | |
|--------------------------|---|
| Nama Use Case | Pengolahan data <i>Router</i> |
| Aktor | Admin <i>Router</i> |
| Deskripsi singkat | Admin <i>Router</i> akan mengelola data <i>Router</i> |
| Prekondisi | Admin <i>Router</i> berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>Router</i> pada menu <i>Router Location</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>Router</i> berhasil ditambah • Data <i>Router</i> berhasil dihapus • Data <i>Router</i> berhasil diubah |
| Flow event | |

| Skenario Normal : Input Router | |
|--|---|
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>Router</i> location | 2. Menampilkan halaman <i>Router</i> location meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data <i>lokasi Router</i> dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form input <i>Router</i> meliputi: <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> Tombol <i>save</i> , <i>reset</i> , dan <i>find</i> . |
| 3. Klik lokasi pada peta | 4. Menampilkan titik baru pada peta lokasi yang dipilih serta otomatis mengisi <i>longitude</i> dan <i>latitude</i> lokasi <i>Router</i> pada form input <i>Router</i> |
| 5. Klik tombol <i>save</i> | 6. Menampilkan halaman <i>Router</i> location meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data <i>lokasi Router</i> dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form input <i>Router</i> meliputi: <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> Tombol <i>save</i> , <i>reset</i> , dan <i>find</i> . |
| Skenario alternatif : Form data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 5a. Klik tombol <i>save</i> | 6a. Menampilkan pesan “ klik lokasi pada peta” |
| Skenario alternatif : klik <i>reset</i> | |

| Aksi aktor | Reaksi sistem |
|---|---|
| 5a. Klik tombol <i>reset</i> | |
| | 6a. Menampilkan halaman <i>Router location</i> dan menghilangkan titik yang telah dibuat |
| Skenario normal : <i>edit Router</i> | |
| Aksi aktor | Reaksi sistem |
| 3a. Klik salah satu titik pada peta | |
| | 4a. Menampilkan pop up detail data <i>Router</i> |
| 5a. Klik <i>edit</i> | |
| | 6a. Menampilkan peta dengan titik yang bisa digeser |
| 7a. Geser titik ke daerah yang diinginkan | |
| 8a. Klik <i>save</i> | |
| | 9a. Menampilkan halaman <i>Router location</i> meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data <i>lokasi Router</i> dengan atribut <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> Form <i>input Router</i> meliputi: <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> Tombol <i>save, reset, dan find.</i> |
| Skenario alternatif : data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 9a. Klik <i>save</i> | |
| | 10a. Menampilkan pesan “ <i>please fill out this field</i> ” |
| Skenario normal : <i>delete Router</i> | |
| Aksi aktor | Reaksi sistem |
| 3. klik pada salah satu titik | |
| | 4. Menampilkan pop up detail data <i>Router</i> |

| | |
|-----------------------|---|
| 5. klik <i>delete</i> | |
| | <p>6. Menampilkan halaman <i>Router location</i> meliputi peta dengan data <i>Router</i> yang telah diinputkan beserta tabel data lokasi <i>Router</i> dengan atribut</p> <ul style="list-style-type: none"> - <i>Id</i> - <i>latitude</i> - <i>longtitude</i> - <i>Action</i> <p>Tombol <i>edit</i> dan <i>delete</i></p> <p>Form input <i>Router</i> meliputi:</p> <ul style="list-style-type: none"> - <i>latitude</i> - <i>longtitude</i> <p>Tombol <i>save</i>, <i>reset</i>, dan <i>find</i>.</p> |

A.4 Skenario melihat Data Router

Merupakan skenario yang menjelaskan alur untuk melihat data *Router* yang dilakukan oleh admin graf. Alur aksi aktor dan reaksi sistem dapat dilihat pada tabel A.7.

Tabel A. 7 Skenario Melihat Data Router Admin Graf

| | |
|--------------------------------------|---|
| Nama Use Case | <i>View data Router</i> |
| Aktor | Admin graf |
| Deskripsi singkat | Admin graf akan melihat data <i>Router</i> |
| Prekondisi | Admin Graf berada pada halaman <i>Dashboard</i> dan akan melihat info data <i>Router</i> pada menu <i>Router Location</i> |
| Prakondisi | Admin Graf berada pada halaman info data <i>Router</i> |
| Flow Event | |
| Skenario normal : View Router | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>Router location</i> | |
| | 2. Menampilkan halaman <i>Router location</i> meliputi peta dengan data <i>Router</i> yang telah diinputkan. |
| 3. Klik salah satu titik | |

| | |
|--|---|
| | 4. Menampilkan detail informasi <i>Router</i> |
|--|---|

A.5 Skenario Mengelola *Point Neighbor*

Merupakan skenario yang menjelaskan alur untuk mengelola *Point Neighbor* yang dilakukan oleh Super Admin dan admin Graf. Alur aksi aktor dan reaksi sistem dapat dilihat pada tabel A.8 dan A9.

Tabel A. 8 Skenario Pengolahan data *Point Neighbor* Super Admin

| | |
|--|--|
| Nama Use Case | Pengolahan data <i>point neighbor</i> |
| Aktor | Super admin |
| Deskripsi singkat | Super admin akan mengelola data <i>point neighbor</i> |
| Prekondisi | Super Admin berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>point neighbor</i> pada menu <i>point neighbor</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>point neighbor</i> berhasil ditambah • Data <i>point neighbor</i> berhasil dihapus • Data <i>point neighbor</i> berhasil diubah |
| Flow event | |
| Skenario Normal : Input <i>Router</i> | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>point neighbor</i> | |
| | 2. Menampilkan halaman <i>point neighbor</i> meliputi peta dengan data <i>point neighbor</i> yang telah diinputkan beserta tabel data <i>point neighbor</i> dengan atribut <ul style="list-style-type: none"> - Id list - Id <i>Router</i> - Titik tetangga - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> |

| | |
|--|---|
| 3. Klik titik pada peta | |
| | 4. Menampilkan informasi detail <i>point neighbor</i> |
| 5. Klik <i>add</i> | |
| | 6. Menampilkan halaman <i>point neighbor</i> meliputi peta dan form tambah <i>point neighbor</i> Form input <i>Router</i> meliputi: - <i>Id Router</i> - Titik tetangga - <i>Latitude</i> - <i>longitude</i> Tombol <i>save, reset,</i> |
| 7. Mengisi form <i>point neighbor</i> | |
| 8. Klik <i>save</i> | |
| | 9. Menampilkan halaman <i>point neighbor</i> meliputi peta dengan data <i>point neighbor</i> yang telah diinputkan beserta tabel data <i>point neighbor</i> dengan atribut - <i>Id list</i> - <i>Id Router</i> - Titik tetangga - <i>Action</i> Tombol <i>edit</i> dan <i>delete</i> |
| Skenario alternatif : Form data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 8a. Klik <i>save</i> | |
| | 9a. Menampilkan pesan “ data tidak lengkap ” |
| Skenario normal : Edit <i>point neighbor</i> | |
| Aksi aktor | Reaksi sistem |
| 3. Klik titik pada peta | |
| | 2. Menampilkan halaman form edit <i>point neighbor</i> - <i>Id list</i> - <i>Id Router</i> -Titik tetangga - <i>Action</i> Tombol <i>save , reset</i> |
| 5. Mengisi form <i>edit</i> | |

| | |
|---|--|
| 6. Klik tombol <i>save</i> | |
| | 8. Menampilkan halaman <i>point neighbor</i> |
| Skenario alternatif : data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 6a. Klik <i>save</i> | |
| | 7a. Menampilkan pesan "data tidak lengkap" |
| Skenario normal : Delete point neighbor | |
| Aksi aktor | Reaksi sistem |
| 5. klik <i>delete</i> | |
| | 4. Menampilkan pesan "apakah yakin ingin menghapus data" |
| 6. klik <i>ya</i> | |
| | 8. Menampilkan halaman <i>point neighbor</i> |

Tabel A. 9 Skenario Pengelolaan data *Point Neighbor* Admin Graf

| | |
|---|---|
| Nama Use Case | Pengolahan data <i>point neighbor</i> |
| Aktor | Admin graf |
| Deskripsi singkat | Admin graf akan mengelola data <i>point neighbor</i> |
| Prekondisi | Admin Graf berada pada halaman <i>Dashboard</i> dan akan menambah, menghapus dan mengubah data <i>point neighbor</i> pada menu <i>point neighbor</i> |
| Prakondisi | <ul style="list-style-type: none"> • Data <i>point neighbor</i> berhasil ditambah • Data <i>point neighbor</i> berhasil dihapus • Data <i>point neighbor</i> berhasil diubah |
| Flow event | |
| Skenario Normal : Input point neighbor | |

| Aksi aktor | Reaksi sistem |
|--|--|
| 1. Klik menu <i>point neighbor</i> | |
| | 2. Menampilkan halaman <i>point neighbor</i> meliputi peta dengan data <i>point neighbor</i> yang telah diinputkan beserta tabel data <i>point neighbor</i> dengan atribut <ul style="list-style-type: none"> - Id list - Id Router - Titik tetangga - Action Tombol <i>edit</i> dan <i>delete</i> |
| 3. Klik titik pada peta | |
| | 4. Menampilkan informasi detail <i>point neighbor</i> |
| 5. Klik <i>add</i> | |
| | 6. Menampilkan halaman <i>point neighbor</i> meliputi peta dan form tambah <i>point neighbor</i> Form input Router meliputi: <ul style="list-style-type: none"> - Id Router - Titik tetangga - Latitude - longitude Tombol <i>save</i> , <i>reset</i> , |
| 7. Mengisi form <i>point neighbor</i> | |
| 8. Klik <i>save</i> | |
| | 9. Menampilkan halaman <i>point neighbor</i> meliputi peta dengan data <i>point neighbor</i> yang telah diinputkan beserta tabel data <i>point neighbor</i> dengan atribut <ul style="list-style-type: none"> - Id list - Id Router - Titik tetangga - Action Tombol <i>edit</i> dan <i>delete</i> |
| Skenario alternatif : Form data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 8a. Klik <i>save</i> | |
| | 9a. Menampilkan pesan “ data tidak lengkap ” |

| Skenario normal : <i>Edit point neighbor</i> | |
|---|--|
| Aksi aktor | Reaksi sistem |
| 3. Klik titik pada peta | |
| | 4. Menampilkan halaman form edit <i>point neighbor</i> |
| 5. Mengisi form <i>edit</i> | |
| 6. Klik tombol <i>save</i> | |
| | 8. Menampilkan halaman <i>point neighbor</i> |
| Skenario alternatif : data tidak lengkap | |
| Aksi aktor | Reaksi sistem |
| 6a. Klik <i>save</i> | |
| | 7a. Menampilkan pesan "data tidak lengkap" |
| Skenario normal : <i>Delete point neighbor</i> | |
| Aksi aktor | Reaksi sistem |
| 5. klik <i>delete</i> | |
| | 4. Menampilkan pesan "apakah yakin ingin menghapus data" |
| 6. klik ya | |
| | 8. Menampilkan halaman <i>point neighbor</i> |

A.6 Skenario Melihat hasil perhitungan *dominating set* dan algoritma *greedy*

Merupakan skenario yang menjelaskan alur untuk melihat hasil perhitungan dari sistem yang dilakukan oleh Super admin, admin Graf, dan admin *Router*. Alur aksi aktor dan reaksi sistem dapat dilihat pada tabel A.10 sampai A12.

Tabel A. 10 Skenario Hasil Perhitungan Super Admin

| | |
|--------------------------|---|
| Nama Use Case | <i>View</i> hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |
| Aktor | Super admin |
| Deskripsi singkat | Super admin melihat hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |

| | |
|--|---|
| Prekondisi | Super admin berada pada halaman <i>Dashboard</i> , dan akan melihat hasil perhitungan sistem pada menu <i>Analysis</i> |
| Prakondisi | Menampilkan hasil dari perhitungan dominating set dan point neighbor |
| Flow event | |
| Skenario Normal : View analysis | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>analysis</i> | |
| | 2. Menampilkan halaman <i>analysis</i> tahap pertama meliputi peta dengan data <i>Router</i> , data point neighbor yang telah diinputkan, dan tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - <i>Latitude</i> - <i>Longitude</i> - Keterangan Tombol <i>next</i> |
| 3. Klik tombol <i>next</i> | |
| | 4. Menampilkan halaman hasil perhitungan tahap kedua meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - <i>List</i> - Keterangan Tombol <i>next</i> |
| 5. Klik tombol <i>next</i> | |
| | 6. Menampilkan halaman hasil perhitungan tahap ketiga meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan Tombol <i>next</i> |
| 7. Klik tombol <i>next</i> | |
| | 8. Menampilkan halaman hasil perhitungan tahap keempat meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan Tombol <i>next</i> |

| | |
|----------------------------|---|
| 9. Klik tombol <i>next</i> | |
| | <p>10. -Menghitung nilai optimasi dengan rumus</p> $\left\lceil \frac{p}{1 + \Delta(G)} \right\rceil \leq \gamma(G) \leq p - \Delta(G)$ <p>- Menampilkan halaman hasil perhitungan tahap kelima meliputi peta dengan hasil optimasi <i>dominating set</i></p> |

