



**IMPLEMENTASI ALGORITMA *GREEDY* PADA PENENTUAN
DOMINATING SET (STUDI KASUS: PENENTUAN SUMUR RESAPAN DI
UNIVERSITAS JEMBER)**

SKRIPSI

Oleh

Ainul Ulfah Maulinda

NIM 122410101077

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016



**IMPLEMENTASI ALGORITMA *GREEDY* PADA PENENTUAN
DOMINATING SET (STUDI KASUS: PENENTUAN SUMUR RESAPAN DI
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

Ainul Ulfah Maulinda

NIM 122410101077

PROGRAM STUDI SISTEM INFORMASI

UNIVERSITAS JEMBER

2016

PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Ayahanda Yoyok Koeswoyo dan Ibunda Latifah Handayani tercinta;
2. Adik perempuan satu-satunya Dania Safira Agustin;
3. Kakek H. Hasyim dan kakek Miskan beserta keluarga besar;
4. Sahabatku bersama dukungan dan doanya;
5. Guru-guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
6. Almamater Program Studi Sistem Informasi Universitas Jember.

MOTO

“So be patient with a beautiful patience”. (QS. 70: 11)¹



¹ Kementerian Agama Republik Indonesia. 2014. *Ummul Mukminin : Al Qur'an dan Terjemahannya untuk Wanita*. Jakarta: OASIS Terrace Resident

PERNYATAAN

Saya yang bertanda tangan di bawah ini:

nama : Ainul Ulfah Maulinda

NIM : 122410101077

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Implementasi Algoritma *Greedy* Pada Penentuan *Dominating set* (Studi Kasus: Penentuan Sumur Resapan Di 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, Oktober 2016

Yang menyatakan,

Ainul Ulfah Maulinda

NIM 122410101077

SKRIPSI

**IMPLEMENTASI ALGORITMA *GREEDY* PADA PENENTUAN
DOMINATING SET (STUDI KASUS: PENENTUAN SUMUR RESAPAN DI
UNIVERSITAS JEMBER)**

Oleh

Ainul Ulfah Maulinda

NIM 122410101077

Pembimbing :

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

Dosen Pembimbing Pemdamping : Yanuar Nurdiansyah, ST., M.Cs.

PENGESAHAN PEMBIMBING

Skripsi berjudul “Implementasi Algoritma *Greedy* Pada Penentuan *Dominating set* (Studi Kasus: Penentuan Sumur Resapan Di Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal : jumat, 28 Oktober 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Disetujui oleh:

Pembimbing I,

Pembimbing II,

Prof. Slamim, M.Comp.Sc.,Ph.D.

NIP 19670420 1992011001

Yanuar Nurdiansyah, ST., M.Cs.

NIP 19820101 2010121004

PENGESAHAN

Skripsi berjudul “Implementasi Algoritma *Greedy* Pada Penentuan *Dominating set* (Studi Kasus: Penentuan Sumur Resapan Di Universitas Jember)”, telah diuji dan disahkan pada:

hari, tanggal : Jum’at, 28 Oktober 2016

tempat : Program Studi Sistem Informasi Universitas Jember

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST., MT
NIP 196906151997021002

Oktalia Juwita, S.Kom., M.MT
NIP 198110202014042001

Mengesahkan
Ketua Program Studi,

Prof. Drs. Slamir, M.Comp.Sc.,Ph.D
NIP. 19670420 1992011001

RINGKASAN

Implementasi Algoritma *Greedy* Pada Penentuan *Dominating set* (Studi Kasus: Penentuan Sumur Resapan Di Universitas Jember)

Perkembangan zaman tidak terlepas dari perkembangan sains dan teknologi. Salah satu bidang yang berkaitan dengan sains dan teknologi adalah matematika. Saat ini banyak permasalahan yang dapat diselesaikan menggunakan teori matematika. Salah satu teori matematika yang digunakan untuk memecahkan masalah sehari-hari adalah teori graf.

Dalam Penelitian ini teori graf digunakan untuk mengetahui titik pendominasi pada graf sumur resapan di Universitas Jember. Pada graf sumur resapan, *node* yang dimiliki ada representasi dari letak sumur resapan di Universitas Jember dan jarak antar sumur resapan di representasikan sebagai *vertex* yang menghubungkan antar sumur. Peletakan *node* dan *vertex* pada graf sumur resapan berdasarkan graf grid, dimana tiap node dihubungkan dengan *node* lain berdasarkan sisi lahan yang berhimpitan.

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 jumlah himpunan pendominasi dari graf sumur resapan dan lokasi sumur resapan yang ditampilkan pada peta.

PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Implementasi Algoritma *Greedy* Pada Penentuan *Dominating set* (Studi Kasus: Penentuan Sumur Resapan Di 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 terima kasih kepada:

1. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Ketua Program Studi Sistem Informasi Universitas Jember;
2. Prof. Drs. Slamini, M.Comp.Sc., Ph.D., selaku Dosen Pembimbing Utama dan Yanuar Nurdiansyah, ST., M.Cs., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Windi Eka Yulia Retnani, S.Kom., MT., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa.
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Universitas Jember;
5. Ayahanda Yoyok Koeswoyo dan Ibunda Latifah Handayani yang selalu mendukung dan mendoakan;
6. Adik perempuan Dania Safira Agustin;
7. Kakek H. Hasyim dan kakek Miskan beserta keluarga besar;
8. sahabat seperjuangan yang selalu menemani dan memberikan semangat serta doa Siti Nurhidayatul Mahmuda, Diah Ayu Sukmawati, Sofia Wardani, Dhasa Yuniar Maharani, Ratna Suryani, Nindi Norya Tindra, Rizki Vadilla, Affan Taruna Santoso, Brelyanes Zully Fambudi;

9. teman-teman TUJUHDEVELOPER yang memberikan pelajaran dan pengalaman yang luar biasa Ainul yaqin, Marceli Aditya, Aji Mukti, Haris Arfan, Rizki Vadilla, M. Huda, Rizki Herdatullah, dan Nindi Norya;
10. teman-teman seperjuanganku FORMATION angkatan 2012 dan semua mahasiswa Program Studi Sistem Informasi yang telah menjadi keluarga kecil bagi penulis selama menempuh pendidikan S1;
11. keluarga besar Himpunan Mahasiswa Sistem Informasi (HIMASIF) Periode 2013-2014;
12. keluarga besar Binary 2, 3, dan 4;
13. keluarga besar Koperasi Mahasiswa Sistem Informasi (KOPMASI) periode 2012-2013, 2013-2014, dan 2014-2015;
14. teman kontrakan jalan jawa 7 yang telah menjadi rumah singgah dan keluarga penulis selama di Jember;
15. semua pihak yang tidak dapat disebutkan satu-persatu.

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

Jember, 28 Oktober 2016

Penulis

DAFTAR ISI

SKRIPSI.....	i
SKRIPSI.....	ii
PERSEMBAHAN.....	iii
MOTO.....	iv
PERNYATAAN.....	v
SKRIPSI.....	vi
PENGESAHAN PEMBIMBING.....	vii
PENGESAHAN.....	viii
RINGKASAN.....	ix
PRAKATA.....	x
DAFTAR ISI.....	xii
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
BAB 1. PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan dan Manfaat.....	3
1.3.1 Tujuan.....	3
1.3.2 Manfaat.....	4
1.4 Batasan Masalah.....	4
1.5 Sistematika Penulisan.....	4

BAB 2.	TINJAUAN PUSTAKA	6
2.1	Penelitian Terdahulu	6
2.2	Algoritma <i>Greedy</i>	7
2.3	<i>Dominating set</i>	8
2.4	Model <i>Waterfall</i>	10
BAB 3.	METODOLOGI PENELITIAN	11
3.1.	Tahapan Penelitian	11
3.1.1	Tahapan Analisis Kebutuhan	11
3.1.2	Tahapan Desain Sistem	14
3.1.3	Tahapan Implementasi Sistem	15
3.1.4	Tahapan Pengujian Sistem	15
3.1.5	Tahapan Pemeliharaan Sistem	16
BAB 4.	PENGEMBANGAN SISTEM.....	17
3.1	Analisis Kebutuhan Sistem	17
3.1.1	Kebutuhan Fungsional	17
3.1.2	Kebutuhan non-fungsional.....	18
3.2	Desain Sistem.....	18
3.2.1	<i>Business Process</i>	18
3.2.2	<i>Use Case Diagram</i>	19
3.2.3	<i>Use Case Scenario</i>	22
3.2.4	<i>Sequence Diagram</i>	27
3.2.5	<i>Activity Diagram</i>	28
3.2.6	<i>Class Diagram</i>	33

3.2.7	<i>Entity Relationship Diagram (ERD)</i>	34
3.3	Pengkodean Sistem.....	34
3.3.1	Kode Program <i>Login</i>	34
3.3.2	Kode Program Pengelolaan Data <i>User</i>	35
3.3.3	Kode Program <i>View</i> Data Sumur.....	35
3.3.4	Kode Program Pengelolaan Data Sumur.....	35
3.3.5	Kode Program Pengelolaan Data <i>Adjacency List</i>	35
3.3.6	Kode Program <i>View</i> Hasil Perhitungan Algoritma <i>Greedy</i> dan <i>Dominating set</i>	35
3.4	Pengujian Sistem.....	40
4.5.1	Metode <i>White Box</i>	41
4.5.2	Metode <i>Black Box</i>	42
BAB 5.	HASIL DAN PEMBAHASAN.....	50
4.1	Hasil Implementasi <i>Coding</i> pada Sistem Informasi Penentu <i>Dominating set</i> Pada Graf Sumur Resapan di Universitas Jember.....	50
4.1.1	Tampilan Halaman <i>Login</i>	50
4.1.2	Tampilan Halaman <i>Dashboard</i>	51
4.1.3	Tampilan Halaman <i>Wells Location</i>	52
4.1.4	Tampilan Halaman <i>Adjacency List</i>	55
4.1.5	Tampilan Halaman <i>Analysis</i>	56
4.1.6	Tampilan Halaman <i>User Management</i>	58
4.2	Hasil Implementasi Algoritma <i>Greedy</i> Pada Sistem Informasi Penentuan <i>Dominating set</i> (Studi kasus: Penentuan Sumur Resapan di Universitas Jember).....	60

4.3	Pengujian Analisis Sistem Informasi Penentu <i>Dominating set</i> pada Sumur Resapan di Universitas Jember	64
4.4	Pembahasan.....	71
4.4.1	Pembahasan Hasil Implementasi Algoritma <i>Greedy</i> Pada Sistem Informasi Penentuan <i>Dominating set</i> (Studi kasus: Penentuan Sumur Resapan di Universitas Jember).....	71
4.4.2	Pembahasan Hasil Pengujian Sistem	73
BAB 6.	PENUTUP	75
6.1	Kesimpulan	75
6.2	Saran	76
DAFTAR PUSTAKA	77
LAMPIRAN A	79
A.1	Skenario Login	79
A.2	Skenario Pengelolaan Data User	82
A.3	Skenario Pengelolaan Data Sumur	85
A.4	Skenarion <i>View</i> Data Sumur	92
A.5	Skenario Pengelolaan Data Adjacency List	93
LAMPIRAN B	99
B.1	<i>Sequence Diagram</i> Login	99
B.2	<i>Sequence Diagram</i> Pengelolaan Data <i>User</i>	101
B.3	<i>Sequence Diagram</i> <i>view</i> Data Sumur	102
B.4	<i>Sequence Diagram</i> Pengelolaan Data Sumur	103
B.5	<i>Sequence Diagram</i> Pengelolaan Data <i>Adjacency List</i>	105
LAMPIRAN C	106

C.1 Activity Diagram Login	106
C.2 Activity Diagram Pengelolaan Data User.....	107
C.3 Activity Diagram View Data Sumur	108
C.4 Activity Diagram Pengelolaan Data Sumur.....	108
C.5 Activity Diagram Pengelolaan Data Adjacency List.....	110
LAMPIRAN D	111
D.1 Kode Program Login	111
D.2 Kode Program Pengelolaan Data User.....	113
D.3 Kode Program View Data Sumur	114
D.4 Kode Program Pengelolaan Data Sumur.....	114
D.5 Kode Program Pengelolaan Data Adjacency List.....	116
LAMPIRAN E	119
E.1 Kelas Controller	119

DAFTAR GAMBAR

Gambar 2. 1 Contoh dari <i>Greedy Algorithm</i> (Alfarisi, 2014)	8
Gambar 2. 2 Graf G dengan himpunan pendominasinya { 1, 3, 5 } (Iswadi, 2011).....	9
Gambar 2. 3 Model <i>Waterfall</i> (Sommerville, 2003).....	10
Gambar 3. 1 Diagram Alir Proses Perhitungan Algoritma <i>Greedy</i> dan <i>Dominating set</i>	13
Gambar 4. 1 <i>Business Process</i>	19
Gambar 4. 2 <i>Use Case Diagram</i>	20
Gambar 4. 3 <i>Sequence Diagram view</i> Hasil Perhitungan algoritma <i>greedy</i> dan <i>dominating set</i>	28
Gambar 4. 4 <i>Activity Diagram view</i> hasil <i>dominating set</i> dan <i>greedy</i> super admin ..	30
Gambar 4. 5 <i>Activity Diagram view</i> hasil <i>dominating set</i> dan <i>greedy</i> admin graf	31
Gambar 4. 6 <i>Activity Diagram view</i> hasil <i>dominating set</i> dan <i>greedy</i> admin sumur .	32
Gambar 4. 7 <i>Class Diagram</i> Aplikasi Penentu Sumur Resapan di Universitas Jember	33
Gambar 4. 8 ERD Aplikasi Penentu Letak Sumur Resapan di Universitas Jember ..	34
Gambar 4. 9 Kode program algoritma <i>greedy</i>	39
Gambar 4. 10 Kode <i>function dominating</i>	39
Gambar 4. 11 Penulisan Kode Program mMetode.....	40
Gambar 4. 12 Listing Program.....	41
Gambar 4. 13 Diagram alir.....	41
Gambar 5. 1 Tampilan Halaman <i>Login</i>	51
Gambar 5. 2 Tampilan Halaman <i>Dashboard</i>	51
Gambar 5. 3 Tampilan Halaman <i>Wells Location</i>	52
Gambar 5. 4 Tampilan Tabel Data Sumur	53
Gambar 5. 5 Tampilan Detail Titik.....	53
Gambar 5. 6 Tampilan <i>Delete</i> Data Sumur	54

Gambar 5. 7 Tampilan <i>edit</i> data sumur.....	54
Gambar 5. 8 Tampilan Tabel Data <i>Adjacency List</i>	55
Gambar 5. 9 Tampilan Halaman <i>Adjacency List</i>	55
Gambar 5. 10 Tampilan edit data <i>Adjacency List</i>	56
Gambar 5. 11 Tampilan Halaman <i>Analysis</i> Tahap 2.....	56
Gambar 5. 12 Tampilan Halaman <i>Analysis</i>	57
Gambar 5. 13 Halaman <i>Analysis</i> Tahap 4.....	57
Gambar 5. 14 Tampilan Halaman <i>Analysis</i> Tahap 3.....	57
Gambar 5. 15 Tampilan <i>Delete Data User</i>	58
Gambar 5. 16 Tampilan Halaman Edit Data <i>User</i>	59
Gambar 5. 17 Tampilan Halaman <i>User Management</i>	58
Gambar 5. 18 Kode Program Tahap 1 Algoritma <i>Greedy</i>	60
Gambar 5. 19 Kode Program tahap 2 Algoritma <i>greedy</i>	61
Gambar 5. 20 Kode Program tahap 2 Algoritma <i>Greedy</i>	61
Gambar 5. 21 Kode Program tahap 3 Algoritma <i>Greedy</i>	62
Gambar 5. 22 Kode Program tahap 4 Algoritma <i>Greedy</i>	62
Gambar 5. 23 Kode Program tahap 5 Algoritma <i>Greedy</i>	63
Gambar 5. 24 Kode Program <i>Dominating set</i>	63
Gambar 5. 25 Graf Data Sumur	64
Gambar 5. 26 Hasil Perhitungan Tahap 1 dan Tahap 2	69
Gambar 5. 27 Hasil Perhitungan Tahap 3	70
Gambar 5. 28 Hasil Perhitungan Tahap 4	70
Gambar 5. 29 Hasil Perhitungan Tahap 5	71

DAFTAR TABEL

Tabel 4. 1 Definisi Aktor	20
Tabel 4. 2 Definisi <i>Use Case</i>	21
Tabel 4. 3 Skenario <i>Use Case View</i> hasil perhitungan <i>dominating set</i> dan <i>greedy</i> Super admin	22
Tabel 4. 4 Skenario <i>Use Case View</i> hasil perhitungan <i>dominating set</i> dan <i>greedy</i> admin graf	24
Tabel 4. 5 Skenario <i>Use Case View</i> hasil perhitungan <i>dominating set</i> dan <i>greedy</i> admin sumur.....	25
Tabel 4. 6 Pengujian <i>Test Case</i> Perhitungan <i>Dominating set</i>	42
Tabel 4. 7 Pengujian <i>Black Box</i>	42
Tabel 5. 1 Tabel hasil perhitungan tahap 1	65
Tabel 5. 2 Hasil Penrhitungan Tahap 2.....	66
Tabel 5. 3 Hasil Perhitungan Tahap 3.....	67
Tabel 5. 4 Hasil Perhitungan Tahap 4.....	68
Tabel 5. 5 Tabel Lokasi Sumur Hasil <i>Dominating set</i>	73

BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, perumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

1.1 Latar Belakang

Perkembangan zaman tidak terlepas dari perkembangan sains dan teknologi. Salah satu bidang yang berkaitan dengan sains dan teknologi adalah matematika. Saat ini banyak permasalahan yang dapat diselesaikan menggunakan teori matematika. Salah satu teori matematika yang digunakan untuk memecahkan masalah sehari-hari adalah teori graf.

Teori graf sebagai salah satu cabang matematika diskrit, teori graf memiliki beragam aplikasi dalam kehidupan sehari-hari. Pada dasarnya graf diaplikasikan untuk memodelkan suatu permasalahan yang dapat dipandang sebagai kumpulan objek dan relasi. Model dari objek berupa representasi titik dan sisi atau sisi yang berarah merupakan representasi dari relasi. Graf dapat digunakan untuk memecahkan berbagai permasalahan yang berkaitan dengan objek diskrit (Alfarisi, 2014).

Dominating set atau himpunan pendominasi merupakan salah satu aplikasi graf yang dapat digunakan untuk memecahkan masalah pada lingkungan di sekeliling kita. Himpunan Dominasi atau disebut dengan *Dominating set* merupakan suatu konsep penentuan suatu titik pada graf dengan ketentuan titik sebagai *Dominating set* menjangkau titik yang ada di sekitarnya dan seminimal mungkin. Masalah *minimum Dominating set* adalah menentukan jumlah minimum titik yang mendominasi dalam sebuah graf (Henning, 2008).

Banyak manfaat *dominating set* dalam kehidupan sehari-hari. Seperti disebutkan oleh (Haynes, 1998) diantaranya dalam rute bus sekolah. Sebagian besar rute bus sekolah beroperasi berdasarkan aturan tertentu. Biasanya aturan tersebut berupaya agar setiap anak berjalan tidak jauh ke tempat pemberhentian bus. Dalam kasus ini permasalahannya di titik-titik mana saja pemberhentian bus ditentukan agar setiap anak berjalan tidak jauh ke pemberhentian tersebut.

Dalam penelitian ini *dominating set* diterapkan pada pencarian titik sumur resapan di Universitas Jember untuk mengetahui sumur resapan mana yang mendominasi sumur resapan lain. Lokasi Universitas Jember di representasikan ke dalam bentuk graf dengan sumur resapan sebagai titik nya. Graf tersebut kemudian dianalisis menggunakan *dominating set* untuk mengetahui letak dan jumlah sumur yang akan dibangun. Sumur resapan digambarkan sebagai titik dan jarak antar sumur sebagai garis.

Pencarian titik pendominasi menggunakan algoritma *greedy*. Algoritma *greedy* adalah salah satu algoritma yang sering dipakai dalam cabang ilmu matematika dan komputer. Algoritma ini merupakan alternatif lain bagi penggunaan algoritma seperti *brute force*, *backtracking*, *dynamic programming*, dan sebagainya dalam pemecahan berbagai persoalan. Salah satu penerapan algoritma *greedy* dapat ditemukan pada topik graf (Passa, 2010).

Pada penelitian dengan judul “*A greedy approximation for minimum connected dominating sets*” yang dilakukan oleh L. Ruan, et al dari *Iowa State University* pada tahun 2004. Penelitian tersebut menjelaskan mengenai cara untuk mencari *minimum dominating set* menggunakan algoritma *greedy*. Penelitian tersebut menghasilkan jumlah minimal *dominating set* yang dapat diperoleh dari sebuah graf menggunakan algoritma *greedy* dengan rasio performa $\ln^{\delta} + 2$ dengan δ adalah derajat terbesar yang dimiliki dalam sebuah graf.

Penelitian ini bertujuan untuk mengetahui *minimum dominating set* pada graf sumur resapan di Universitas Jember. Algoritma pencarian yang digunakan untuk memecahkan masalah tersebut adalah algoritma *greedy*.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana pengimplementasian algoritma *greedy* untuk menentukan *dominating set* pada Graf Sumur Resapan di Universitas Jember?
2. Bagaimana merancang dan membangun sistem informasi penentu *dominating set* pada graf sumur resapan di lingkungan Universitas Jember menggunakan algoritma *greedy*?

1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

1.3.1 Tujuan

Adapun tujuan yang ingin dicapai dalam penelitian ini adalah:

1. Untuk mengetahui cara penerapan algoritma *greedy* dalam penentuan *dominating set* pada graf sumur resapan di Universitas Jember.
2. Untuk membuat perancangan dan pembangunan Sistem Informasi penentu *dominating set* pada graf sumur resapan di lingkungan Universitas Jember menggunakan algoritma *greedy*.

1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Diharapkan dapat membantu dalam menentukan *dominating set* pada graf sumur resapan di Universitas Jember.
2. Diharapkan dapat menjadi dasar bagi peneliti pada penelitian selanjutnya untuk pengembangan sistem informasi yang lebih baik.
3. Dapat meningkatkan keilmuan tentang Sistem Informasi penentu *dominating set* pada sumur resapan di lingkungan Universitas Jember menggunakan algoritma *greedy*, sekaligus sebagai media untuk penyelesaian tugas akhir jenjang S1 pada Program Studi Sistem Informasi, Universitas Jember.

1.4 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan Sistem Informasi penentu *dominating set* pada sumur resapan di lingkungan Universitas Jember menggunakan algoritma *greedy*. Adapun batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Sistem informasi ini digunakan untuk menentukan *dominating set* pada graf sumur resapan di Universitas Jember
2. Pencarian titik pendominasi menggunakan algoritma *greedy*
3. Peletakan *node* dan *vertex* menggunakan graf grid, sumur resapan direpresentasikan sebagai *node* dan jarak antar sumur resapan direpresentasikan sebagai *vertex*
4. Sistem Informasi dibangun berbasis *Website*

1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

1. Pendahuluan

Bab kesatu ini memuat uraian tentang latar belakang, perumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing tertuang secara eksplisit dalam subbab tersendiri.

2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep pemetaan strata desa siaga aktif, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

3. Metode Penelitian

Bab ini menguraikan tentang tempat dan waktu penelitian, metode penelitian, metode pengumpulan data, metode analisis data, dan teknik pengembangan sistem yang digunakan dalam penelitian.

4. Pengembangan Sistem

Bab ini berisi uraian tentang langkah-langkah yang ditempuh dalam proses menganalisis dan merancang sistem yang hendak dibangun meliputi desain, pengkodean, dan pengujian sistem.

5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

6. Penutup

Bab ini terdiri atas kesimpulan atas penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.

BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

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 HestiAgustin, 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).

Penelitian lainnya dengan judul “A greedy approximation for minimum connected dominating sets” yang dilakukan oleh L. Ruan, et al dari Iowa State University pada tahun 2004. Penelitian tersebut menjelaskan mengenai cara untuk mencari *minimum dominating set* menggunakan *algoritma greedy*. Penelitian tersebut menghasilkan jumlah minimal *dominating set* yang dapat diperoleh dari sebuah graf menggunakan *algoritma greedy* dengan rasio performa $\ln^{\delta} + 2$ dengan δ adalah derajat terbesar yang dimiliki dalam sebuah graf (L. Ruan, 2004).

Berdasarkan penelitian tersebut dapat disimpulkan bahwa dalam penelitian penentuan titik pendominasi pada sumur resapan di Universitas Jember menggunakan metode *dominating set* dapat diketahui jumlah *minimum dominating set* dari vertex

sumur. Kemudian dalam melakukan pencarian jumlah *minimum dominating set* dapat menggunakan *algoritma greedy*.

2.2 Algoritma Greedy

Algoritma *greedy* merupakan metode yang paling populer dalam memecahkan persoalan optimasi. Hanya ada dua macam persoalan optimasi, yaitu maksimasi dan minimasi. Algoritma *greedy* adalah algoritma yang memecahkan masalah langkah per langkah. Algoritma *greedy* membentuk solusi langkah per langkah. Pada setiap langkah, terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkahnya merupakan pilihan, untuk membuat langkah optimum lokal (*local optimum*) dengan harapan bahwa langkah sisanya mengarah ke solusi optimum global (*global optimum*) (Zulhidayati, 2013).