Tabel A. 11 Skenario Hasil Perhitungan Admin Graf

| | |
|--|--|
| Nama Use Case | View hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |
| Aktor | Admin graf |
| Deskripsi singkat | Admin graf melihat hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |
| Prekondisi | Admin Graf berada pada halaman <i>Dashboard</i> , dan akan melihat hasil perhitungan sistem pada menu <i>Analysis</i> |
| Prakondisi | Menampilkan hasil dari perhitungan <i>dominating set</i> dan <i>point neighbor</i> |
| Flow event | |
| Skenario Normal : View analysis | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>analysis</i> | |
| | <p>2. Menampilkan halaman <i>analysis</i> tahap pertama meliputi peta dengan data <i>Router</i> , data <i>point neighbor</i> yang telah diinputkan, dan tabel <i>analysis</i> dengan atribut :</p> <ul style="list-style-type: none"> - <i>Id Router</i> - <i>Latitude</i> - <i>Longitude</i> - Keterangan <p>Tombol <i>next</i></p> |
| 3. Klik tombol <i>next</i> | |

| | |
|----------------------------|--|
| | <p>4. Menampilkan halaman hasil perhitungan tahap kedua meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut :</p> <ul style="list-style-type: none"> - <i>Id Router</i> - <i>List</i> - Keterangan <p>Tombol <i>next</i></p> |
| 5. Klik tombol <i>next</i> | |
| | <p>6. Menampilkan halaman hasil perhitungan tahap ketiga meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut :</p> <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan <p>Tombol <i>next</i></p> |
| 7. Klik tombol <i>next</i> | |
| | <p>8. Menampilkan halaman hasil perhitungan tahap keempat meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut :</p> <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan <p>Tombol <i>next</i></p> |
| 9. Klik tombol <i>next</i> | |
| | <p>10. -Menghitung nilai optimasi dengan rumus</p> $\left[\frac{p}{1 + \Delta(G)} \right] \leq \gamma(G) \leq p - \Delta(G)$ <ul style="list-style-type: none"> - Menampilkan halaman hasil perhitungan tahap kelima meliputi peta dengan hasil optimasi <i>dominating set</i> |

Tabel A. 12 Skenario Hasil Perhitungan Admin Router

| | |
|--------------------------|--|
| Nama Use Case | View hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |
| Aktor | Admin Router |
| Deskripsi singkat | Admin Router melihat hasil perhitungan <i>dominating set</i> dan algoritma <i>greedy</i> |

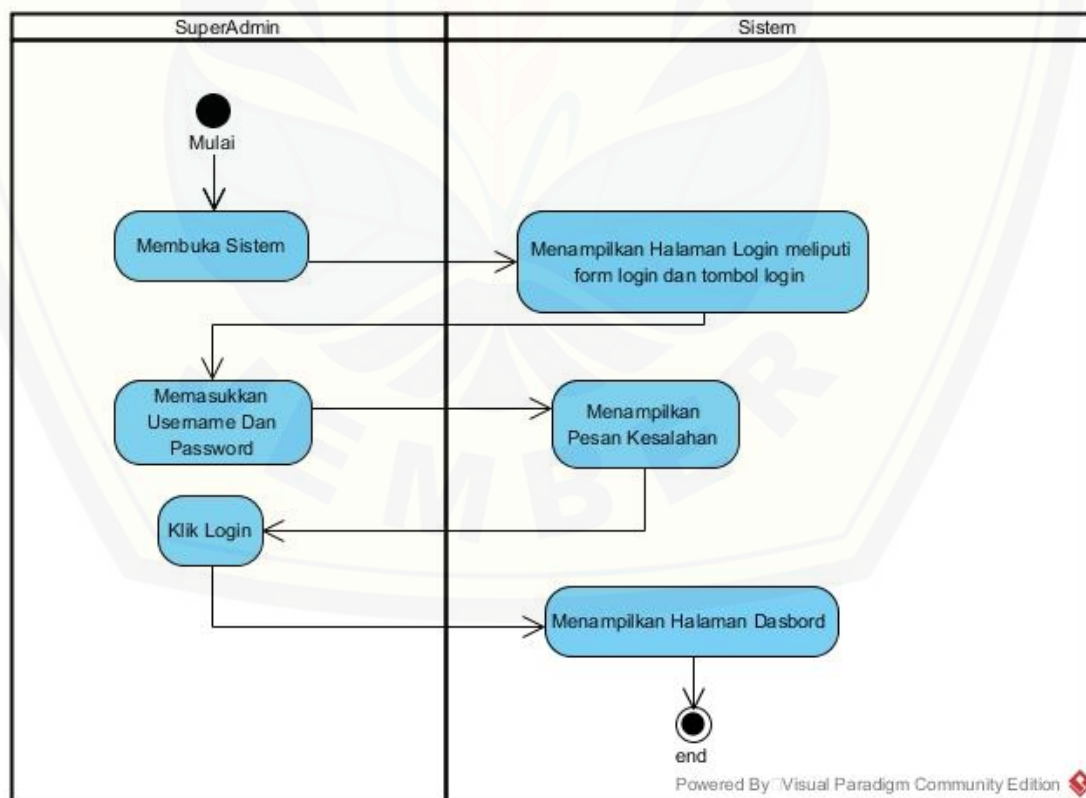
| | |
|--|---|
| Prekondisi | Admin <i>Router</i> berada pada halaman <i>Dashboard</i> , dan akan melihat hasil perhitungan sistem pada menu <i>Analysis</i> |
| Prakondisi | Menampilkan hasil dari perhitungan dominating set dan point neighbor |
| Flow event | |
| Skenario Normal : View analysis | |
| Aksi aktor | Reaksi sistem |
| 1. Klik menu <i>analysis</i> | |
| | 2. Menampilkan halaman <i>analysis</i> tahap pertama meliputi peta dengan data <i>Router</i> , data point neighbor yang telah diinputkan, dan tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - <i>Latitude</i> - <i>Longitude</i> - Keterangan Tombol <i>next</i> |
| 3. Klik tombol <i>next</i> | |
| | 4. Menampilkan halaman hasil perhitungan tahap kedua meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - <i>List</i> - Keterangan Tombol <i>next</i> |
| 5. Klik tombol <i>next</i> | |
| | 6. Menampilkan halaman hasil perhitungan tahap ketiga meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan Tombol <i>next</i> |
| 7. Klik tombol <i>next</i> | |
| | 8. Menampilkan halaman hasil perhitungan tahap keempat meliputi peta dengan keterangan titik serta tabel <i>analysis</i> dengan atribut : <ul style="list-style-type: none"> - <i>Id Router</i> - Keterangan Tombol <i>next</i> |

| | |
|-----------------------------------|---|
| <p>9. Klik tombol <i>next</i></p> | |
| | <p>10. -Menghitung nilai optimasi dengan rumus</p> $\left[\frac{p}{1 + \Delta(G)} \right] \leq \gamma(G) \leq p - \Delta(G)$ <p>- Menampilkan halaman hasil perhitungan tahap kelima meliputi peta dengan hasil optimasi <i>dominating set</i></p> |

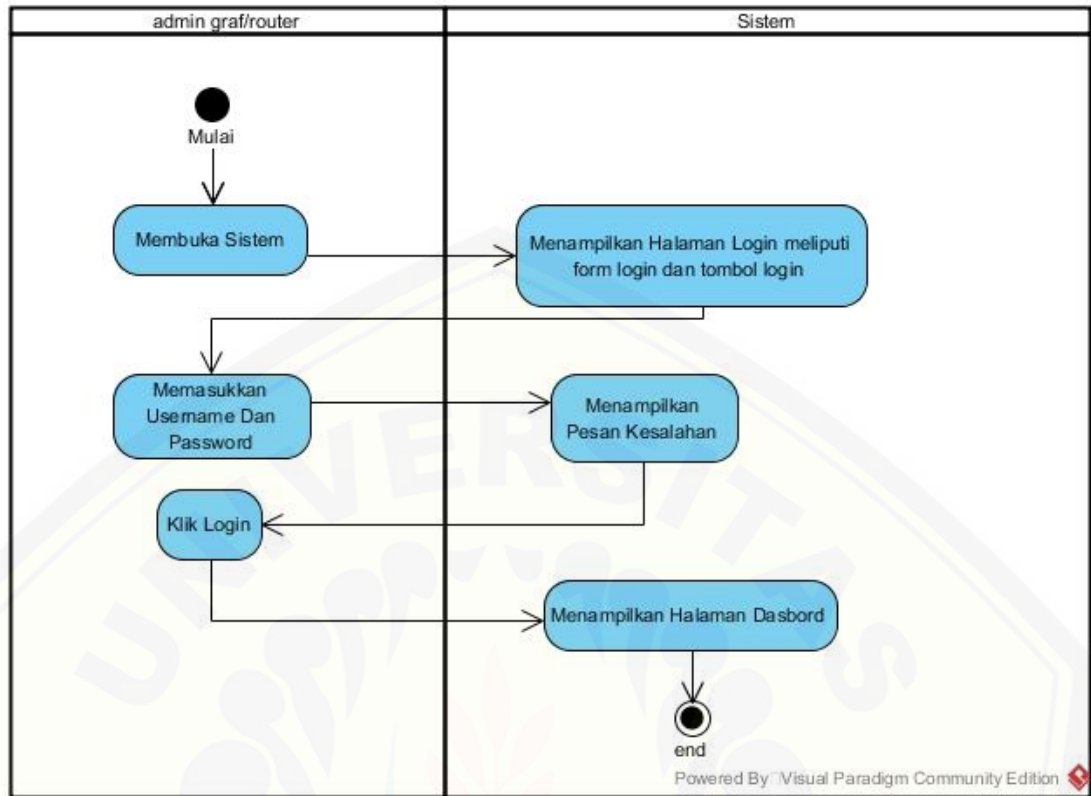
B. Activity Diagram

B.1 Activity Diagram Login

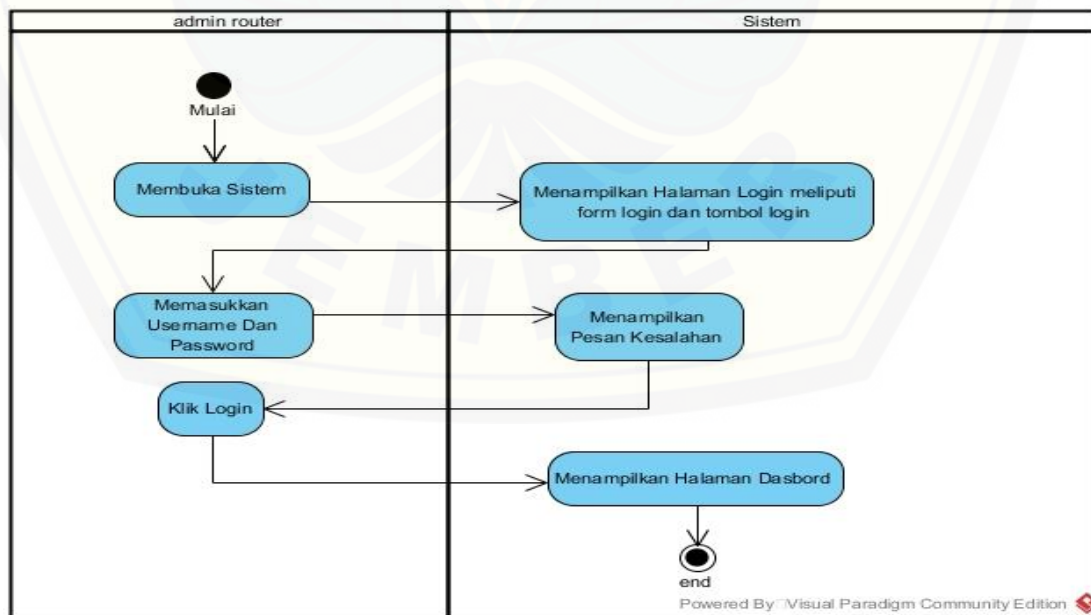
Activity Diagram login dapat dilihat pada gambar B.1 sampai dengan B.3.