Komponen algoritma *greedy* terdiri dari (Ardiansyah, 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

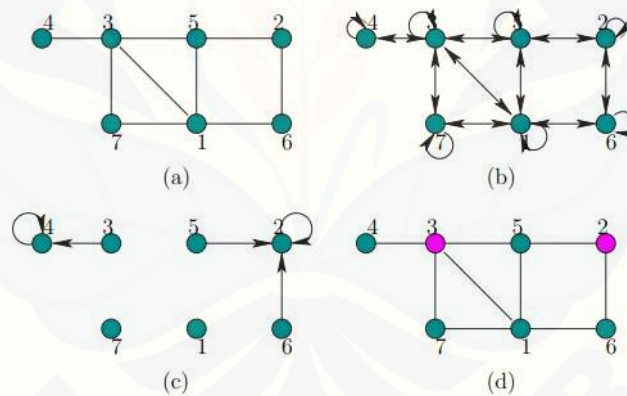
Fungsi yang memeriksa apakah suatu kandidat yang terpilih akan menimbulkan solusi yang layak, yaitu kandidat tersebut, bersama dengan himpunan solusi yang terpilih tidak akan melanggar kendala yang berlaku pada masalah.

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).

Contoh dari langkah-langkah *greedy algorithm* dalam menentukan *dominating set* seminimal mungkin dapat dilihat di Gambar 2.1.



Gambar 2. 1 Contoh dari *Greedy Algorithm* (Alfarisi, 2014)

2.3 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 (G) (Alanko, 2011).

Dalam (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 .

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

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

Keterangan :

p : jumlah titik-titik yang ada pada sebarang graf

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

$\gamma(G)$: jumlah perkiraan *dominating set*

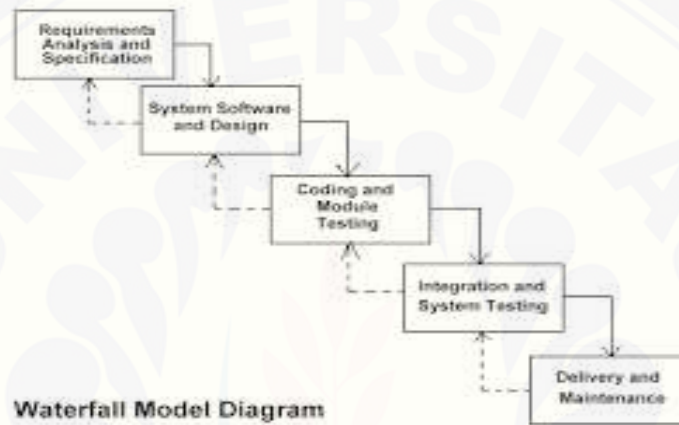
Contoh sebuah graf G dengan himpunan pendominasinya dapat dilihat di Gambar 2.2.



Gambar 2. 2 Graf G dengan himpunan pendominasinya $\{1, 3, 5\}$
(Iswadi, 2011)

2.4 Model Waterfall

Model *waterfall* merupakan metode yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem sampai pada analisis, desain, kode, test dan pemeliharaan (Sommerville, 2003). Gambaran model *waterfall* dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Model *Waterfall* (Sommerville, 2003)

BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang gambaran tahapan yang sistematis yang dilakukan untuk menganalisa data untuk menjawab perumusan masalah sehingga dapat mencapai tujuan sebenarnya dari penelitian. Pada metodologi penelitian akan dijelaskan tentang tahapan dari penelitian.

3.1. Tahapan Penelitian

Rancang bangun aplikasi pada penelitian ini menggunakan pemodelan *System Development Life Cycle (SDLC) waterfall*. Penggunaan *SDLC Waterfall* disesuaikan dengan kebutuhan sistem informasi penentu *dominating set* pada graf sumur resapan di Universitas Jember.

3.1.1 Tahapan Analisis Kebutuhan

Penelitian dilakukan di Kampus Tegal Boto Universitas Jember yang beralamat di Jalan Kalimantan No. 26 Jember. Waktu penelitian dilakukan selama 4 bulan dimulai pada bulan Januari 2015 sampai April 2016. Tahapan analisis kebutuhan pada penelitian yang telah dilakukan tersebut dimulai dengan tahap pengumpulan data meliputi studi pustaka, dan studi lapangan. Tahapan yang selanjutnya yaitu tahap pengolahan data dan gambaran umum sistem yang akan dibangun.

1. Tahapan Pengumpulan Data

Teknik pengumpulan data yang akan dilakukan pada penelitian ini adalah sebagai berikut:

a. Studi Pustaka

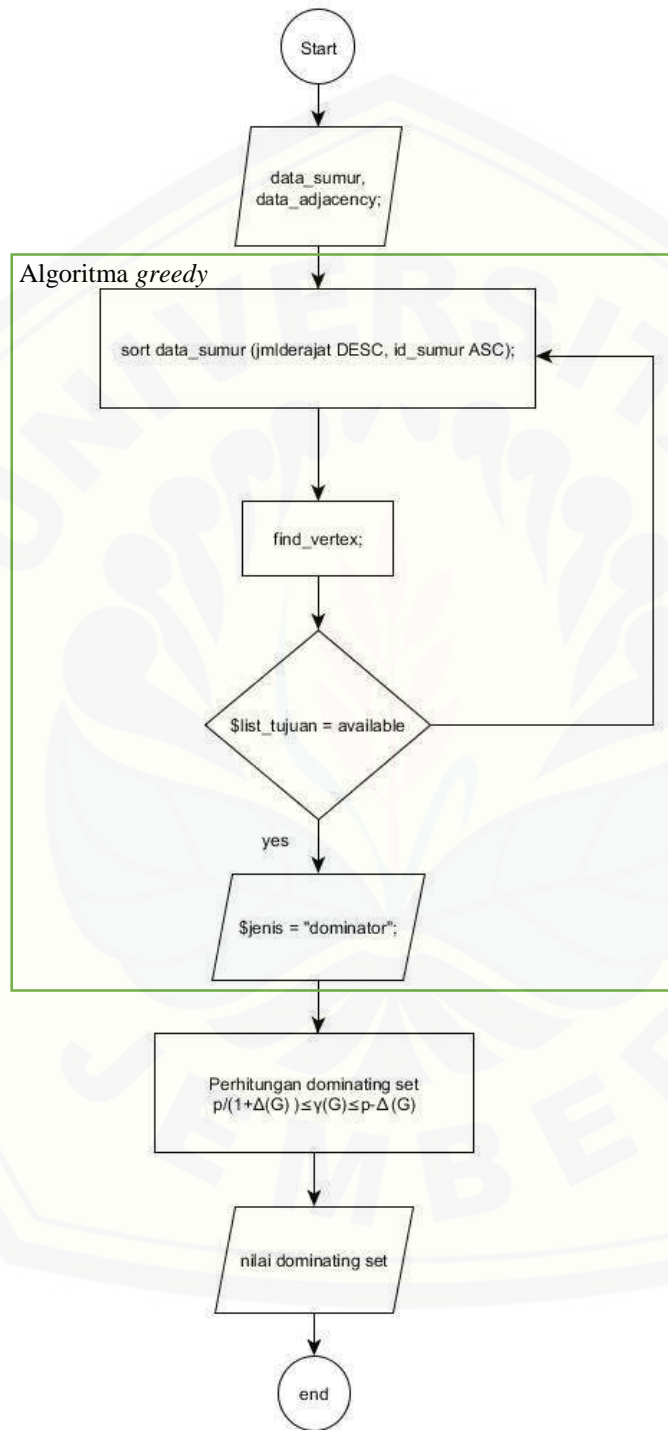
Studi pustaka ini dilakukan dengan tujuan sebagai dasar pembahasan penyusunan dasar teori yang digunakan dalam penelitian. Sumber yang digunakan sebagai studi pustaka berupa buku, jurnal, dan karya ilmiah dari penelitian yang sejenis sebelumnya.

b. Studi Lapang

Dalam 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 sumur resapan sampai jarak antar sumur resapan. Studi lapang juga dilakukan untuk melakukan *tracking* lahan Universitas Jember.

2. Tahap Pengolahan Data

Tahap pengolahan data dimulai dengan melakukan analisis dari data yang telah didapatkan. Data yang digunakan dalam penelitian ini adalah data lokasi sumur 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 sumur resapan dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Proses Perhitungan Algoritma *Greedy* dan *Dominating set*

3. Gambaran Umum Sistem

Sistem Informasi yang dibangun merupakan sistem informasi untuk penentuan *dominating set* atau himpunan pendominasi pada graf dengan studi kasus sumur resapan di Universitas Jember. Sistem informasi penentuan letak sumur resapan berbasis *Web* dan menggunakan database *MySQL*. Sistem informasi ini dibangun untuk mengetahui titik pendominasi pada graf sumur resapan di Universitas Jember. Titik yang digunakan adalah sumur resapan dan garisnya adalah jarak antar sumur resapan. Untuk menentukan titik dan garis dilakukan pemetaan sesuai dengan kondisi di Universitas Jember.

3.1.2 Tahapan Desain Sistem

Tahap yang selanjutnya yaitu desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang menggunakan konsep *Object-Oriented Programming* (OOP). Pemodelan UML yang akan digunakan sebagai berikut:

1. *Business Process*

Business Process merupakan proses untuk menggambarkan inputan data yang dibutuhkan oleh sistem, *output* dari sistem, serta tujuan dari pembuatan sistem.

2. *Usecase Diagram*

Usecase Diagram adalah rangkaian atau uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Aktor yang dapat mengakses aplikasi yaitu admin super, pakar sumur, dan pakar graf. Fitur yang disediakan oleh aplikasi ini adalah *home, well's location, adjacency list, analisis, login, logout*.

3. Skenario

Skenario merupakan penggambaran tahapan yang dilakukan oleh aktor dan tanggapan oleh sistem. Skenario menggambarkan keadaan normal saat setiap aktor mengakses aplikasi dan keadaan alternatif yang terjadi pada suatu kondisi tertentu.

4. *Sequence Diagram*

Sequence Diagram adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*.

5. *Activity Diagram*

Activity Diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir.

6. *Class Diagram*

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

7. *Entity Relationship Diagram (ERD)*

ERD merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

3.1.3 Tahapan Implementasi Sistem

Pada tahap ini desain yang telah dibuat akan diimplementasikan ke dalam kode program. Hal yang dilakukan dalam tahap implementasi antara lain:

1. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)* dengan bantuan *framework Code Igniter (CI)*.
2. Manajemen basisdata menggunakan MySQL.

3.1.4 Tahapan Pengujian Sistem

Pada tahap ini dilakukan pengujian untuk mengetahui apakah sistem informasi yang dibangun sudah sesuai dengan kebutuhan dari *user* atau belum. Pengujian yang dilakukan menggunakan cara sebagai berikut:

1. *White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian. Pengujian *white box* menggunakan *cyclomatic Complexity (CC)*. *Cyclomatic complexity* merupakan *software metric* yang menyediakan ukuran kuantitatif dari kompleksitas logikal suatu program. Ketika digunakan dalam konteks metode ujicoba berbasis alur, nilai yang dikomputasi untuk kompleksitas *cyclomatic* mendefinisikan jumlah *independent path* dalam himpunan basis suatu program dan menyediakan batas atas untuk sejumlah ujicoba yang harus dilakukan untuk memastikan bahwa seluruh perintah telah dieksekusi sedikitnya satu kali (Ayuliana, 2009). Kompleksitas *cyclomatic* dapat dicari dengan cara berikut:

$$V(G) = E - N + 2$$

Dimana E = jumlah edge, dan N = jumlah node

2. *Black Box testing* merupakan pengujian *software* yang memfokuskan pada keperluan fungsional dari *software*. Pengujian *black box* dilakukan oleh *user* dari sistem informasi yang dibangun. Pengujian dilakukan dengan cara menjalankan program dan melakukan analisis terhadap *input* dan *output* program.

3.1.5 Tahapan Pemeliharaan Sistem

Perawatan diadakan untuk mengatasi masalah pada sistem dilain waktu ketika aplikasi sudah dapat digunakan oleh *user*. Selama *user* menemui bug pada aplikasi ini, maka *user* langsung dapat mengkonfirmasi kepada *developer* untuk segera ditangani oleh *developer*.