Gambar B. 1 Login Super Admin



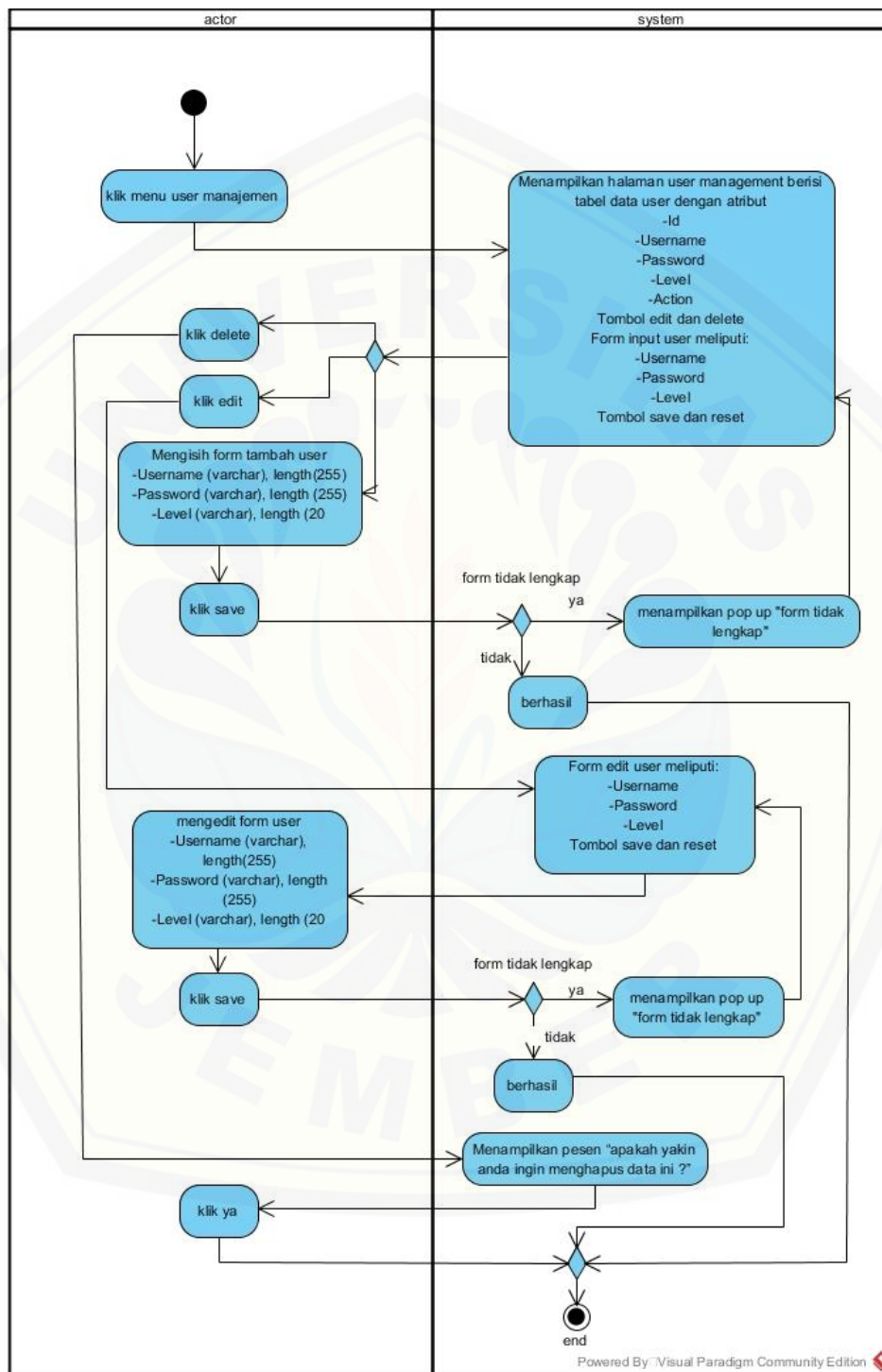
Gambar B. 2 Login Admin Graf



Gambar B. 3 Login Admin Router

B.2 Activity Diagram Mengelola Data User

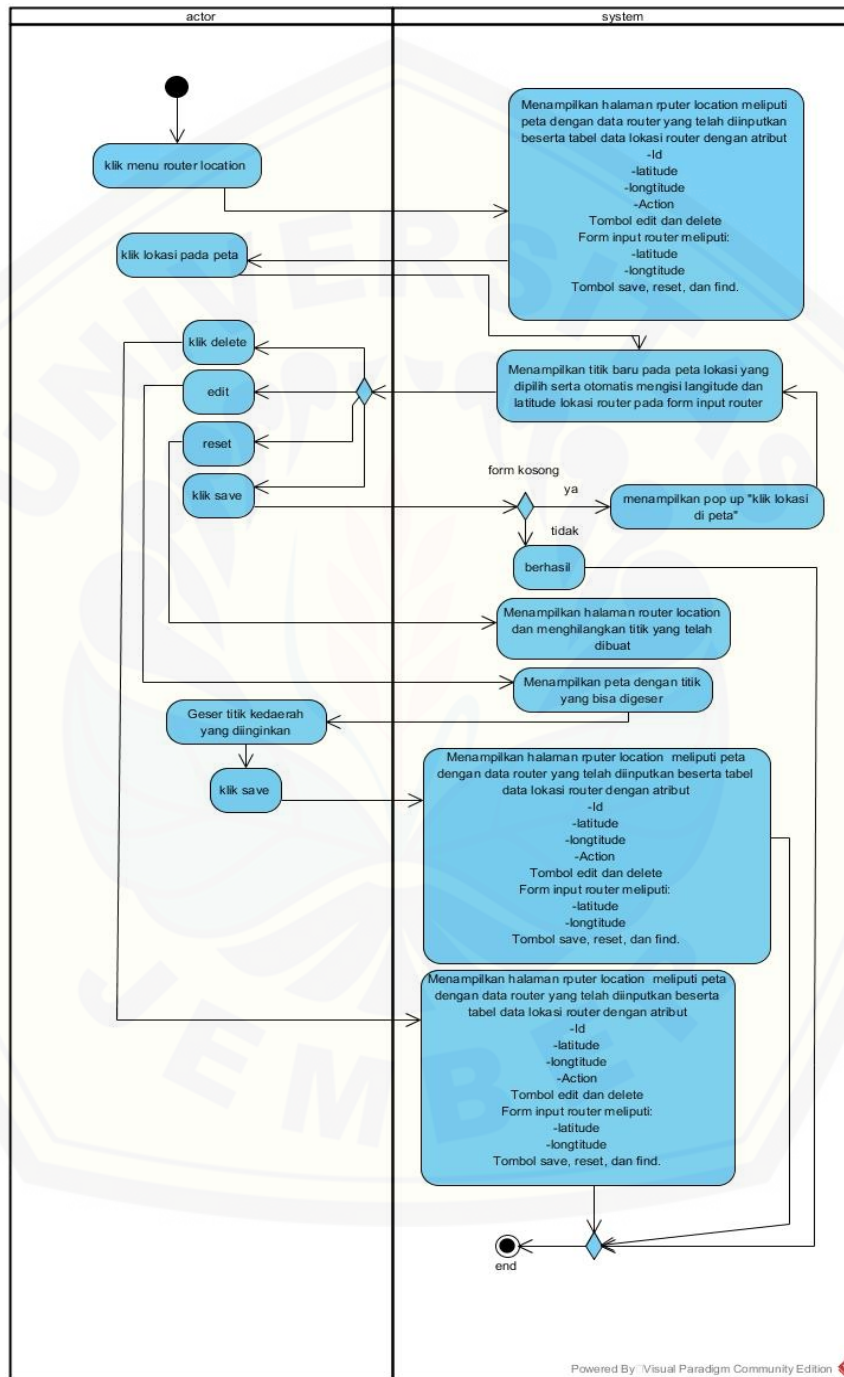
Activity diagram mengelola data user dapat dilihat pada gambar B.4.



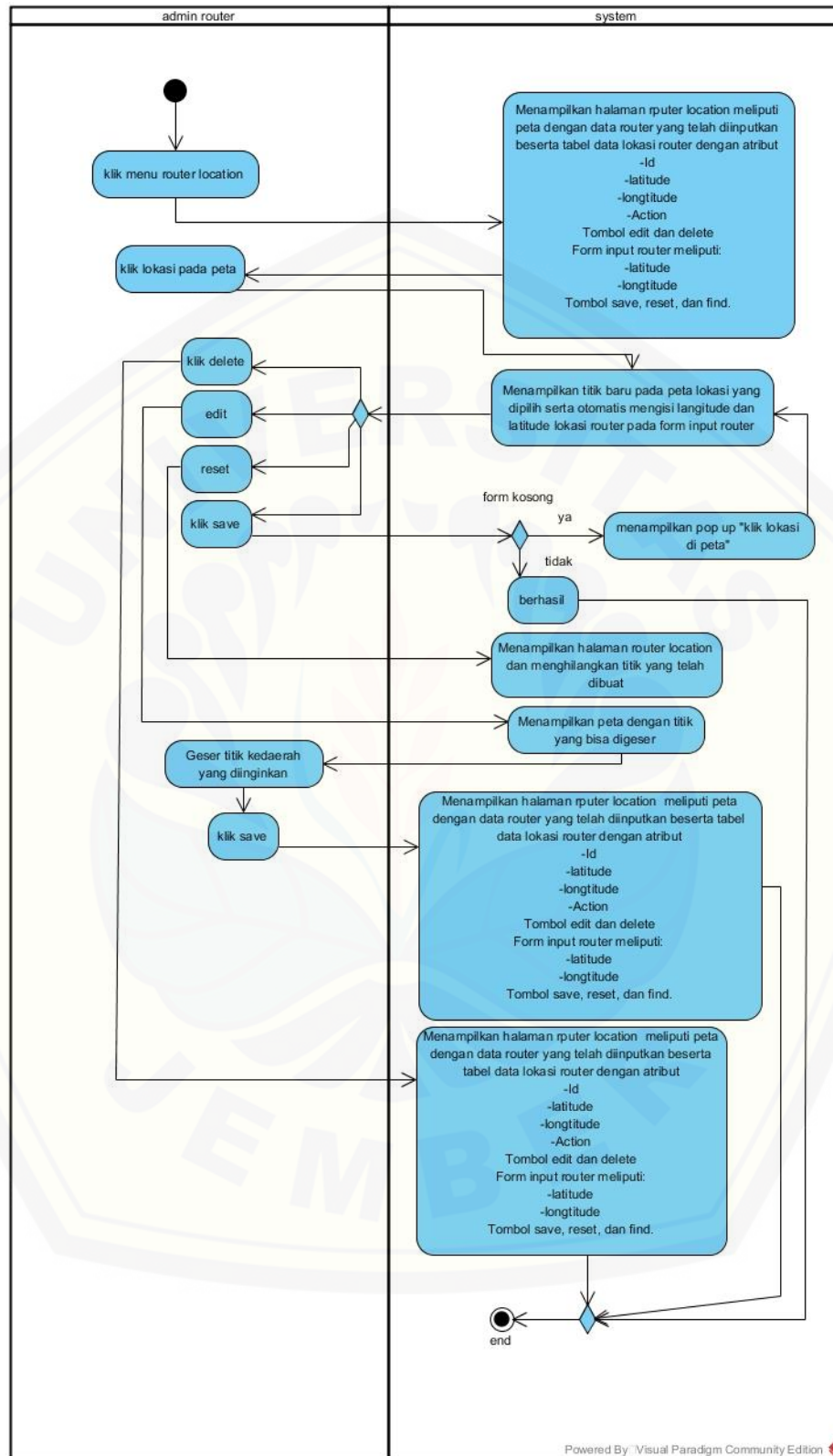
Gambar B. 4 Pengelolaan Data User

B.3 Activity Diagram Mengelola Data Router

Activity Diagram mengelola data Router dapat dilihat pada gambar B.5 dan B.6.