BAB 4. PENGEMBANGAN SISTEM

Bab ini akan membahas tentang pengembangan sistem informasi penentu *dominating set* pada graf sumur resapan 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 Sistem

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 *dominating set* pada graf sumur resapan di Universitas Jember yaitu:

1. Sistem dapat mengelola data *user* (*create, read, update, delete*).
2. Sistem dapat mengelola data lokasi sumur (*create, read, update, delete*).
3. Sistem dapat mengelola data *adjacency list* (*create, read, update, delete*).
4. Sistem dapat menampilkan hasil penentuan letak sumur resapan 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 sumur resapan di Universitas Jember yaitu:

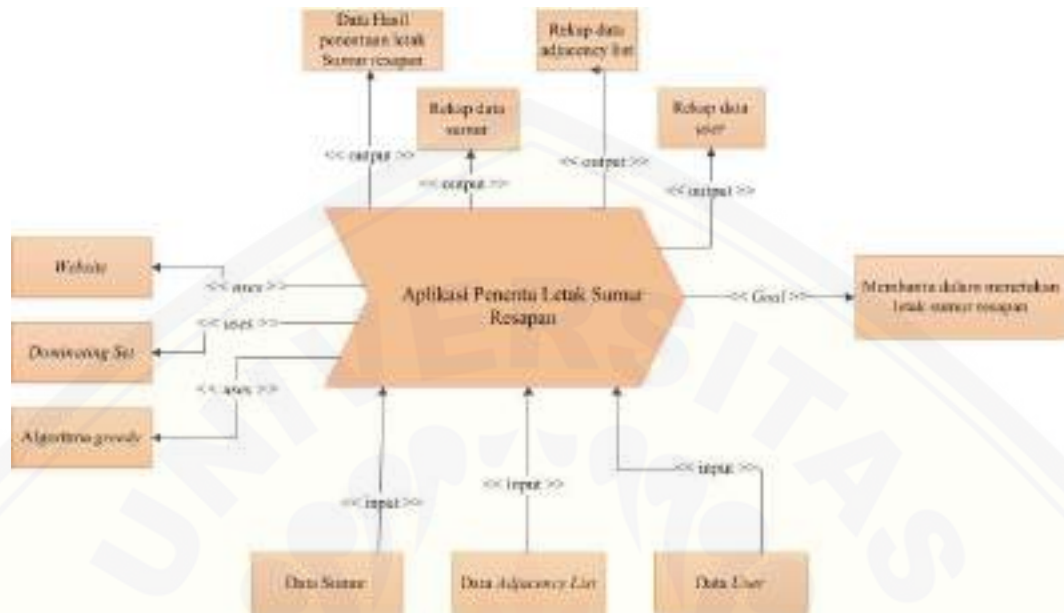
1. Sistem memiliki batasan hak akses pengguna dengan menggunakan *username* dan *password*.
2. Sistem berbasis *website*.
3. Sistem menggunakan framework *CodeIgniter*.

4.2 Desain Sistem

Desain sistem yang dibuat berdasarkan gambaran sistem yang telah dijelaskan pada bab metodologi penelitian subbab 3.1.2. Desain yang dibuat meliputi *business process*, *use case diagram*, *use case scenario*, *Sequence Diagram*, *Activity Diagram*, *class diagram*, dan ERD.

4.2.1 Business Process

Business Process adalah suatu kumpulan aktivitas yang terstruktur untuk mencapai suatu tujuan tertentu atau untuk menghasilkan sebuah produk. Ada beberapa komponen di dalamnya, meliputi data yang menjadi masukan (*input*), data masukan yang kemudian diolah menjadi data keluaran (*output*), media yang digunakan (*uses*), dan tujuan yang ingin dicapai (*goal*). *Business Process* aplikasi ini dapat dilihat pada Gambar 4.1.

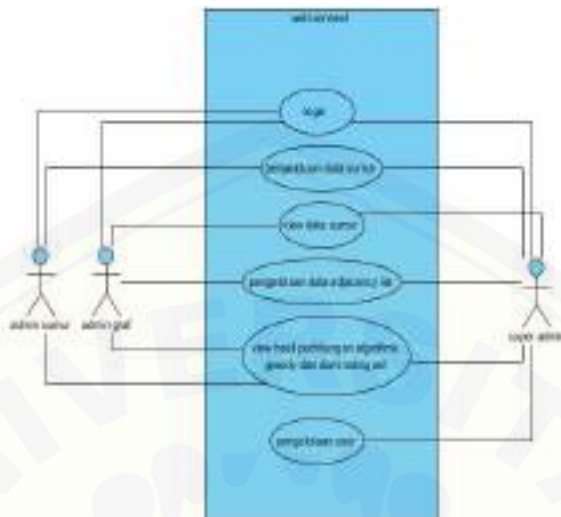


Gambar 4. 1 Business Process

4.2.2 Use Case Diagram

Use case diagram merupakan pemodelan yang dibuat untuk dapat menggambarkan interaksi antara aktor dengan sistem informasi penentu lokasi sumur yang akan dibangun. Melalui *use case diagram* dapat diketahui interaksi yang dapat dilakukan aktor terhadap sistem sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. Pada Gambar 4.2 digambarkan *use case diagram* sistem informasi prediksi yang terdiri dari tiga aktor dengan enam *use case*.

Penjelasan tentang definisi aktor dan definisi *use case* dalam *use case diagram* pada Gambar 4.2 akan dijelaskan di bawah ini.



Gambar 4. 2 Use Case Diagram

1. Definisi Aktor

Definisi aktor akan memberikan penjelasan mengenai aktor-aktor sebagai pengguna dari sistem informasi sumur resapan yang akan dibangun. Terdapat tiga aktor seperti yang dijelaskan pada tabel 4.1.

Tabel 4. 1 Definisi Aktor

No.	Aktor	Deskripsi
1.	Superadmin	Merupakan aktor yang memiliki hak akses penuh terhadap sistem informasi. Aktor superadmin dapat melakukan <i>login</i> , mengolah data sistem secara keseluruhan meliputi <i>data user</i> , data lokasi sumur, data adjacency list, dan analisis <i>dominating set</i> .
2.	Admin Sumur	Merupakan aktor yang memiliki hak akses untuk mengelola data sumur. Pakar sumur juga memiliki hak akses untuk melihat hasil analisis <i>dominating set</i> .

3.	Admin Graf	Merupakan aktor yang memiliki hak akses untuk mengelola data adjacency list dan data analisis <i>dominating set</i> . Pakar sumur juga memiliki hak akses untuk melihat lokasi sumur resapan.
----	------------	---

2. Definisi *Use Case*

Definisi use case merupakan penjelasan dari masing-masing use case atau fitur-fitur dari sistem informasi sumur resapan yang akan dibangun. Terdapat enam fitur 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 <i>user</i> untuk masuk ke dalam sistem informasi
2.	Pengelolaan data sumur	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data sumur
3.	Pengelolaan data adjacency list	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data adjacency list
4.	<i>View</i> hasil perhitungan algoritma <i>greedy</i> dan <i>dominating set</i>	Menggambarkan proses melihat hasil perhitungan menggunakan algoritma <i>greedy</i> dan <i>dominating set</i>
5.	Pengelolaan <i>user</i>	Menggambarkan proses menambah, mengedit, menghapus, dan melihat data <i>user</i>
6.	<i>View</i> data sumur	Menggambarkan proses melihat data sumur

4.2.3 Use Case Scenario

Use case scenario digunakan untuk menjelaskan alur sistem sesuai dengan yang ada pada *Use case diagram* seperti pada Gambar 4.2.

1. Skenario *Use Case Login*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case login* dijelaskan pada lampiran A.1.

2. Skenario *Use Case Pengelolaan Data User*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data user* dapat dilihat pada lampiran A.2.

3. Skenario *Use Case View Data Sumur*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case view data Sumur* dapat dilihat pada lampiran A.3.

4. Skenario *Use Case Pengelolaan Data Sumur*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case Pengelolaan Data kontrol objektif* dapat dilihat pada lampiran A.4.

5. Skenario *Use Case Pengelolaan Data Adjacency List*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case pengelolaan data Adjacency List* dapat dilihat pada lampiran A.5.

6. Skenario *Use Case View Hasil Perhitungan Algoritma Greedy dan Dominating set*

Penjelasan urutan aksi aktor dan reaksi sistem pada skenario normal dan skenario alternatif skenario *use case View Hasil Perhitungan Algoritma Greedy dan Dominating set* dapat dilihat pada tabel 4.3, tabel 4.4, dan tabel 4.5.

Tabel 4. 3 Skenario *Use Case View* hasil perhitungan *dominating set* dan *greedy* Super admin

Nama Use Case	<i>View</i> hasil perhitungan <i>dominating set</i> dan <i>greedy</i>
Aktor	Super admin

Deskripsi Singkat	Super admin akan melihat hasil perhitungan <i>dominating set</i> dan <i>greedy</i>
Prekondisi	Data sumur dan data adjacency list yang telah diisi
Prakondisi	Menampilkan hasil perhitungan <i>dominating set</i> dan <i>greedy</i>
Flow Events	
Skenario Normal : View Sumur	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>analysis</i>	
	2. Menampilkan halaman <i>analysis</i> meliputi peta dengan data sumur dan data adjacency list yang telah diinputkan serta tombol <i>analysis</i>
3. Klik tombol <i>analysis</i>	
	4. Menampilkan hasil perhitungan tahap 1
5. Klik <i>next</i>	
	6. Menampilkan hasil perhitungan tahap 2
7. Klik <i>next</i>	
	8. Menampilkan hasil perhitungan tahap 3
9. Klik <i>next</i>	

	10. Hasil perhitungan tahap 4
11. Klik next	
	12. Menampilkan hasil analisis tahap 5

Tabel 4. 4 Skenario *Use Case View* hasil perhitungan *dominating set* dan *greedy* admin graf

Nama Use Case	View hasil perhitungan <i>dominating set</i> dan <i>greedy</i>	
Aktor	Admin graf	
Deskripsi Singkat	Admin graf akan melihat hasil perhitungan <i>dominating set</i> dan <i>greedy</i>	
Prekondisi	Data sumur dan data <i>adjacency list</i> yang telah diisi	
Prakondisi	Menampilkan hasil perhitungan <i>dominating set</i> dan <i>greedy</i>	
Flow Events		
Skenario Normal : View Sumur		
Aksi Aktor	Reaksi Sistem	
1. Klik menu <i>analysis</i>		
	2. Menampilkan halaman <i>analysis</i> meliputi peta dengan data sumur dan data <i>adjacency list</i> yang telah diinputkan serta tombol <i>analysis</i>	
3. Klik tombol <i>analysis</i>		

	4. Menampilkan hasil perhitungan tahap 1
5. Klik <i>next</i>	
	6. Menampilkan hasil perhitungan tahap 2
7. Klik <i>next</i>	
	8. Menampilkan hasil perhitungan tahap 3
9. Klik <i>next</i>	
	10. Hasil perhitungan tahap 4
11. Klik <i>next</i>	
	12. Menampilkan hasil analisis tahap 5

Tabel 4. 5 Skenario *Use Case View* hasil perhitungan *dominating set* dan *greedy* admin sumur

Nama Use Case	<i>View</i> hasil perhitungan <i>dominating set</i> dan <i>greedy</i>
Aktor	Admin sumur
Deskripsi Singkat	Admin sumur akan melihat hasil perhitungan <i>dominating set</i> dan <i>greedy</i>
Prekondisi	Data sumur dan data adjacency list yang telah diisi
Prakondisi	Menampilkan hasil perhitungan <i>dominating set</i> dan <i>greedy</i>

Flow Events	
Skenario Normal : <i>View Sumur</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>analysis</i>	
	2. Menampilkan halaman <i>analysis</i> meliputi peta dengan data sumur dan data adjacency list yang telah diinputkan serta tombol <i>analysis</i>
3. Klik tombol <i>analysis</i>	
	4. Menampilkan hasil perhitungan tahap 1
5. Klik <i>next</i>	
	6. Menampilkan hasil perhitungan tahap 2
7. Klik <i>next</i>	
	8. Menampilkan hasil perhitungan tahap 3
9. Klik <i>next</i>	
	10. Hasil perhitungan tahap 4
11. Klik <i>next</i>	
	12. Menampilkan hasil analisis tahap 5

4.2.4 *Sequence Diagram*

Sequence Diagram adalah diagram yang digunakan untuk menggambarkan interaksi yang terjadi antarobjek di dalam sistem yang disusun pada sebuah urutan dan rangkaian waktu pada aplikasi penilaian tingkat kematangan tata kelola TI.

1. *Sequence Diagram Login*

Penggambaran *Sequence Diagram login* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.1.

2. *Sequence Diagram* Pengelolaan Data *User*

Penggambaran *Sequence Diagram* pengelolaan data *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.2.

3. *Sequence Diagram view* Data Sumur

Penggambaran *Sequence Diagram view* data sumur digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.3.

4. *Sequence Diagram* Pengelolaan Data Sumur

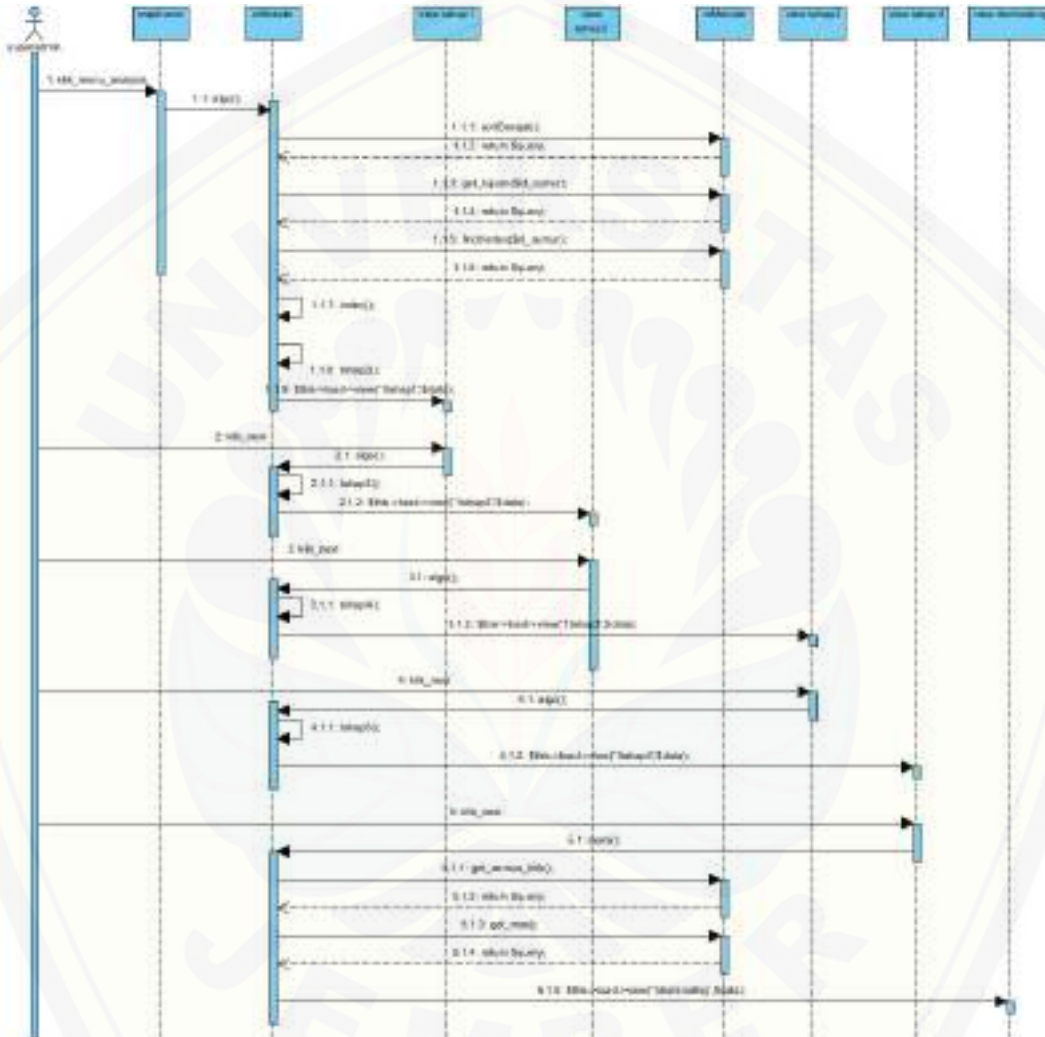
Penggambaran *Sequence Diagram* pengelolaan data sumur digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.4.

5. *Sequence Diagram* Pengelolaan Data *Adjacency List*

Penggambaran *Sequence Diagram* pengelolaan data *Adjacency List* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada lampiran B.5.

6. *Sequence Diagram View* Hasil Perhitungan Algoritma *Greedy* dan *Dominating set*

Penggambaran *Sequence Diagram View* Hasil Perhitungan Algoritma *Greedy* dan *Dominating set* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat seperti yang ditunjukkan pada Gambar 4.3.

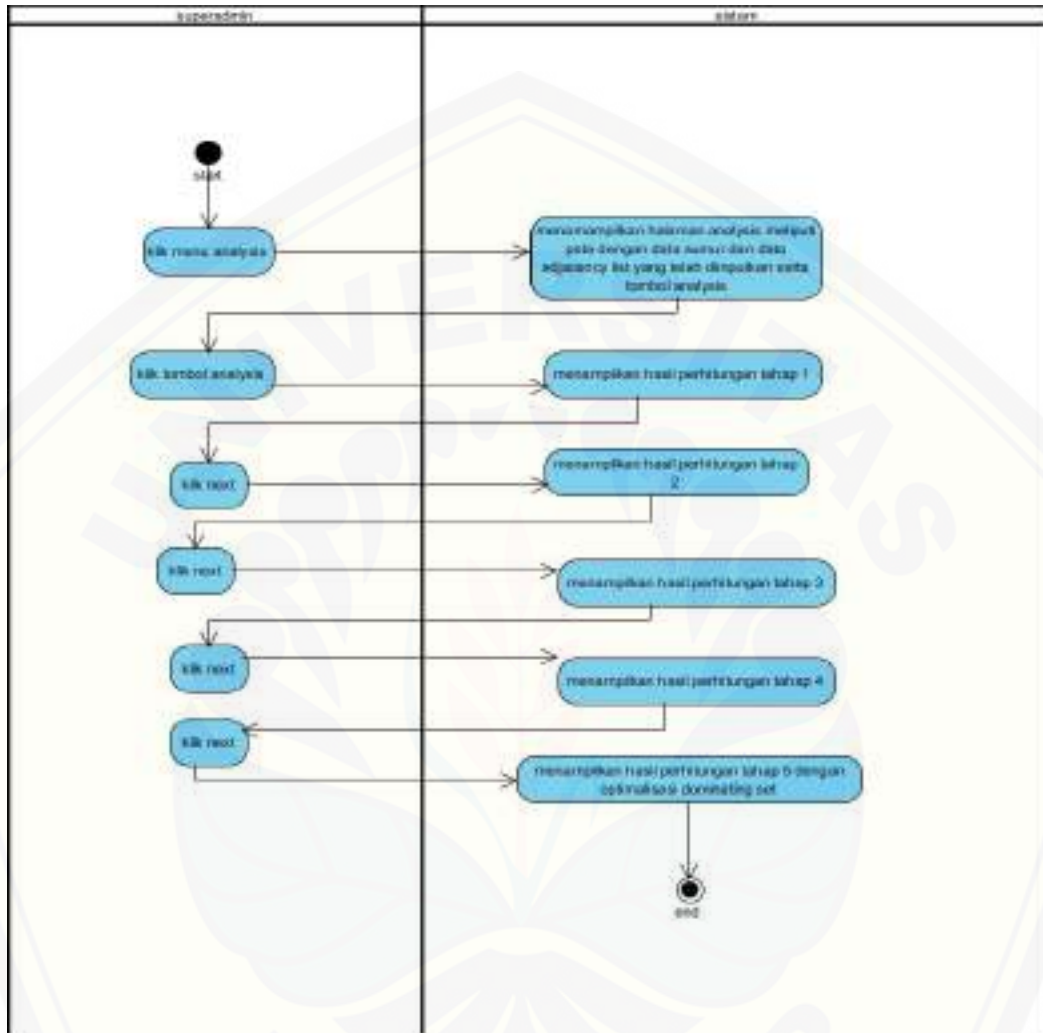


Gambar 4. 3 *Sequence Diagram view* Hasil Perhitungan algoritma *greedy* dan *dominating set*

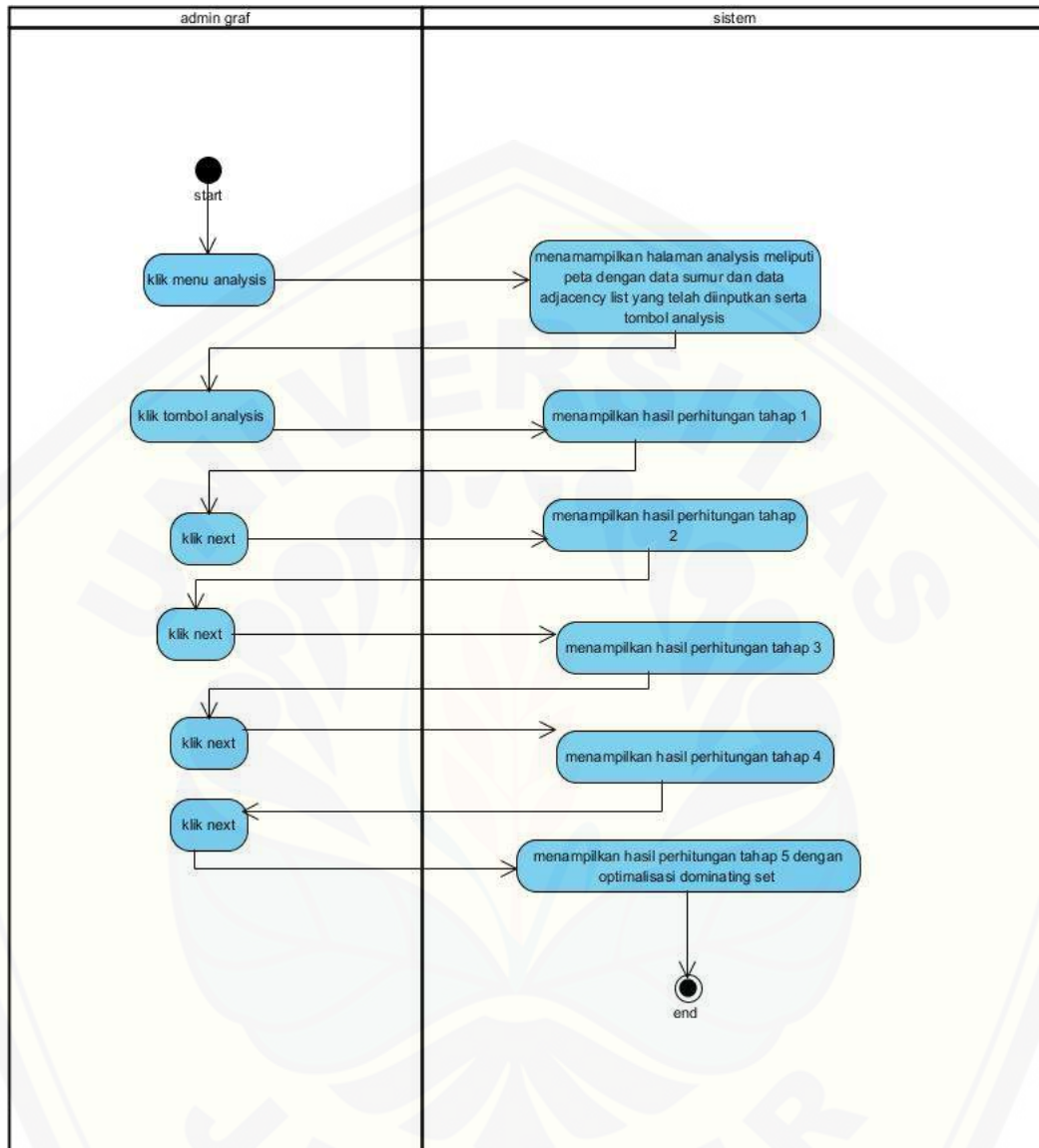
4.2.5 Activity Diagram

Activity Diagram menggambarkan alur aktivitas pada aplikasi penilaian tingkat kematangan tata kelola TI.