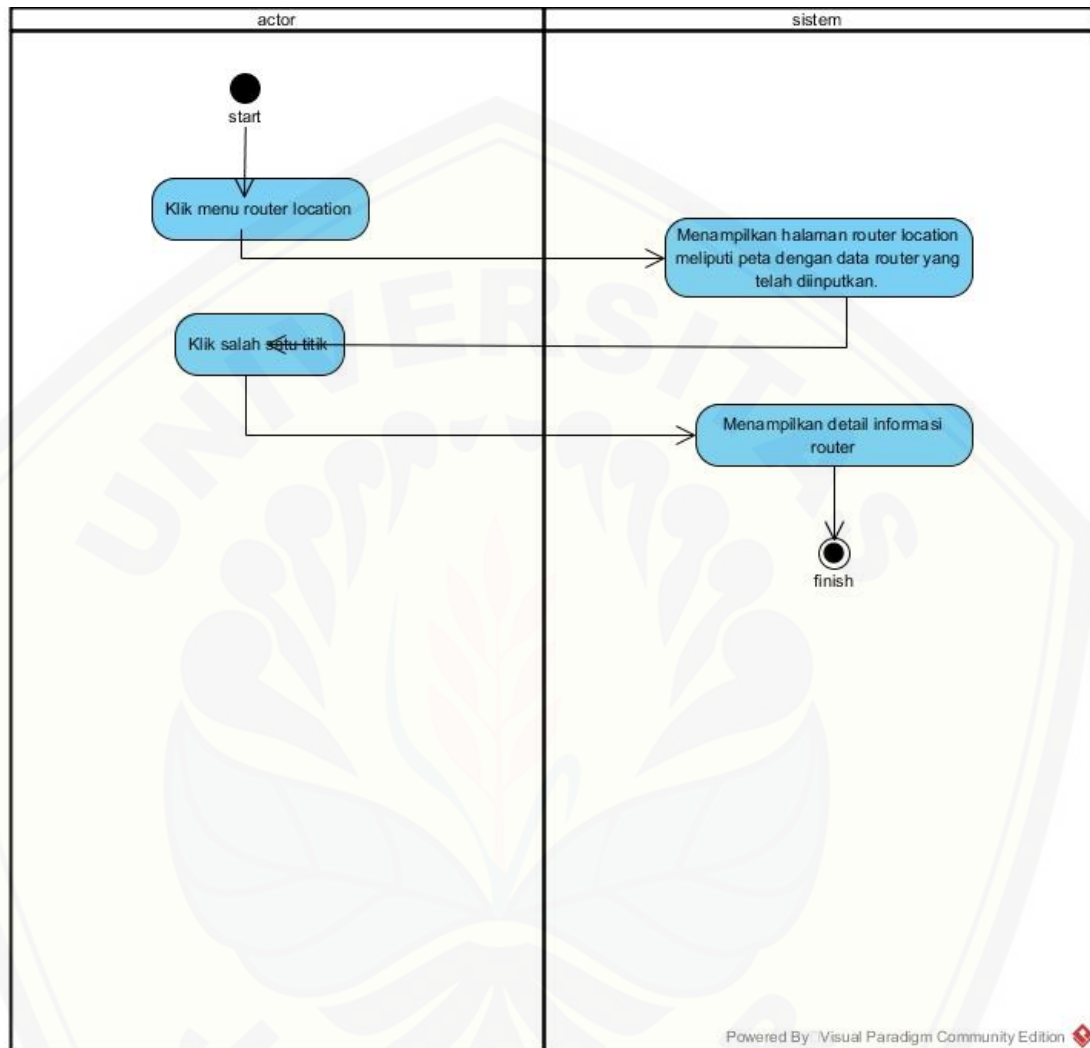
Gambar B. 5 Mengelola Data Router



Gambar B. 6 Mengelola Data Router admin Graf

B.4 Activity Diagram Melihat Data Router

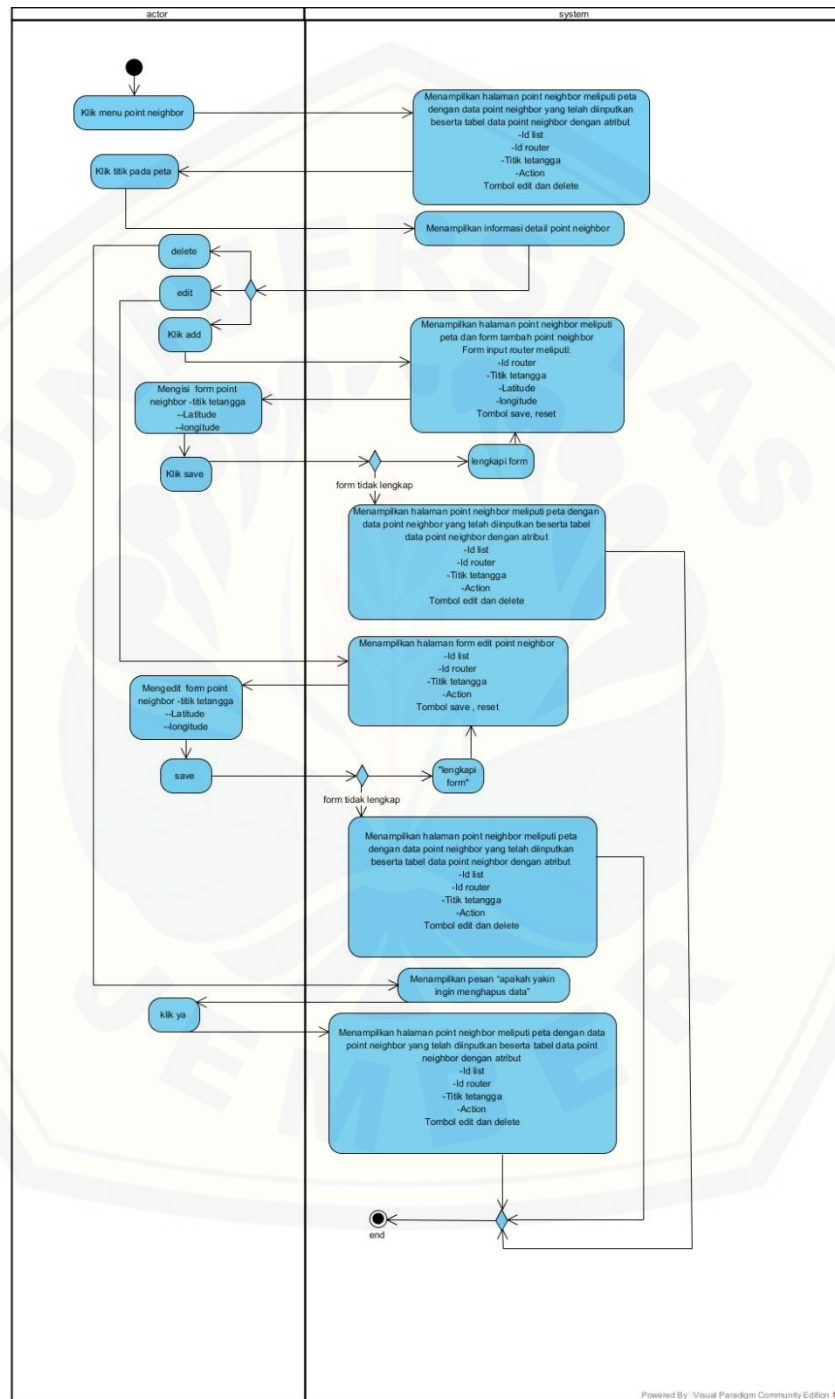
Activity Diagram melihat data Router dapat dilihat pada gambar B.7.



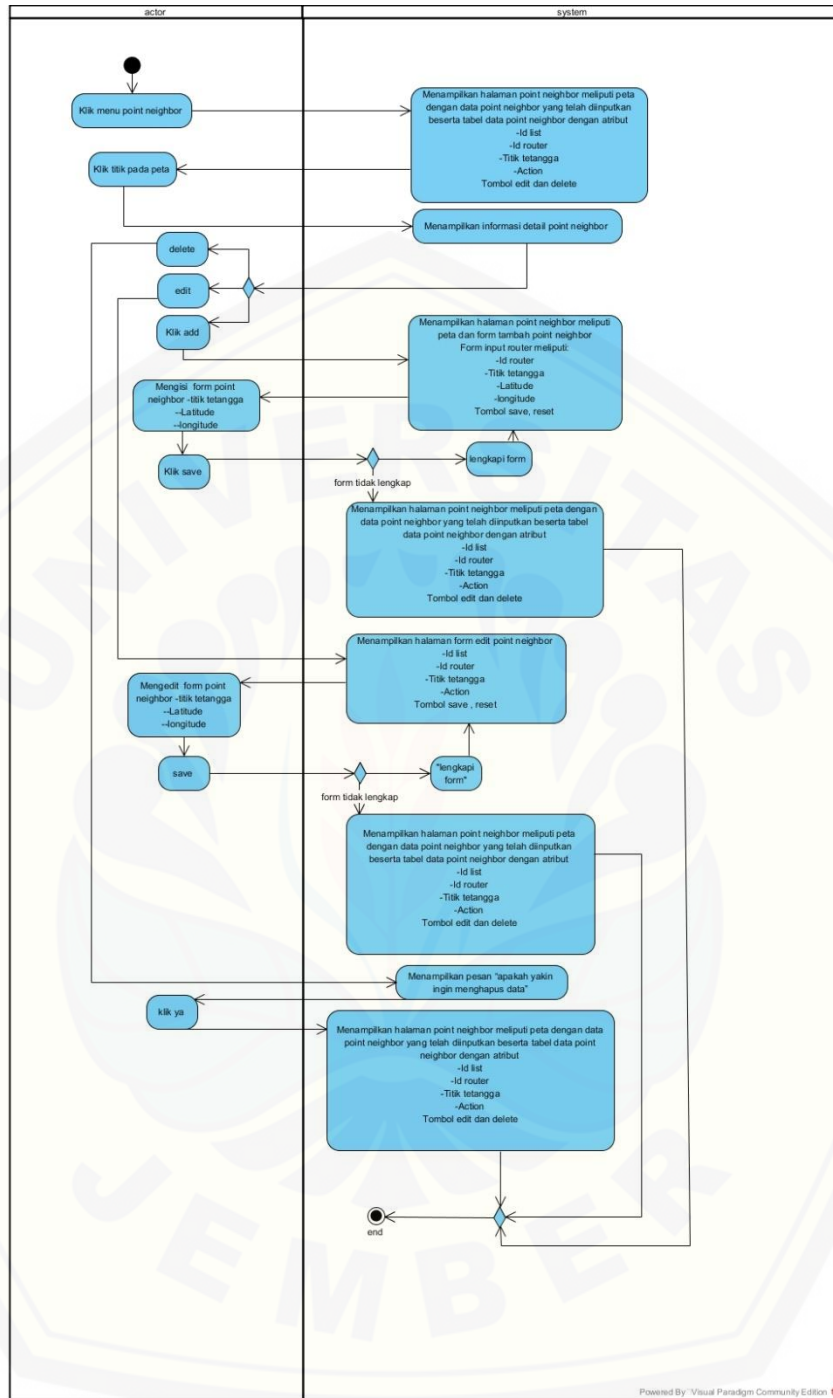
Gambar B. 7 Melihat Data Router Admin Graf

B.5 Activity Diagram Mengelola Point Neighbor

Activity Diagram mengelola data *point neighbor* dapat dilihat pada gambar B.8 dan B.7 .



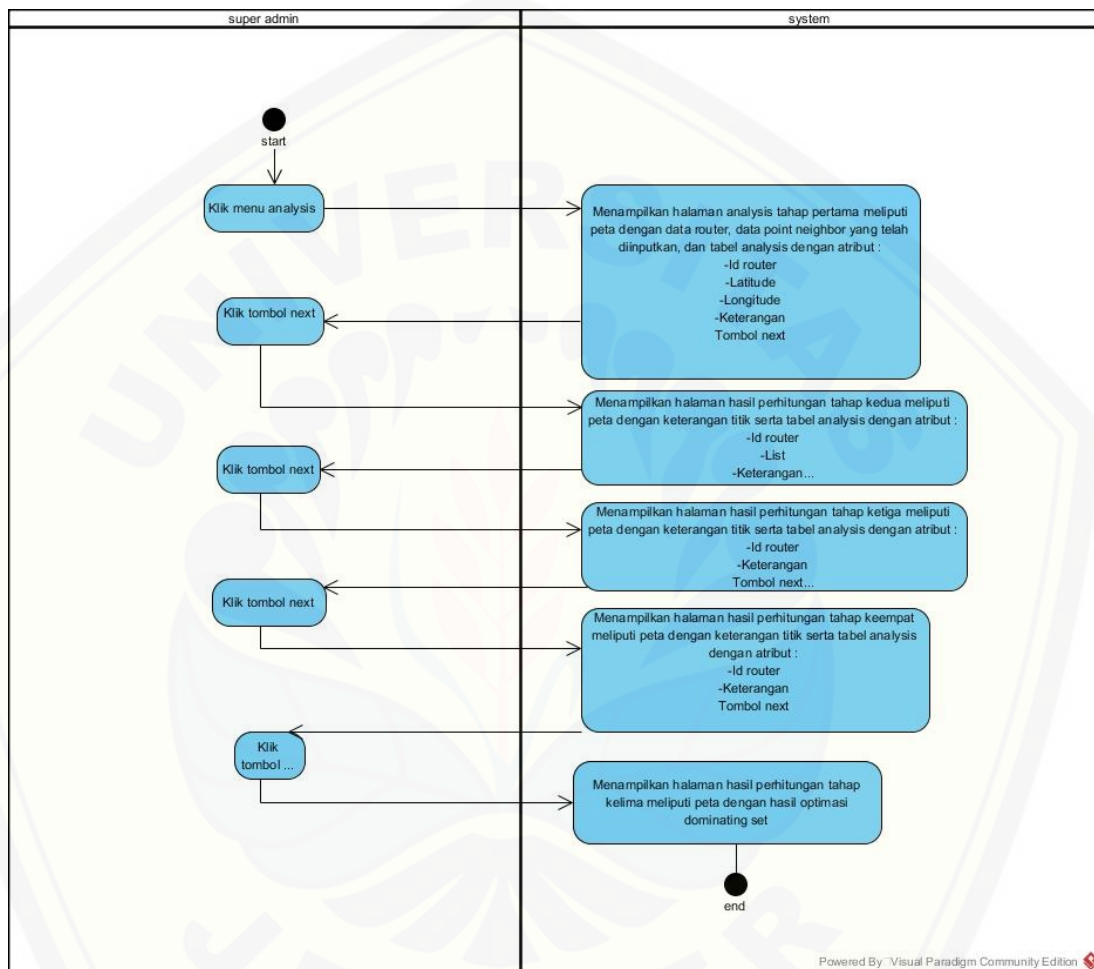
Gambar B. 8 Pengelolaan *Point Neighbor* Super Admin



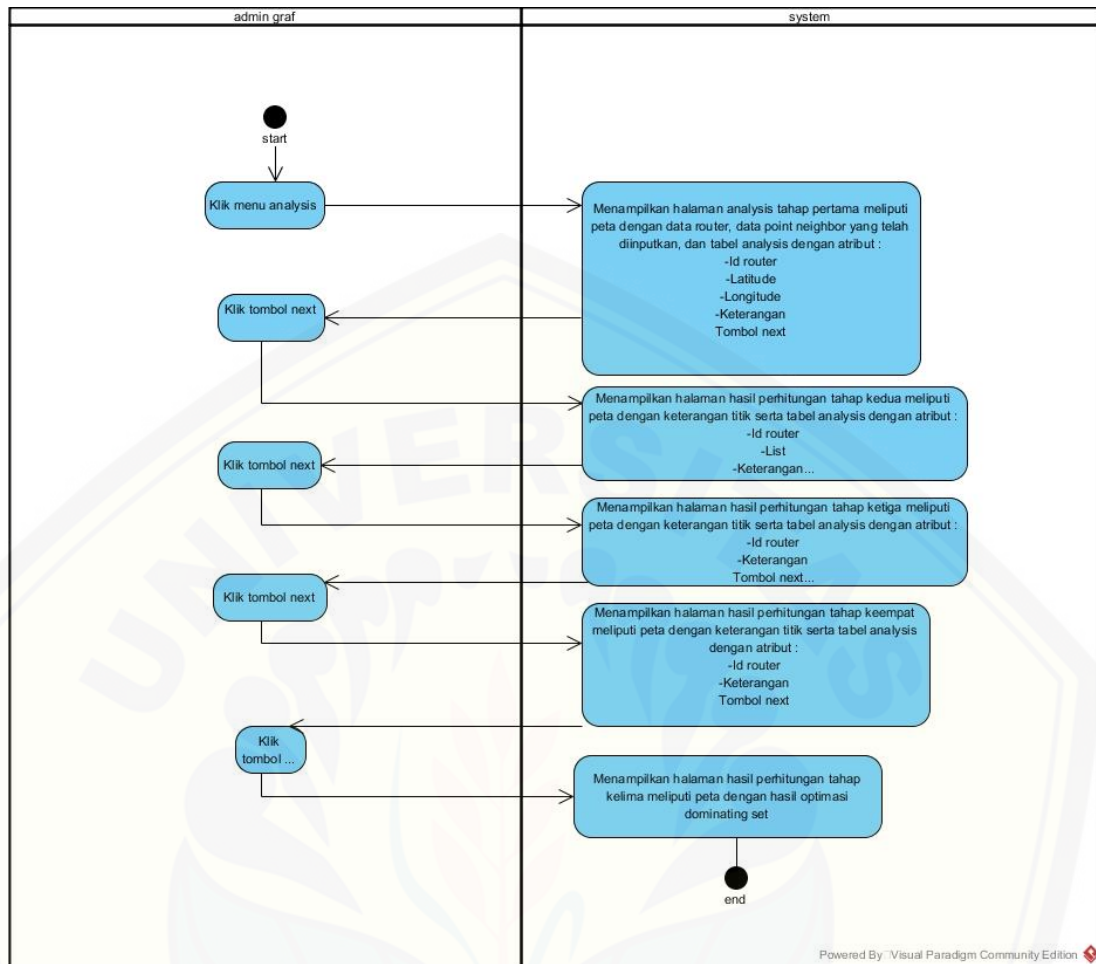
Gambar B. 9 Pengelolaan *Point Neighbor* Admin Router