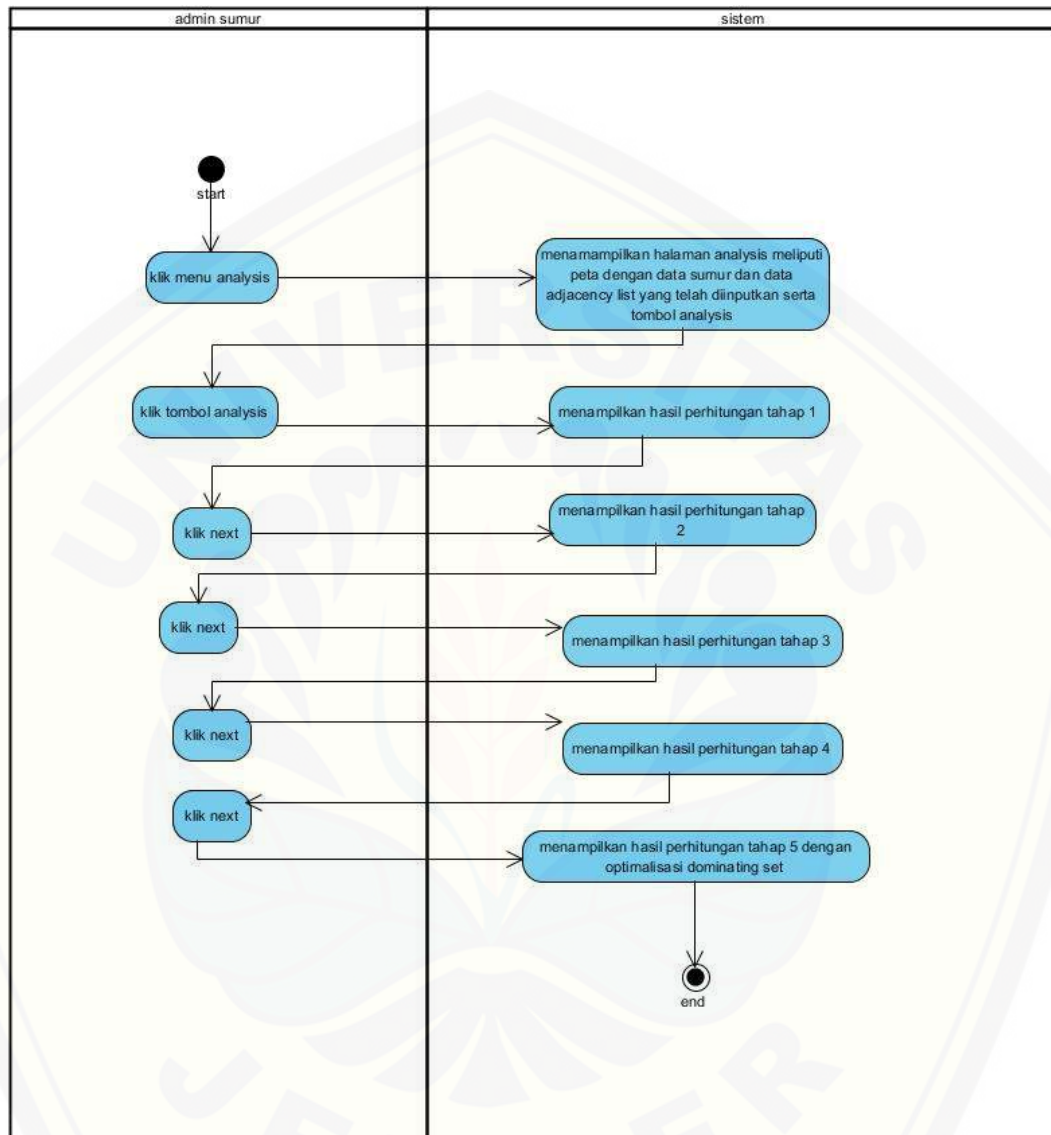
1. *Activity Diagram Login*
Activity Diagram login dapat dilihat pada lampiran C.
2. *Activity Diagram Pengelolaan Data User*
Activity Diagram pengelolaan data *user* dapat dilihat pada lampiran C.
3. *Activity Diagram View Data Sumur*
Activity Diagram view data domain dapat dilihat pada lampiran C.
4. *Activity Diagram Pengelolaan Data Sumur*
Activity Diagram pengelolaan data Sumur dapat dilihat pada lampiran C.
5. *Activity Diagram Pengelolaan Data Adjacency List*
Activity Diagram pengelolaan data *Adjacency List* dapat dilihat pada lampiran C.
6. *Activity Diagram View Hasil Perhitungan Algoritma Greedy dan Dominating set*
Activity Diagram View Hasil Perhitungan Algoritma *Greedy* dan *Dominating set* dapat dilihat pada Gambar 4.3, Gambar 4.4, dan Gambar 4.5.



Gambar 4. 4 Activity Diagram view hasil dominating set dan greedy super admin



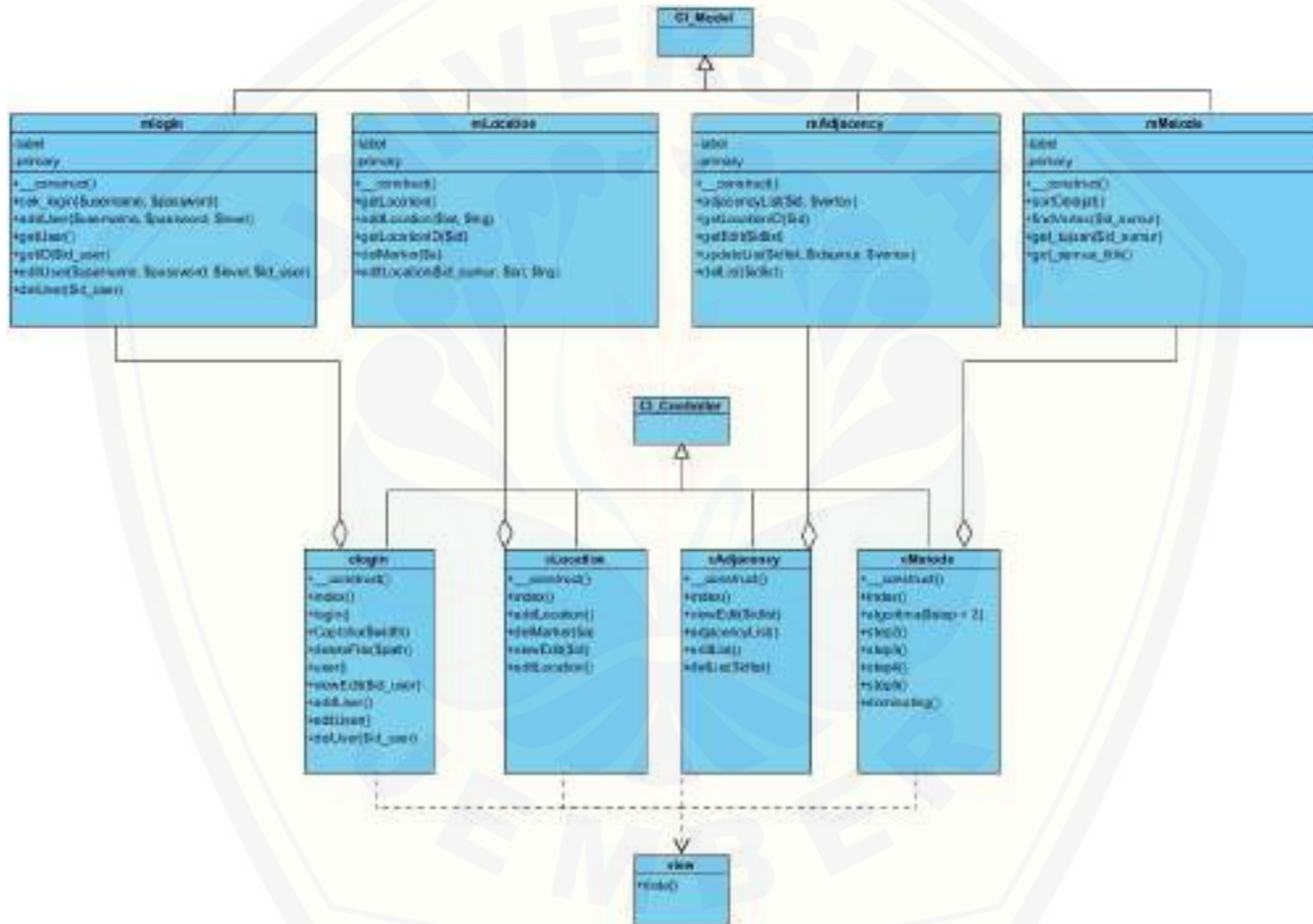
Gambar 4. 5 Activity Diagram view hasil dominating set dan greedy admin graf



Gambar 4. 6 Activity Diagram view hasil dominating set dan greedy admin sumur

4.2.6 Class Diagram

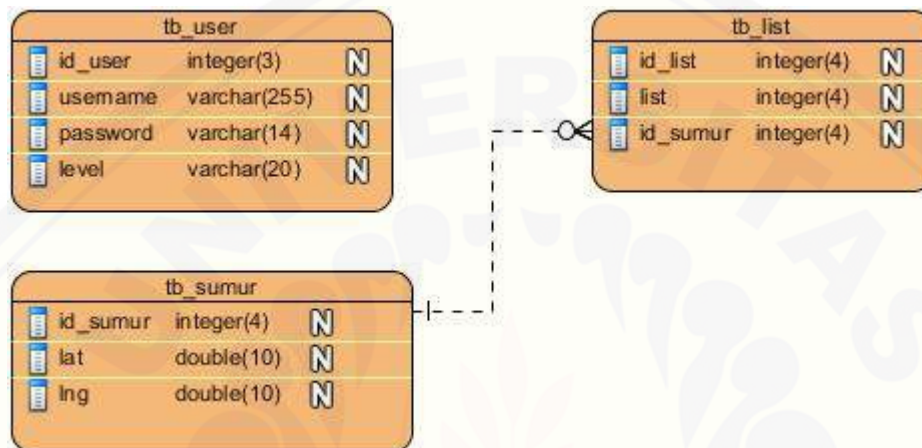
Class Diagram menggambarkan hubungan antarkelas yang digunakan untuk membangun suatu sistem. *Class Diagram* aplikasi penentu letak sumur resapan di Universitas Jember dapat dilihat pada Gambar 4.7



Gambar 4.7 *Class Diagram* Aplikasi Penentu Sumur Resapan di Universitas Jember

4.2.7 Entity Relationship Diagram (ERD)

ERD merupakan gambaran komponen dan struktur database yang digunakan dalam pembangunan sistem. ERD aplikasi penentu letak sumur resapan di Universitas Jember dapat dilihat pada Gambar 4.8.



Gambar 4. 8 ERD Aplikasi Penentu Letak Sumur Resapan di Universitas Jember

4.3 Pengkodean Sistem

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 *view* login, *controller* clogin, dan model mlogin. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.2 Kode Program Pengelolaan Data *User*

Kode program pengelolaan data user terletak pada kelas *view* user, editUser, controller clogin, dan model mlogin. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.3 Kode Program *View* Data Sumur

Kode program *view* data sumur terletak pada kelas *view* inputmap, controller cLocation dan model mLocation. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.4 Kode Program Pengelolaan Data Sumur

Kode program pengelolaan data sumur terletak pada kelas *view* inputmap, editSumur, controller cLocation, dan model mLocation. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.5 Kode Program Pengelolaan Data *Adjacency List*

Kode program pengelolaan data sumur terletak pada kelas *view* adjacencyList, editList, controller cAdjacency, dan model mAdjacency. Penulisan kode program ini dapat dilihat pada lampiran D.

4.3.6 Kode Program *View* Hasil Perhitungan Algoritma *Greedy* dan *Dominating set*

Kode program *view* Hasil Perhitungan Algoritma *Greedy* dan *dominating set* terletak pada kelas *view* tahap1, tahap2, tahap3, tahap4, dominating, controller cMetode, dan model mMetode. Penulisan kode program ini dapat dilihat pada Gambar.

1. Kelas controller cMetode

Penulisan kode program algoritma *greedy* dan *dominating set* dapat dilihat pada gambar 4.9 dan 4.10.

```

<?php
if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class CMetode extends CI_Controller {

private $dominatorSamaYangTerpakai = [];
private $listTidak = [];
private $sisane = [];
private $jomblo = [];

public function __construct(){
    parent::__construct();
    $this->load->helper('url');
    $this->load->model('mMetode', "a");
    $this->load->library('session');
}

public function index(){
    $list_titik = $this->a->sortDerajat();
    $list_tujuan = array(); //node yg sudah di cover
    $tujuan_gagal = array();
    foreach ($list_titik as $l){
        $tujuan = $this->a->get_tujuan($l["id_sumur"]);
        $available = false;
        foreach ($list_tujuan as $t){
            foreach ($tujuan as $t){
                if ($t["list"] == $l["id_sumur"] || $t["list"] == $l["id_sumur"]){
                    $available = true;
                }
            }
        }
    }

    if (!$available){

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

```

```

}
$indek = 0;
$titik = 0;
$l = true;
foreach($list_tujuan as $row){
if($titik == $row['id_sumur'] && $l){
$temp[$indek++] = $row['id_sumur'];
$l = false;
}

if($titik != $row['id_sumur'] && $titik != 0){

$titik = $row['id_sumur'];
$l = true;
}else if($titik == 0){
$titik = $row['id_sumur'];
}
$temp[$indek++] = $row['list'];
}
$tidak = array();
$list_tidak = array();
$count = 0;
foreach($list_titik as $list){
foreach($temp as $t){
if($list['id_sumur'] == $t){
$count++;
}
}
if($count == 0){
array_push($tidak,$list['id_sumur']);
$this->listTidak[] = [
'jenis' => 'tidak',
'isi' => $this->a->findVertex($list['id_sumur'])[0] ];
}

$count = 0;
}
$titik_habis = array();
foreach($tidak as $list){
$kosongSemua = true;
$tujuan = $this->a->get_tujuan($list);
for($i=0;$i<count($tujuan);$i++){
if(in_array($tujuan[$i]['list'], $temp)){
$tujuan[$i]['list'] = null;
$tujuan[$i]['id_sumur'] = null;
}
}
}
for($i=0;$i<count($tujuan);$i++){
if($tujuan[$i]['id_sumur'] != "" || $tujuan[$i]['list'] != ""){
array_push($list_tidak, array('id_sumur' => $tujuan[$i]['id_sumur'], 'list' => $tujuan[$i]['list']));
$kosongSemua = false;
}
}