B.6 Activity Diagram melihat hasil Perhitungan *Dominating Set* dan algoritma *greedy*

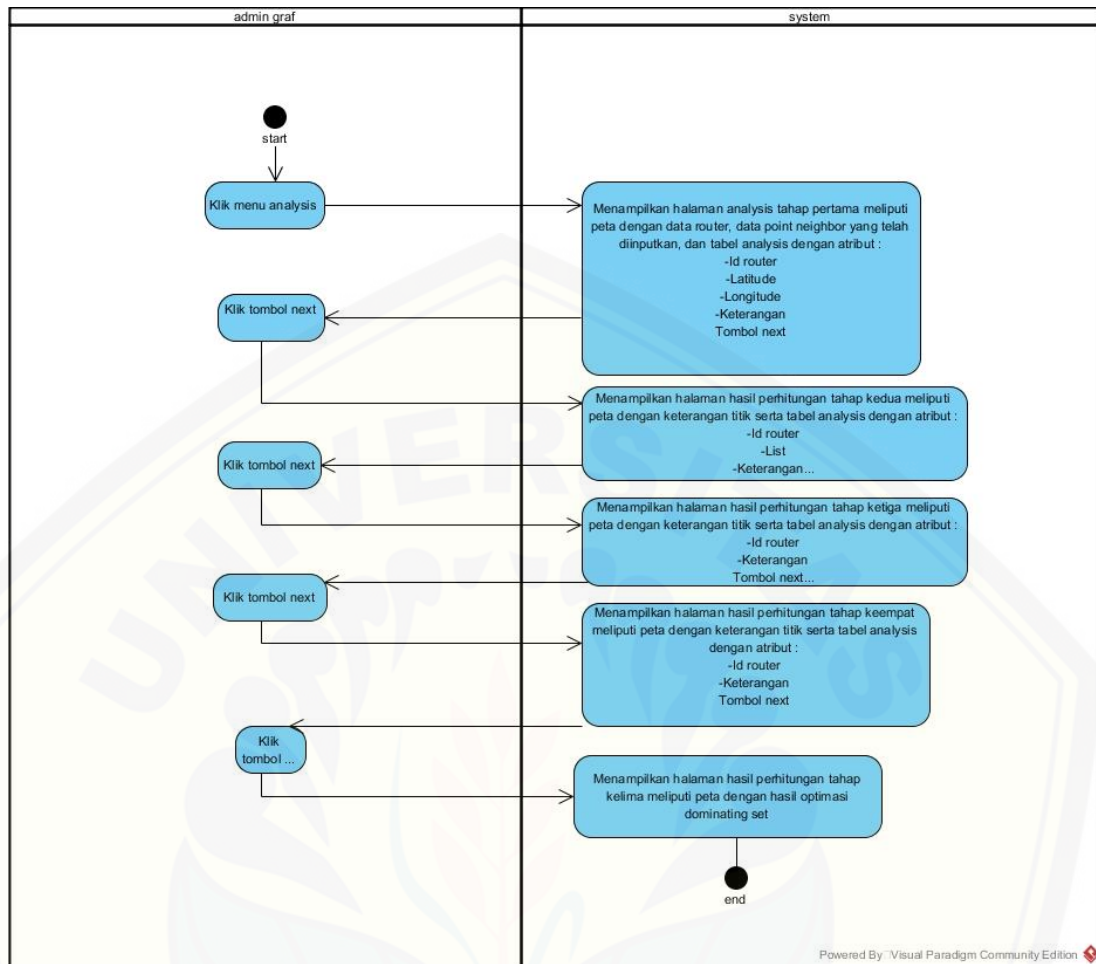
Activity Diagram melihat hasil perhitungan dapat dilihat pada gambar B.10 sampai B.12.



Gambar B. 10 Melihat Hasil Perhitungan Super Admin



Gambar B. 11 Melihat Hasil Perhitungan Admin Graf

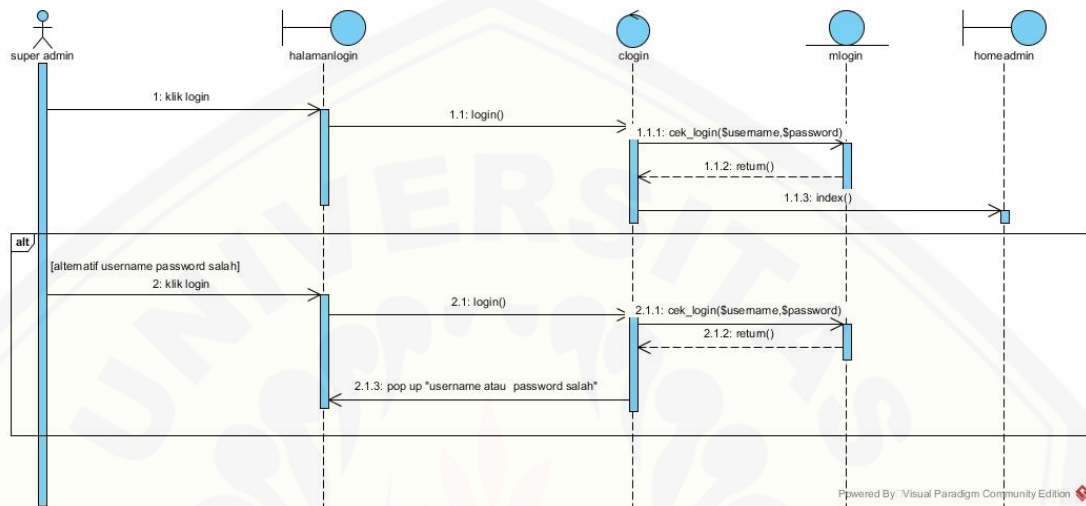


Gambar B. 12 Melihat Hasil Perhitungan Admin Router

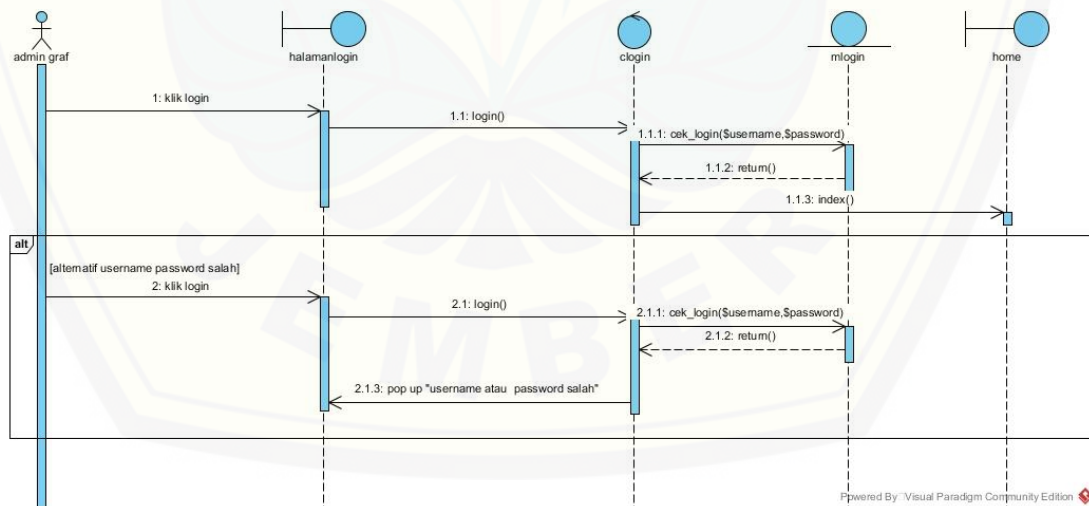
C. Sequence Diagram

C.1 Sequence Diagram Login

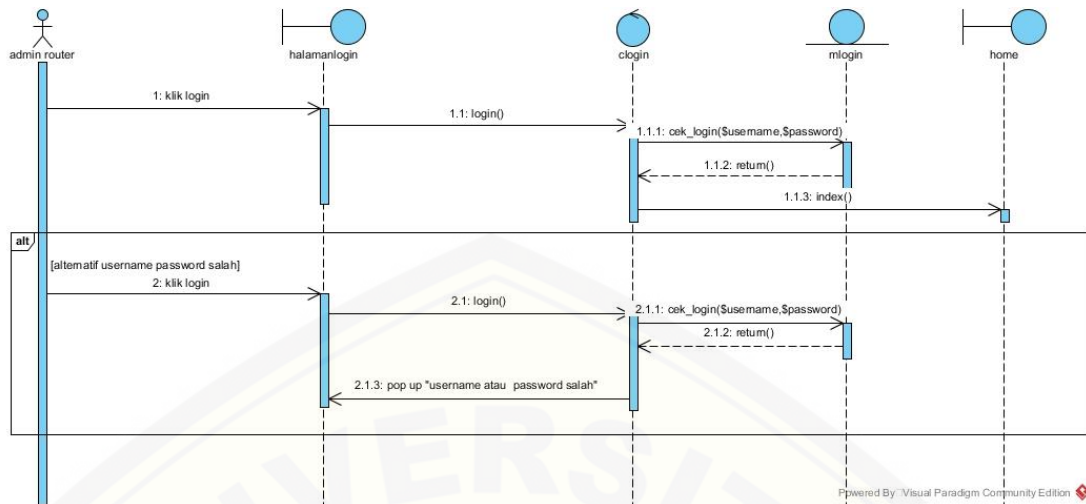
Sequence diagram login dapat dilihat pada gambar C.1 sampai dengan gambar C.3.



Gambar C. 1 Login Super Admin



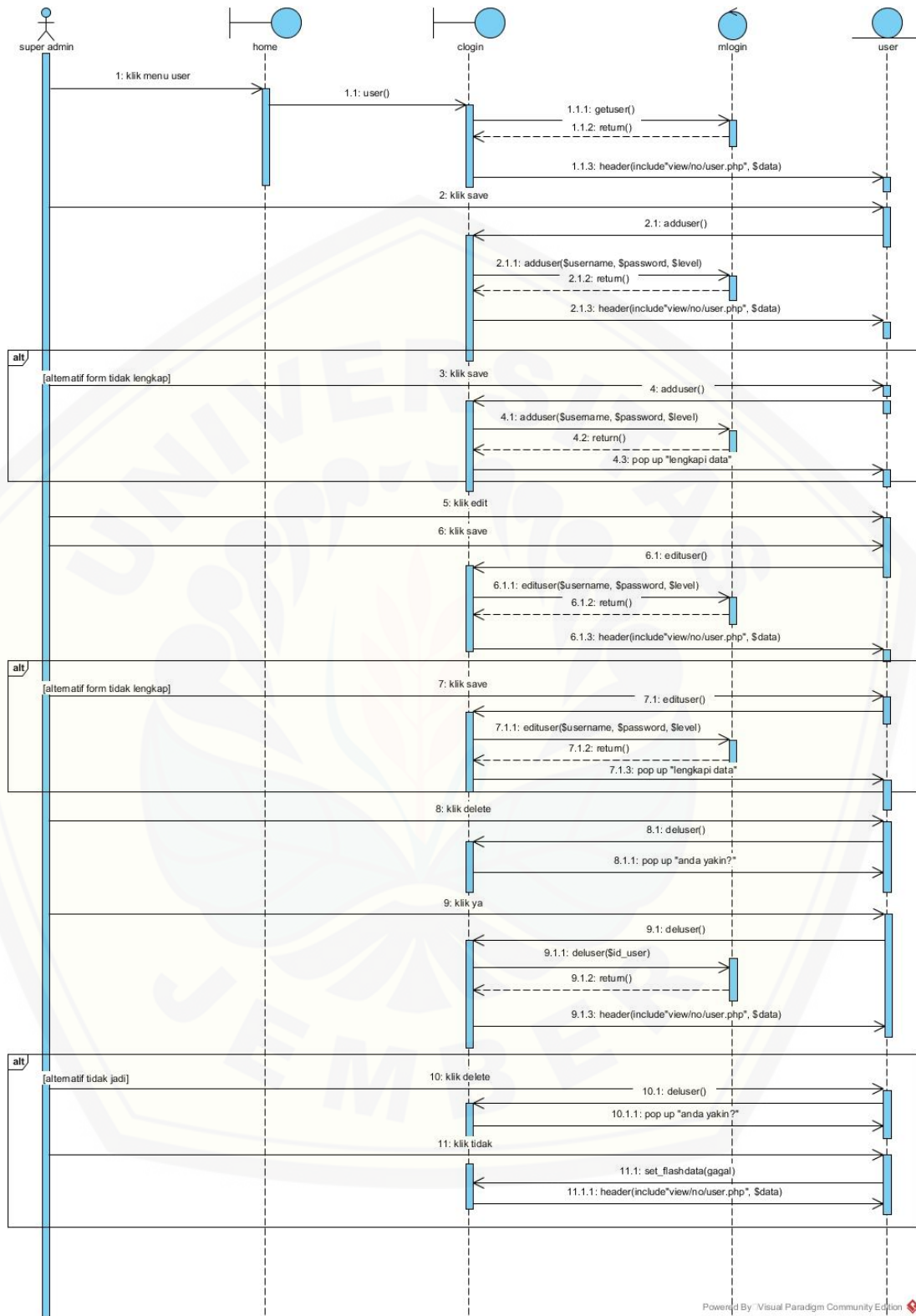
Gambar C. 2 Login Admin Graf



Gambar C. 3 Login Admin Router

C.2 Sequence Diagram Mengelola Data User

Sequence diagram mengelola data user dapat dilihat pada gambar C.4.

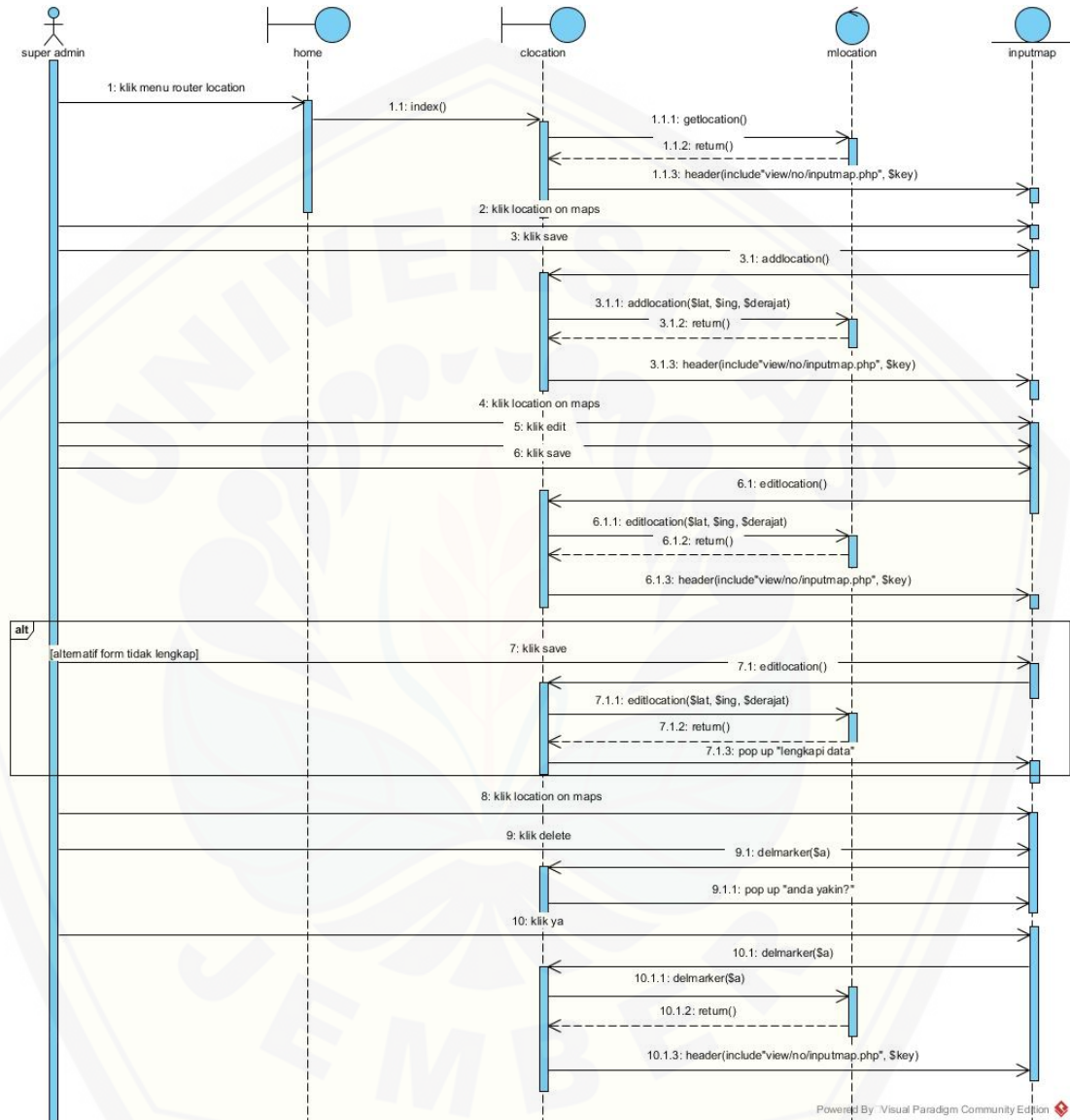


Powered By Visual Paradigm Community Edition

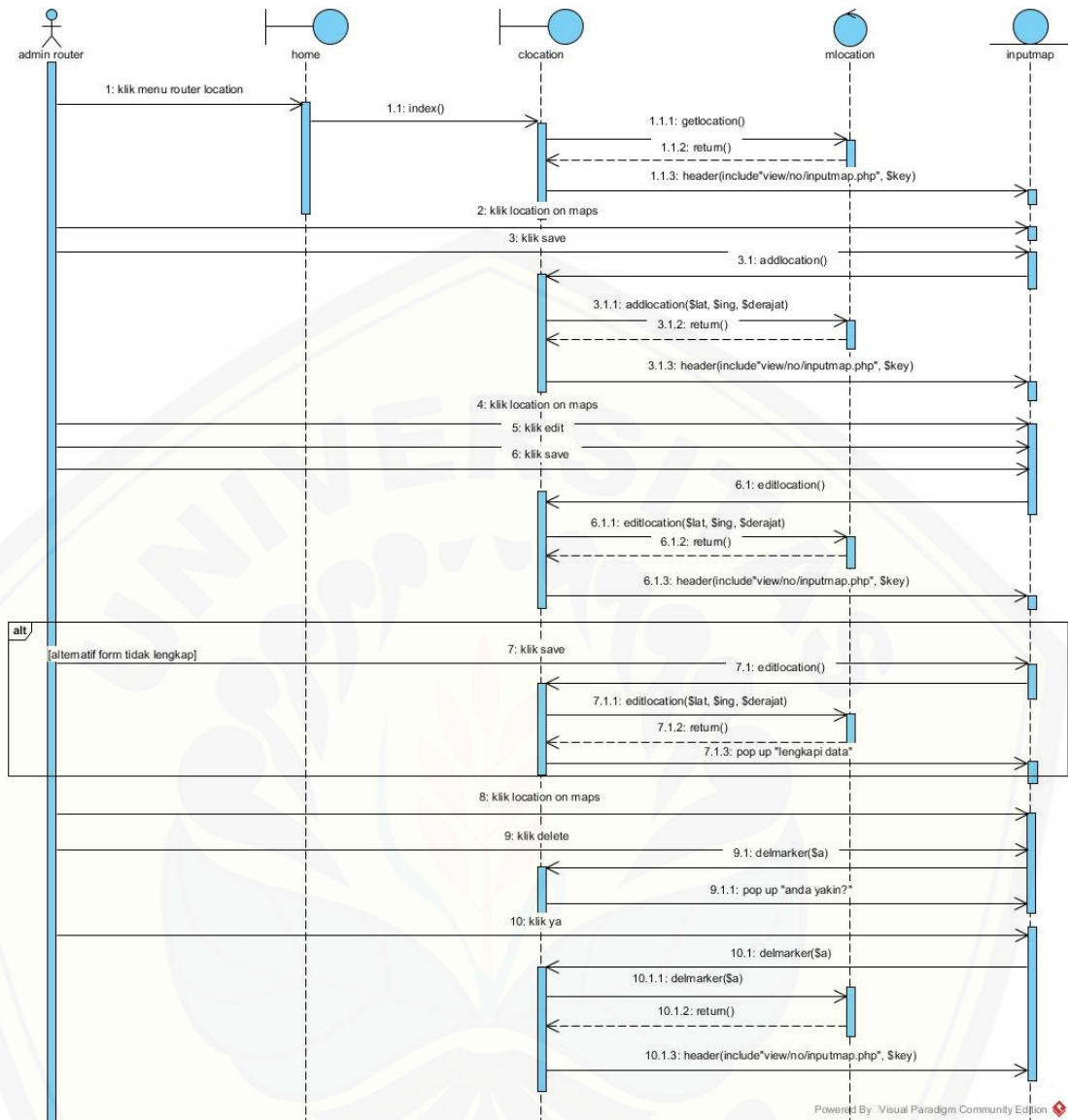
Gambar C. 4 Pengelolaan Data User

C.3 Sequence Diagram Mengelola Data Router

Sequence diagram mengelola data Router dapat dilihat pada gambar C.5 dan C6



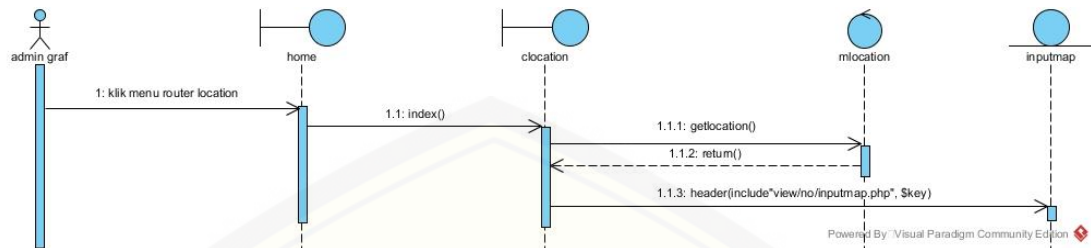
Gambar C. 5 Pengelolaan Data Router Super Admin



Gambar C. 6 Pengelolaan Data Router Admin Router

C.4 Sequence Diagram Melihat Data Router

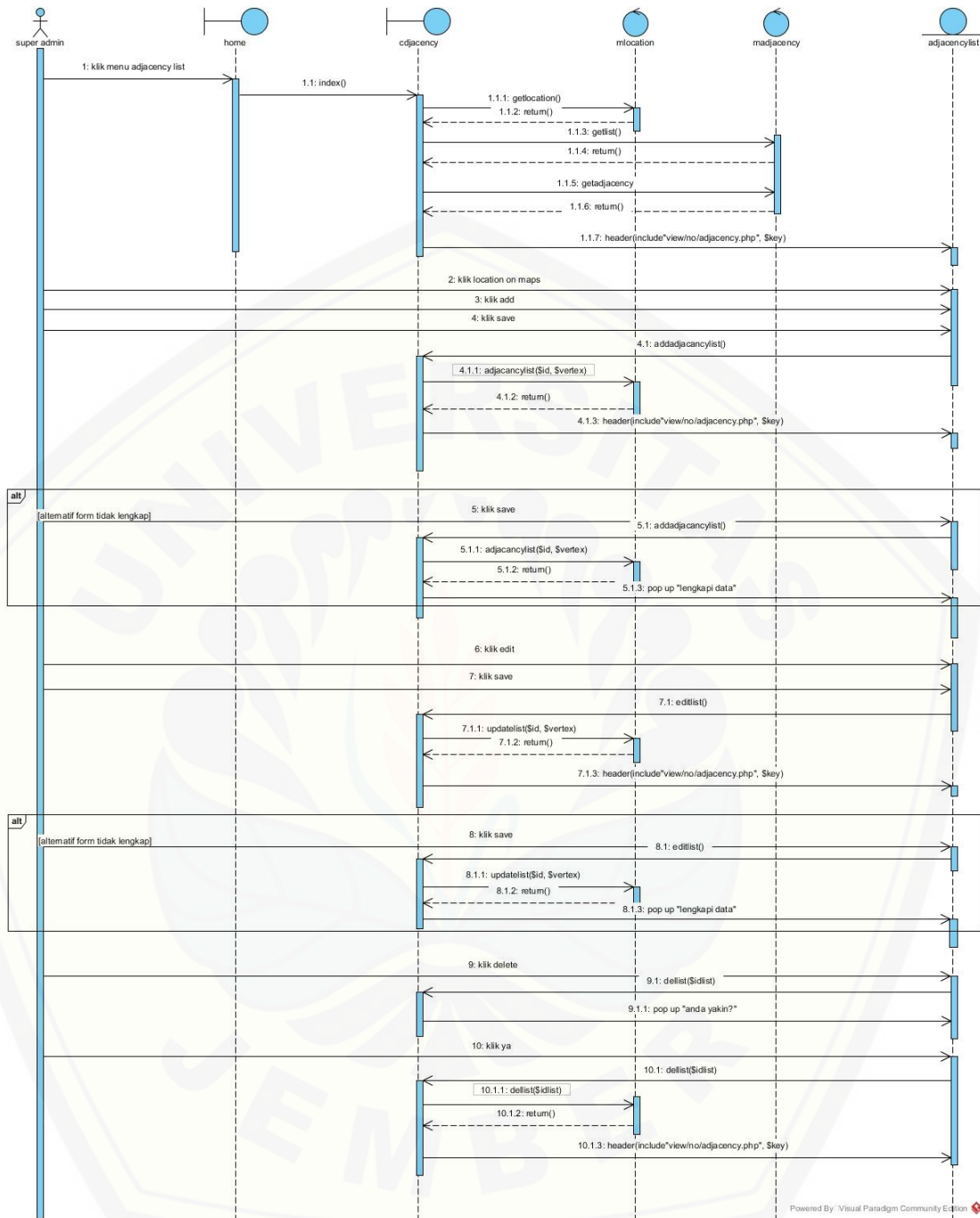
Sequence diagram data melihat data Router dapat dilihat pada gambar C.7.