```



```

}
if ($kosongSemua & count($tujuan) > 0) {
    $this->jomblo[] = [
        'jenis' => 'jomblo',
        'isi' => $this->a->findVertex($list)[0] ];
    }
}
$list_tidak_dominan = array();
$count = 0;
for($i=0;$i<count($list_tidak);$i++){
    if($i==0){
        $titik_bantu = $list_tidak[$i]['id_sumur'];
    }
    if($list_tidak[$i]['id_sumur'] == $titik_bantu )
    {
        $count++;
    }
    else
    {
        $list_tidak_dominan[$titik_bantu] = $count;
        $count = 1;
        $titik_bantu = $list_tidak[$i]['id_sumur'];
    }
}

arsort($list_tidak_dominan);

foreach($list_tidak_dominan as $key => $value){
    $c = 0;
    $tempTitik = 0;
    $list= "";
    $indexAnunya = -1;
    foreach($list_tidak as $lt){
        if($lt['id_sumur'] == $key){
            if(!in_array($lt['id_sumur'],$temp)){
                $c++;
                array_push($temp, $lt['list']);
                array_push($list_tujuan, array('id_sumur' => $lt['id_sumur'], 'list' => $lt['list']));
            }
            if ($tempTitik != $lt['id_sumur']) {
                $this->sisane[] = [
                    'jenis' => 'sisaDominator',
                    'isi' => $this->a->find_titik($lt['id_sumur'])[0] ];
                $indexAnunya = count($this->sisane)-1;
                $tempTitik = $lt['id_sumur'];
                $list = "";
            }
        }
        $this->sisane[] = [
            'jenis' => 'sisaTerpakai',
            'isi' => $this->a->find_titik($lt["list"])[0],
            'dominator' => $lt['id_sumur'] ];
        $list .= $lt["list"].",";
    }
}

```

```

        $this->sisane[$indexAnunya]['list'] = $list;
        if($c == $value)
        {
            array_push($temp, $t['id_sumur']);
        }
    }
}
}
foreach($tidak as $t){
    if(!in_array($t, $temp)){
        array_push($list_tujuan, array('id_sumur' => $t, 'list' => $t));
    }
}
$data['dominan'] = array();

foreach($list_tujuan as $list){
    if(!in_array($list['id_sumur'], $data['dominan'])){
        array_push($data['dominan'], $list['id_sumur']);
    }
}

return $data['dominan'];
}
}

```

Gambar 4. 9 Kode program algoritma *greedy*

```

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'];

    $data['min'] = round($p / (1+$g));
    $data['ds'] = count($this->index());
    $data['max'] = $p - $g;

    $this->load->view('sidebar');
    $this->load->view('1dominating',$data);
    $this->load->view('footer',$data);
}

```

Gambar 4. 10 Kode *function dominating*

2. Kelas mMetode

Penulisan kode program mMetode dapat dilihat pada gambar.

```
<?php
```

```

if(!defined('BASEPATH')) exit ('No direct script access allowed');
class MMetode extends CI_Model{
    public function sortDerajat() {
        $query = $this->db->query("SELECT l.id_sumur, COUNT(l.id_sumur) as jmlderajat, s.lat,
s.lng FROM `tb_list` l join tb_sumur s on(l.id_sumur=s.id_sumur) group by l.id_sumur order by
jmlderajat desc, id_sumur ASC")->result_array();
        return $query;}
    function findVertex($id_sumur) {
        return $this->db->query("SELECT `id_sumur`, `lng`, `lat` FROM tb_sumur WHERE
id_sumur = $id_sumur")->result_array(); }
    function get_tujuan($id_sumur) {
        return $this->db->query("SELECT `id_sumur`, `list` FROM tb_list WHERE id_sumur =
$id_sumur ORDER BY id_sumur ASC")->result_array();
    }
    public function get_semua_titik() {
        return $this->db->query("SELECT COUNT(id_sumur) as jumlah from tb_sumur")-
>result_array()[0]['jumlah'];
    }
}
?>

```

Gambar 4. 11 Penulisan Kode Program mMetode

4.4 Pengujian Sistem

Tahapan pengujian sistem merupakan suatu tahapan yang dilakukan secara sistematis untuk menguji dan mengevaluasi sistem dengan menggunakan sebuah metode pengujian sistem. Hal tersebut dilakukan dengan tujuan untuk mengevaluasi apakah kebutuhan sistem telah terpenuhi dan sistem layak untuk digunakan oleh pengguna. Agar pengujian yang dilakukan lebih valid, maka tahap pengujian sistem informasi prediksi pembelian daya dilakukan dengan menggunakan dua metode, yaitu *white box* dan *black box*.

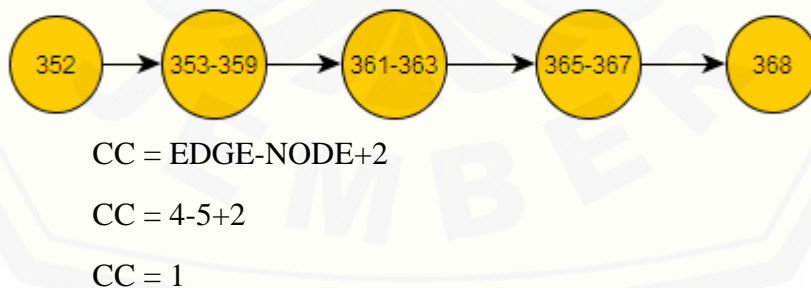
4.4.1 Metode White Box

Pengujian white box pada aplikasi penentu letak sumur resapan di Universitas Jember ini dengan cara menggambar diagram alir, menghitung kompleksitas siklomatiknya (CC), dan membuat tabel pengujian test case. Pengujian listing program ditunjukkan pada gambar 4.12, gambar digram alir ditunjukkan pada gambar 4.13, dan tabel pengujian alur test case ditunjukkan pada tabel 4.6. Sedangkan untuk fitur yang lain dapat dilihat pada lampiran E.

```

352 public function dominating(){
353     $this->load->library('../controllers/cMetode');
354     $this->index();
355     $hmm = array_merge($this->dominatorSamaYangTerpakai,$this->sisane,$this->jomblo);
356     $data['nodes'] = $hmm;
357     $list_titik = $this->a->sortDerajat();
358     $sp = $this->a->get_semua_titik();
359     $sg = $list_titik[0]['jmlDerajat'];
360
361     $data['min'] = round($sp / (1-$sg));
362     $data['ds'] = count($this->index());
363     $data['max'] = $sp - $sg;
364
365     $this->load->view('sidebar');
366     $this->load->view('ldominating',$data);
367     $this->load->view('Footer',$data);
368 }
    
```

Gambar 4. 12 Listing Program



Gambar 4. 13 Diagram alir

Jalur 1 : 352-353-354-356-357-358-359-360-361-362-363-364-365-366-367-368

Tabel 4. 6 Pengujian *Test Case* Perhitungan *Dominating set*

Jalur 1	
<i>Test Case</i>	Jika berhasil melakukan perhitungan <i>dominating set</i>
Target yang Diharapkan	Menampilkan hasil perhitungan <i>dominating set</i>
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	352-353-354-356-357-358-359-360-361-362-363-364-365-366-367-368

4.4.2 Metode *Black Box*

Pengujian black box berfungsi untuk menguji sistem dari segi spesifikasi fungsional sistem dengan tujuan mengetahui apakah fungsi-fungsi, inputan, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Hasil pengujian dengan metode black box dapat dilihat pada Tabel 4.7.

Tabel 4. 7 Pengujian *Black Box*

No.	Menu	Fungsi	Kasus	Hasil	Ket
1.	Pengelolaan Data <i>User</i>	Menu ini digunakan untuk tambah, edit, hapus dan lihat data <i>user</i>	Ketika Super Admin memilih data user dan tidak ada data <i>user</i> yang tersimpan	Menampilkan tabel yang kosong	OK
			Ketika Super admin memilih data <i>user</i> dan ada <i>user</i> yang tersimpan	Menampilkan list data <i>user</i> dalam bentuk tabel	OK

			Ketika super admin memasukkan data <i>user</i> di form <i>add data user</i> kemudian klik tombol simpan	Menampilkan halaman data <i>user</i>	OK
			Ketika super admin menyimpan data <i>user</i> dan data yang diinputkan kurang lengkap	Menampilkan peringatan disebelah kolom yang belum diisi “harap isi bidang ini”	OK
			Ketika super admin memilih tombol edit	Menampilkan form edit data <i>user</i>	OK
			Ketika super admin klik tombol simpan pada form yang telah diedit data usernya	Menampilkan halaman data user	OK
			Ketika super admin menyimpan data yang telah	Menampilkan peringatan disebelah kolom yang belum diisi	OK

			diubah dan datanya kurang lengkap	“harap isi bidang ini”	
			Ketika super admin klik tombol <i>delete</i>	Menampilkan pesan “Apakah anda yakin menghapus data ini?”	OK
			Ketika superadmin memilih tombol ya	Menghapus data yang dipilih dan menampilkan halaman data <i>user</i>	OK
			Ketika superadmin memilih tombol tidak	Menampilkan halaman data <i>user</i>	OK
2.	Pengelolaan data sumur	Menu ini digunakan untuk input, update, delete data sumur	Ketika super admin memilih menu <i>wells location</i> dan tidak ada data yang tersimpan	Menampilkan peta dan tabel yang kosong	OK
			Ketika super admin memilih menu <i>wells location</i> dan	Menampilkan peta dan tabel yang berisi data sumur	OK

			ada data yang tersimpan		
			Ketika super admin input lokasi sumur dan klik simpan	Menampilkan marker dengan data sumur	OK
			Ketika super admin klik tombol <i>find me</i>	Menampilkan marker sesuai lokasi <i>user</i>	OK
			Ketika super admin klik salah satu marker	Menampilkan pop up yang berisi detail data sumur	OK
			Ketika super admin klik tombol reset setelah klik peta	Menampilkan halaman wells location dengan menghilangkan marker yang sebelumnya dibuat	OK
			Ketika super admin memilih tombol edit	Menampilkan form edit	OK
			Ketika super admin mengubah data	Menampilkan pesan berhasil dan kembali ke	OK

			dan klik simpan	halaman wells location	
			Ketika super admin klik tombol <i>delete</i>	Menampilkan pesan “Apakah anda yakin menghapus data ini?”	OK
			Ketika superadmin memilih tombol ya	Menghapus data yang dipilih dan menampilkan halaman data <i>user</i>	OK
			Ketika superadmin memilih tombol tidak	Menampilkan halaman data <i>user</i>	OK
3.	Pengelolaan adjacency list	Menu ini digunakan untuk input, update, dan delete data adjacency list	Ketika super admin memilih menu adjacency list dan tidak ada data yang diinputkan oleh super admin	Menampilkan halaman adjacency list dengan tabel yang kosong	OK
			Ketika super admin memilih menu adjacency list	Menampilkan data adjacency list dalam bentuk tabel	OK

			dan ada data yang telah diinputkan		
			Ketika superadmin input data adjacency list dan klik simpan	Menampilkan pesan berhasil dan halaman adjacency list	OK
			Ketika super admin klik tombol edit	Menampilkan form edit	OK
			Ketika super admin mengubah data adjacency list dan klik simpan	Menampilkan pesan berhasil dan halaman adjacency list	OK
			Ketika super admin klik tombol <i>delete</i>	Menampilkan pesan “Apakah anda yakin menghapus data ini?”	OK
			Ketika superadmin memilih tombol ya	Menghapus data yang dipilih dan menampilkan halaman data <i>user</i>	OK

			Ketika superadmin memilih tombol tidak	Menampilkan halaman data <i>user</i>	OK
4.	View hasil perhitungan algoritma <i>greedy</i> dan metode <i>dominating set</i>	Menu ini memiliki fitur untuk menghitung data sumur dan data adjacency list, menampilkan hasil perhitungan menggunakan algoritma <i>greedy</i> dan menampilkan hasil <i>dominating set</i>	Ketika admin klik analisis	Menampilkan perhitungan tahap 1	OK
			Ketika super admin klik data titik	Menampilkan pop up detail status titik	OK
			Ketika super admin klik tombol next	Menampilkan hasil perhitungan tahap 2	OK

			pada halaman tahap 1		
			Ketika super admin klik tombol next pada halaman tahap 2	Menampilkan hasil perhitungan tahap 3	OK
			Ketika super admin klik tombol next pada halaman tahap 3	Menampilkan hasil perhitungan tahap 4	OK
			Ketika super admin klik tombol next pada halaman tahap 4	Menampilkan hasil perhitungan <i>dominating set</i>	OK

BAB 5. HASIL DAN PEMBAHASAN

Bab ini menjelaskan mengenai hasil penelitian yang telah dilakukan serta pembahasan sistem yang telah dibuat. Pembahasan dilakukan guna menjelaskan dan memaparkan bagaimana penelitian ini menjawab perumusan masalah serta tujuan dan manfaat dari penelitian ini seperti yang telah ditentukan pada awal penelitian.