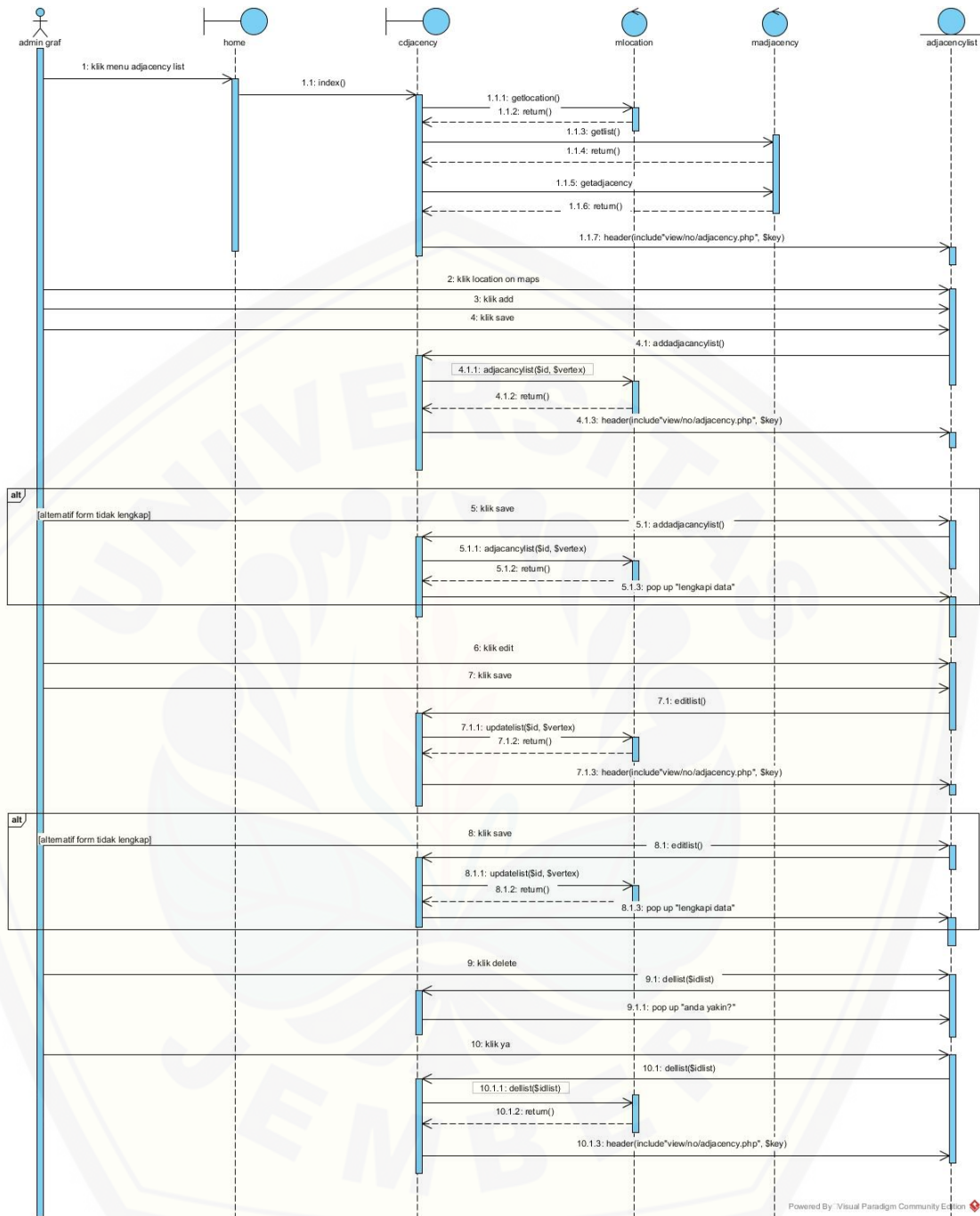
Gambar C. 7 Melihat Data Router

C.5 Sequence Diagram Mengelola Point Neighbor

Sequence diagram mengelola *point neighbor* dapat dilihat pada gambar C.8 dan C.9.



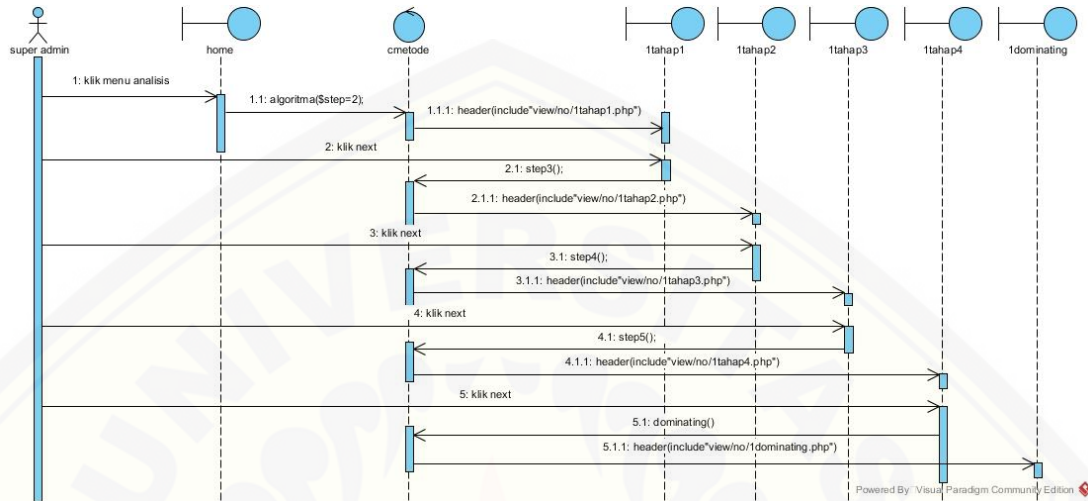
Gambar C. 8 Pengelolaan *Point Neighbor* Super Admin



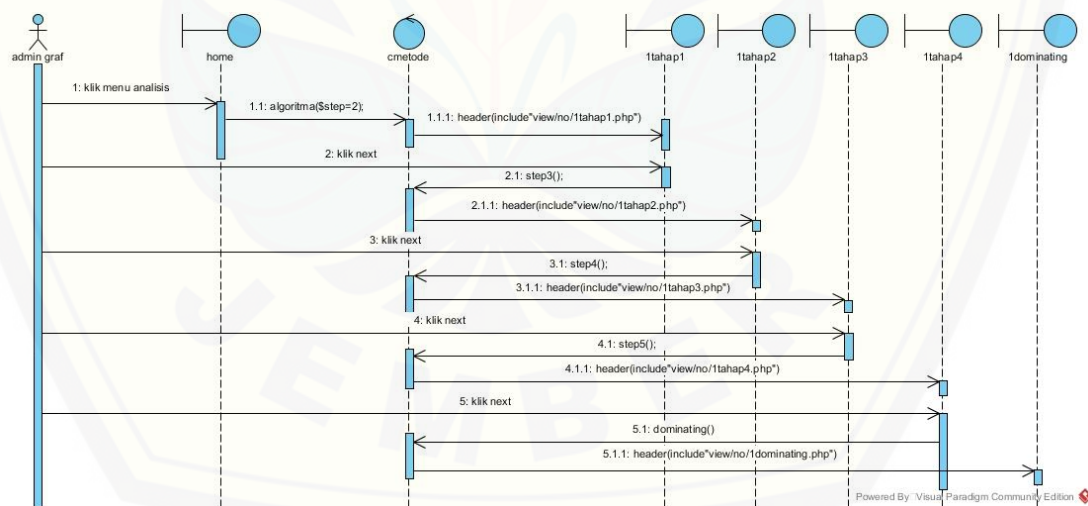
Gambar C. 9 Pengelolaan *Point Neighbor* Admin Graf

C.6 Sequence Diagram Melihat hasil perhitungan

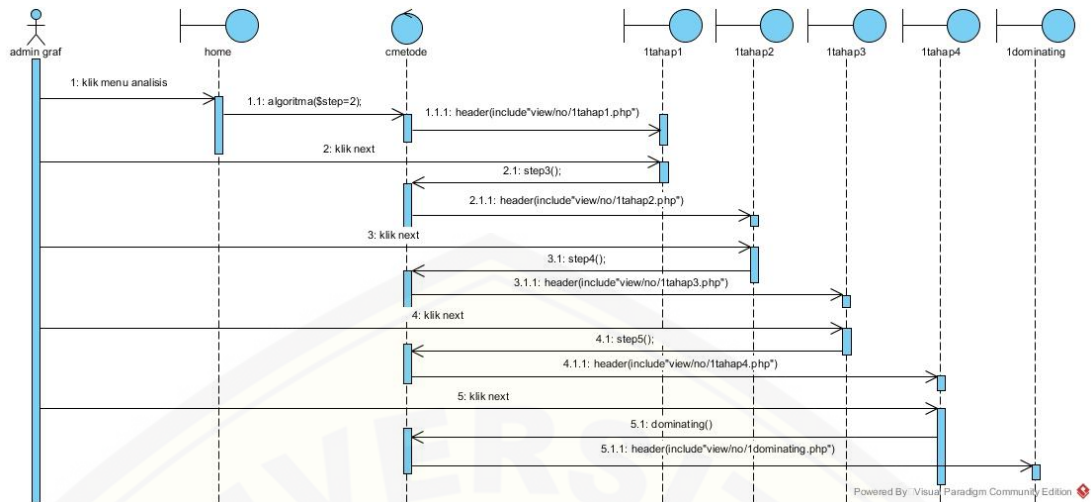
Sequence diagram melihat hasil perhitungan dapat dilihat pada gambar C.9 sampai C.11.



Gambar C. 10 Melihat Hasil Perhitungan Super Admin



Gambar C. 11 Melihat Hasil Perhitungan Admin Graf



Gambar C. 12 Melihat Hasil Perhitungan Admin Router

D. Kode Program

D.1 Kode Program login

Kode program login dapat di lihat pada gambar D.1 dan D.2

```

public function index() {
    if ($this->session->userdata('level') == 'superadmin') {
        $this->load->view('sidebar1');
        $this->load->view('content1');
        $this->load->view('footer1');
    } else if ($this->session->userdata('level') == 'admin router') {
        $this->load->view('sidebar1');
        $this->load->view('content1');
        $this->load->view('footer1');
    } else if ($this->session->userdata('level') == 'admin graf') {
        $this->load->view('sidebar1');
        $this->load->view('content1');
        $this->load->view('footer1');
    }
}

public function login() {
    $username = $this->input->post('username');
    $password = $this->input->post('password');
}
  
```

Gambar D. 1 clogin

```
public function cek_login($username, $password) {
    $query = $this->db->query("SELECT * from tb_login where username = '$username' and password='$password'");
    if ($query->num_rows() > 0) {
        foreach ($query->result_array() as $row) {
            $session_data=array(
                'user_on' => true,
                'id_user' => $row['id_user'],
                'username' => $username,
                'password' => $password,
                'level' => $row['level']
            );
            $this->session->set_userdata($session_data);
            if ($row['level'] == 'superadmin') {
                $this->session->set_userdata('superadmin', TRUE);
            } else if ($row['level'] == 'admin graf') {
                $this->session->set_userdata('admin graf', TRUE);
            } else if ($row['level'] == 'admin router') {
                $this->session->set_userdata('admin router', TRUE);
            }
            return TRUE;
        }
    } else {
        return FALSE;
    }
}
```

Gambar D. 2 mlogin

D.2 Pengolahan Data Router

kode pengolahan data *Router* dapat dilihat pada gambar D.3 sampai D.6

```
public function index() {
    $this->load->view('sidebar1');
    $key['lokasi'] = $this->mLocation->getLocation();
    $this->load->view('inputmap',$key);
    $this->load->view('footer1');
}

public function addLocation() {
    $lat = $this->input->post('lat');
    $lng = $this->input->post('lng');

    $input = $this->mLocation->addLocation($lat, $lng, $derajat);
    if($input==true){
        $this->session->set_flashdata('sukses', 'add success');
        redirect('cLocation');
    } else {
        $this->session->set_flashdata('gagal', 'add failed');
        redirect('cLocation');
    }
}

public function viewEdit($id)
{
    $this->load->view('sidebar1');
    $key['detail'] = $this->mLocation->getLocationID($id);
    $this->load->view('editLocation',$key);
    $this->load->view('footer1');
}

public function editLocation()
{
    $id_router = $this->input->post('id_router');
    $lat = $this->input->post('lat');
    $lng = $this->input->post('lng');
    $input = $this->mLocation->editLocation($id_router, $lat, $lng);
    if($input==true){
        $this->session->set_flashdata('sukses', 'edit success');
        redirect('cLocation');
    } else {
        $this->session->set_flashdata('gagal', 'edit failed');
        redirect('cLocation');
    }
}
}
```

Gambar D. 3 Pengelolaan data *Router clocation*

```
public function delMarker($a)
{
    $delete = $this->mLocation->delMarker($a);
    if ($delete) {
        $this->session->set_flashdata('sukses', 'delete success');

        redirect('cLocation');
    } else {
        $this->session->set_flashdata('gagal', 'delete failed');

        redirect('cLocation');
    }
}
}
```

Gambar D. 4 pengelolaan data *Router clocation*

```
if(!defined('BASEPATH')) exit ('No direct script access allowed');  
class MLocation extends CI_Model{  
  
    public function getLocation()  
    {  
        $query = $this->db->query("select * from tb_router");  
        return $query->result_array();  
    }  
  
    public function addLocation($lat, $lng, $derajat)  
    {  
        $query = $this->db->query("INSERT into tb_router (lat, lng) values ('".$lat."','".$lng."");");  
        return $query;  
    }  
  
    public function getLocationID($id)  
    {  
        $query = $this->db->query("SELECT * from tb_router WHERE id_router = '".$id."'");  
        return $query->result_array();  
    }  
  
    public function editLocation($id_router, $lat, $lng)  
    {  
        $query = $this->db->query("UPDATE `tb_router` SET `id_router`='".$id_router."', `lat`='".$lat."', `lng`='".$lng."' WHERE id_router = '".$id_router."'");  
        return $query;  
    }  
  
    public function delMarker($a) {  
        $query = $this->db->query("DELETE FROM tb_router WHERE id_router = '".$a."'");  
        if ($query) {  
            return TRUE;  
        } else {  
            return FALSE;  
        }  
    }  
}
```

Gambar D. 5 pengolahan data *Router mlocation*

D.3 Pengolahan data *point neighbor*

Pengolahan data point neighbor dapat dilihat pada gambar D.6 sampai dengan D.8


```

public function index() {
    $this->load->view('sidebar1');
    $key['lokasi'] = $this->mLocation->getLocation();
    $key['list'] = $this->mAdjacency->getList();
    $key['latlng'] = $this->mAdjacency->getLatlng();
    $this->load->view('adjacencyList',$key);
    $this->load->view('footer1');
}

public function detailList($id_router)
{
    $this->load->view('sidebar1');
    $key['lokasi'] = $this->mLocation->getLocation();
    $key['list'] = $this->mAdjacency->detailList($id_router);
    $this->load->view('detailList',$key);
    $this->load->view('footer1');
}

public function viewEdit($idlist)
{
    $this->load->view('sidebar1');
    $key['lokasi'] = $this->mLocation->getLocation();
    $key['list'] = $this->mAdjacency->getList();
    $key['idlist'] = $this->mAdjacency->getEdit($idlist);
    $this->load->view('editList',$key);
    $this->load->view('footer1');
}

public function adjacencyList()
{
    // $id = $this->input->post('id');
    $id_router = $this->input->post('id_router');
    $list = $this->input->post('list');
    $input = $this->mAdjacency->adjacencyList($id_router, $list);
    if($input==true){
        $this->session->set_flashdata('sukses', 'add success');
        redirect('cAdjacency');
    } else {
        $this->session->set_flashdata('gagal', 'add failed');
        redirect('cAdjacency');
    }
}
}

```

Gambar D. 6 pengelolaan data *point neighbor cAdjacency*

```

public function editList()
{
    $idlist = $this->input->post('idlist');
    $id_router = $this->input->post('id_router');
    $list = $this->input->post('list');
    $lat = $this->input->post('lat');
    $lng = $this->input->post('lng');
    $input = $this->mAdjacency->updateList($idlist, $id_router, $list, $lat, $lng);
    if($input==true){
        $this->session->set_flashdata('sukses', 'edit success');
        redirect('cAdjacency');
    } else {
        $this->session->set_flashdata('gagal', 'edit failed');
        redirect('cAdjacency');
    }
}

public function delList($idlist)
{
    $delete = $this->mAdjacency->delList($idlist);
    if ($delete) {
        $this->session->set_flashdata('sukses', 'delete success');
        redirect('cAdjacency');
    } else {
        $this->session->set_flashdata('gagal', 'delete failed');
        redirect('cAdjacency');
    }
}
}

```

Gambar D. 7 pengelolaan data *point neighbor cAdjacency*

```
public function adjacencyList($id, $vertex)
{
    $query = $this->db->query("INSERT into tb_list (id_router, list) values ('".$id."','".$vertex."");");
    return $query;
}

public function getList()
{
    $query = $this->db->query("select * from tb_list");
    return $query->result_array();
}

public function getLatLong()
{
    $query = $this->db->query("select a.id_router, b.lat, b.lng, a.list, a.lat_list, a.lng_list from tb_list a join tb_router b on (a.id_router=b.id_router)");
    return $query->result_array();
}

public function detailList($id_router)
{
    $query = $this->db->query("select * from tb_router where id_router = '".$id_router.'");");
    return $query->result_array();
}

public function getEdit($idlist)
{
    $query = $this->db->query("select * from tb_list where id_list = '".$idlist.'");");
    return $query->result_array();
}

public function updateList($idlist, $id_router, $list, $lat, $lng)
{
    $query = $this->db->query("UPDATE `tb_list` SET `id_list`='".$idlist."', `id_router`='".$id_router."', `list`='".$list."', `lat_list`='".$lat."', `lng_list`='".$lng.'" WHERE id_list = '".$idlist.'");");
    return $query;
}

public function delList($idlist)
{
    $query = $this->db->query("DELETE FROM `tb_list` WHERE id_list = '".$idlist.'");");
    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}
}
```

Gambar D. 8 pengolahan data *point neighbor mAdjacency*

D.4 Melihat hasil Perhitungan

Kode program melihat hasil perhitungan dapat dilihat pada gambar D.9 sampai dengan D.11

```

public function index()
{
    $list_titik = $this->a->sortDerajat();
    $list_tujuan = array(); //node yg sudah di cover
    $tujuan_gagal = array();
    #1 Menyeleksi titik dan tujuan berdasarkan titik dgn derajat terbesar
    foreach ($list_titik as $l)
    {
        $tujuan = $this->a->get_tujuan($l["id_router"]);

        $available = false;
        foreach ($list_tujuan as $lt)
        {
            foreach ($tujuan as $t)
            {
                // echo $lt["tujuan"] . " | " . $t["titik"] . " | " . $t["tujuan"] . "<br />";
                if ($lt["list"] == $t["id_router"] || $lt["list"] == $t["list"])
                {
                    $available = true;
                }
            }
        }

        if (!$available)
        {
            $sanu = "";
            foreach ($tujuan as $t)
            {
                array_push($list_tujuan, array('id_router' => $t["id_router"], "list" => $t["list"]));
                $sanu .= $t["list"].",";
                $this->dominatorSamaYangTerpakai[] = [
                    'jenis' => 'terpakai',
                    'isi' => $this->a->findVertex($t["list"])[0],
                    'dominator' => $l["id_router"]
                ];
            }
            $this->dominatorSamaYangTerpakai[] = [
                'jenis' => 'dominator',
                'isi' => $l,
                'list' => $sanu
            ];
        }
    }
}

```

Gambar D. 9 Melihat hasil perhitungan *cmetode*

```

private function step2()
{
    $this->index();
    $data['nodes'] = $this->dominatorSamaYangTerpakai;
    $this->load->view('sidebar1');
    $this->load->view('tahap1', $data);
    $this->load->view('footer1');
}

private function step3()
{
    $this->index();
    $hmm = array_merge($this->dominatorSamaYangTerpakai, $this->listTidak);
    $data['nodes'] = $hmm;
    $this->load->view('sidebar1');
    $this->load->view('tahap2', $data);
    $this->load->view('footer1');
}

private function step4()
{
    $this->index();
    $hmm = array_merge($this->dominatorSamaYangTerpakai, $this->sisane, $this->jomblo);
    $data['nodes'] = $hmm;
    $this->load->view('sidebar1');
    $this->load->view('tahap3', $data);
    $this->load->view('footer1');
}

public function step5()
{
    $this->index();
    $hmm = array_merge($this->dominatorSamaYangTerpakai, $this->sisane, $this->jomblo);
    $data['nodes'] = $hmm;
    $this->load->view('sidebar1');
    $this->load->view('tahap4', $data);
    $this->load->view('footer1');
}

public function dominating(){
    $this->load->library('../controllers/cMetode');
    $this->index();
    $hmm = array_merge($this->dominatorSamaYangTerpakai, $this->sisane, $this->jomblo);
    $data['nodes'] = $hmm;
    $list_titik = $this->a->sortDerajat();
    $p = $this->a->get_semua_titik();
    $g = $list_titik[0]['jmlDerajat'];
}

```

Gambar D. 10 melihat hasil perhitungan *cmetode*

```

public function sortDerajat() // untuk mengurutkan derajat dari besar ke kecil
{
    $query = $this->db->query("SELECT l.id_router, COUNT(l.id_router) as jmlderajat, s.lat, s.lng FROM `tb_list` l join tb_router s on(l.id_router=s
    .id_router) group by l.id_router order by jmlderajat desc, id_router ASC")->result_array();
    return $query;
}

function findvertex($id_router) // untuk mengambil titik
{
    return $this->db->query("SELECT `id_router`, `lng`, `lat` FROM tb_router WHERE id_router = $id_router")->result_array();
}

function get_tujuan($id_router) // untuk memanggil tetangga dari titik
{
    return $this->db->query("SELECT `id_router`, `list` FROM tb_list WHERE id_router = $id_router ORDER BY id_router ASC")->result_array();
}

public function get_semua_titik()
{
    return $this->db->query("SELECT COUNT(id_router) as jumlah from tb_router")->result_array()[0]['jumlah'];
}
}

```

Gambar D. 11 Melihat hasil perhitungan *mmedode*

D.5 Pengelolaan data user

Kode program pengelolaan data user dapat dilihat pada gambar D.12 sampai dengan D.

```

public function addUser()
{
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $level = $this->input->post('level');

    $input = $this->mlogin->addUser($username, $password, $level);
    if($input==true){
        $this->session->set_flashdata('sukses');
        redirect('clogin/user');
    }else{
        $this->session->set_flashdata('gagal');
        redirect('clogin/user');
    }
}

public function editUser()
{
    $id_user = $this->input->post('id_user');
    $username = $this->input->post('username');
    $password = $this->input->post('password');
    $level = $this->input->post('level');

    $update = $this->mlogin->editUser($username, $password, $level, $id_user);
    if($update==true){
        $this->session->set_flashdata('sukses');
        redirect('clogin/user');
    }else{
        $this->session->set_flashdata('gagal');
        redirect('clogin/user');
    }
}

public function delUser($id_user)
{
    $delete = $this->mlogin->delUser($id_user);
    if($delete==true){
        $this->session->set_flashdata('sukses');
        redirect('clogin/user');
    }else{
        $this->session->set_flashdata('gagal');
        redirect('clogin/user');
    }
}
}

```

Gambar D. 12 pengelolaan data user *clogin*

```

public function addUser($username, $password, $level)
{
    $query = $this->db->query("INSERT into tb_login (username, password, level) values ('".$username."','".$password."','".$level."')");
    if ($query) {
        return TRUE;
    } else {
        return FALSE;
    }
}

public function getUser()
{
    $query = $this->db->query("SELECT * from tb_login");
    return $query->result_array();
}

public function getID($id_user)
{
    $query = $this->db->query("SELECT * from tb_login where id_user = ".$id_user." ");
    return $query->result_array();
}

public function editUser($username, $password, $level, $id_user)
{
    $query = $this->db->query("UPDATE tb_login set username = '".$username."', password='".$password."', level='".$level.'" where id_user = ".$id_user." ");
    return $query;
}

public function delUser($id_user)
{
    $query = $this->db->query("DELETE from tb_login where id_user = ".$id_user." ");
    return $query;
}

```

Gambar D. 13 Pengelolaan data *user mlogin*

D.6 Melihat data *Router*

Kode program melihat data *Router* dapat dilihat pada gambar D.14 sampai dengan D.15

```

public function index() {
    $this->load->view('sidebar1');
    $key['lokasi'] = $this->mLocation->getLocation();
    $this->load->view('inputmap',$key);
    $this->load->view('footer1');
}

public function addLocation() {
    $lat = $this->input->post('lat');
    $lng = $this->input->post('lng');

    $input = $this->mLocation->addLocation($lat, $lng, $derajat);
    if($input==true){
        $this->session->set_flashdata('sukses', 'add success');
        redirect('cLocation');
    } else {
        $this->session->set_flashdata('gagal', 'add failed');
        redirect('cLocation');
    }
}

public function viewEdit($id)
{
    $this->load->view('sidebar1');
    $key['detail'] = $this->mLocation->getLocationID($id);
    $this->load->view('editLocation',$key);
    $this->load->view('footer1');
}

public function editLocation()
{
    $id_router = $this->input->post('id_router');
    $lat = $this->input->post('lat');
    $lng = $this->input->post('lng');
    $input = $this->mLocation->editLocation($id_router, $lat, $lng);
    if($input==true){
        $this->session->set_flashdata('sukses', 'edit success');
        redirect('cLocation');
    } else {
        $this->session->set_flashdata('gagal', 'edit failed');
        redirect('cLocation');
    }
}

```

Gambar D. 14 Melihat data *Router*

```
if(!defined('BASEPATH')) exit ('No direct script access allowed');  
class MLocation extends CI_Model{  
  
    public function getLocation()  
    {  
        $query = $this->db->query("select * from tb_router");  
        return $query->result_array();  
    }  
  
    public function addLocation($lat, $lng, $derajat)  
    {  
        $query = $this->db->query("INSERT into tb_router (lat, lng) values ('".$lat."','".$lng."')");  
        return $query;  
    }  
  
    public function getLocationID($id)  
    {  
        $query = $this->db->query("SELECT * from tb_router WHERE id_router = '".$id."'");  
        return $query->result_array();  
    }  
  
    public function editLocation($id_router, $lat, $lng)  
    {  
        $query = $this->db->query("UPDATE `tb_router` SET `id_router`='".$id_router."',`lat`='".$lat."',`lng`='".$lng."' WHERE id_router = '".$id_router."'");  
        return $query;  
    }  
  
    public function delMarker($a) {  
        $query = $this->db->query("DELETE FROM tb_router WHERE id_router = '".$a."'");  
        if ($query) {  
            return TRUE;  
        } else {  
            return FALSE;  
        }  
    }  
}
```

Gambar D. 15 Melihat data *Router*