4.5 Hasil Implementasi *Coding* pada Sistem Informasi Penentu *Dominating set* Pada Graf Sumur Resapan di Universitas Jember

Hasil implementasi sistem informasi penentu *dominating set* pada sumur resapan yang dibangun pada penelitian ini terdiri atas beberapa fitur yang dapat diakses oleh pengguna aplikasi. Sistem ini dapat memudahkan para pengguna untuk mengetahui letak sumur resapan dan membantu dalam penentuan lokasi sumur resapan yang akan dibangun. Sistem ini dapat diakses oleh tiga user yaitu super admin, admin graf dan admin sumur sebagai pengguna sistem. Sistem informasi ini memiliki beberapa fitur sebagai berikut.

4.5.1 Tampilan Halaman *Login*

Halaman *Login* merupakan halaman ketika super admin, admin graf maupun admin sumur akan mengakses sistem informasi ini. Dalam tampilan halaman login aktor diharuskan memasukkan *username*, *password*, dan *captcha* untuk mengakses sistem informasi sesuai dengan hak akses yang dimiliki. Halaman *login* pada sistem informasi ini ditunjukkan dengan Gambar 5.1.



Gambar 5. 1 Tampilan Halaman *Login*

4.5.2 Tampilan Halaman *Dashboard*

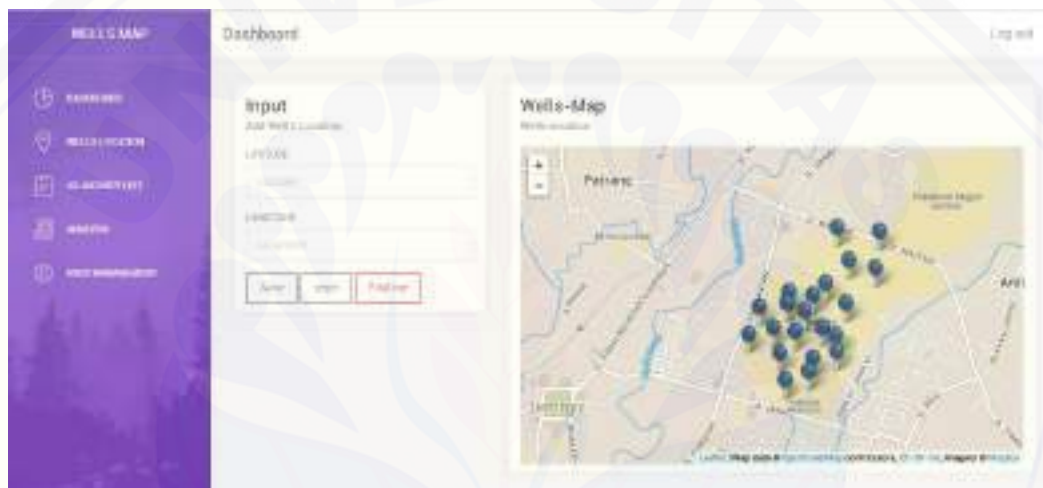
Tampilan halaman *dashboard* merupakan halaman utama yang ada ketika super admin *login* untuk mengakses menu yang dipunyai oleh super admin. Tampilan *dashboard* berupa panel yang berisi profil singkat universitas jember dan peta universitas jember. Tampilan halaman dashboard dapat dilihat pada Gambar 5.2.



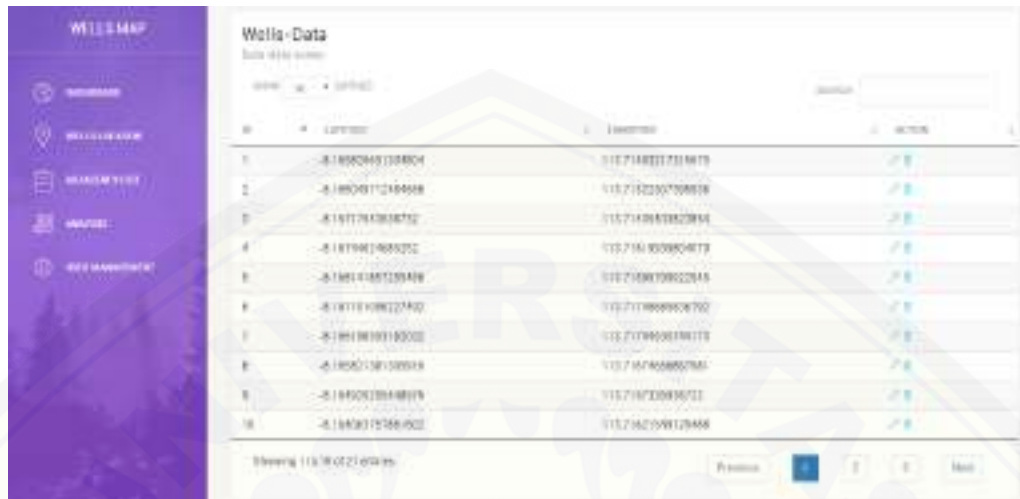
Gambar 5. 2 Tampilan Halaman *Dashboard*

4.5.3 Tampilan Halaman *Wells Location*

Halaman menu *wells location* merupakan menu yang digunakan untuk *input*, *update*, dan *delete* data sumur. Pada halaman ini berisi *form input* lokasi sumur, peta Universitas Jember, dan tabel data sumur. Pada *form input* lokasi sumur terdapat atribut *latitude* dan *longitude*. Detail lokasi sumur dapat dilihat pada *pop up marker* yang ada di peta. Halaman *wells location*, tampilan detail titik sumur, tampilan edit data sumur, dan tampilan *delete* data sumur ditunjukkan oleh gambar 5.3 sampai gambar 5.7.



Gambar 5. 3 Tampilan Halaman *Wells Location*

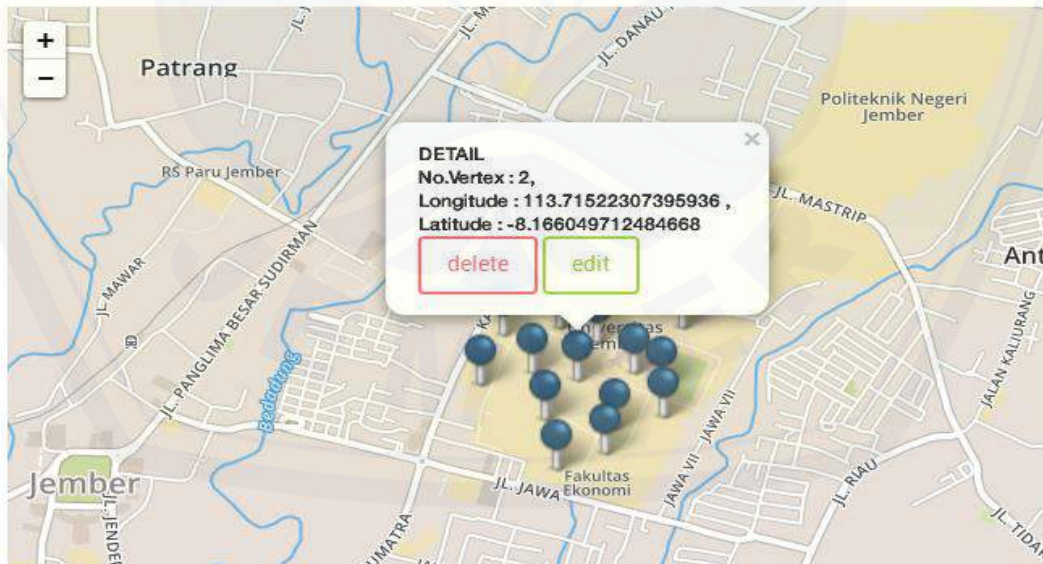


The screenshot shows a web application interface for 'Wells-Data'. On the left is a purple sidebar with navigation icons for 'HOME', 'WELLS MAP', 'WELLS DATA', 'WELLS INFO', 'WELLS', and 'WELLS HISTORY'. The main content area is titled 'Wells-Data' and contains a table with columns for 'No', 'Longitude', 'Latitude', and 'Action'. The table lists 10 data points. Below the table, there is a search bar and a 'Refresh' button.

No	Longitude	Latitude	Action
1	-8.1662049130804	113.7148217318479	[edit] [delete]
2	-8.1660491248468	113.71522307395936	[edit] [delete]
3	-8.167178308712	113.7148058520854	[edit] [delete]
4	-8.167941485252	113.714100080473	[edit] [delete]
5	-8.166148512948	113.7148070822845	[edit] [delete]
6	-8.1671810822432	113.711888958702	[edit] [delete]
7	-8.1661883918032	113.7179639184175	[edit] [delete]
8	-8.1660138138938	113.7157468862881	[edit] [delete]
9	-8.1660238448918	113.7147228936711	[edit] [delete]
10	-8.1660312484932	113.7142598128488	[edit] [delete]

Gambar 5. 4 Tampilan Tabel Data Sumur

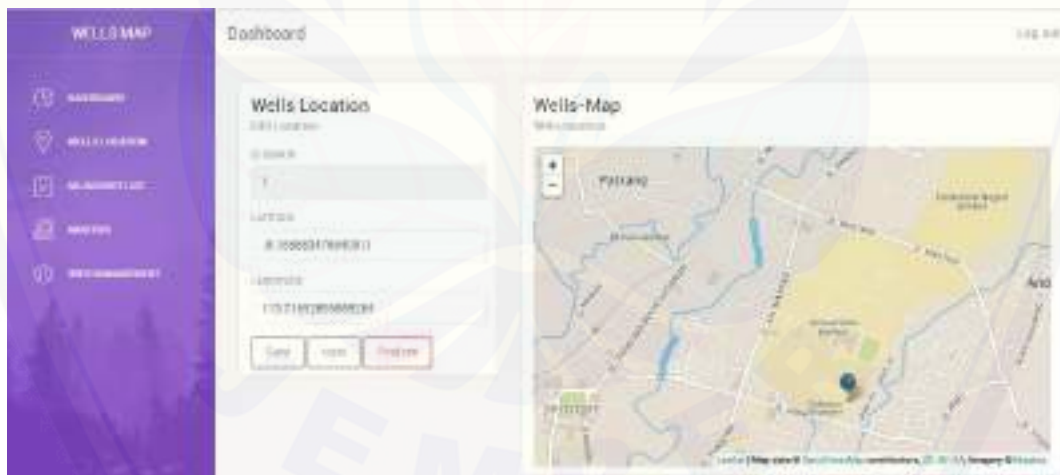
Wells-Map Wells location



Gambar 5. 5 Tampilan Detail Titik



Gambar 5. 6 Tampilan *Delete* Data Sumur



Gambar 5. 7 Tampilan edit data sumur

4.5.4 Tampilan Halaman *Adjacency List*

Halaman menu *adjacency list* merupakan menu yang digunakan untuk *input*, *update*, dan *delete* data *adjacency list*. Halaman ini dapat diakses oleh super admin dan admin graf. Halaman *Adjacency List*, tampilan *edit data list*, dan tampilan *delete data list* dapat dilihat pada gambar 5.8 sampai 5.10.

ID	WELLS	WELLS2	WELLS3	ACTION
1	1	2		/3
2	1	3		/3
4	1	18		/3
8	2	1		/3
6	2	3		/3
10	2	8		/3
11	3	8		/3
12	3	1		/3
13	3	2		/3
15	3	4		/3

Gambar 5. 8 Tampilan Tabel Data *Adjacency List*



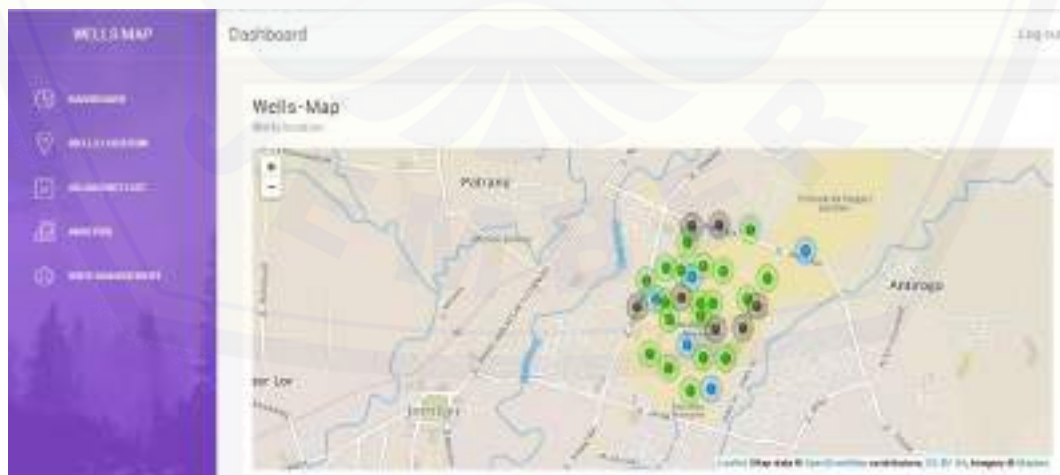
Gambar 5. 9 Tampilan Halaman *Adjacency List*



Gambar 5. 10 Tampilan edit data *Adjacency List*

4.5.5 Tampilan Halaman *Analysis*

Halaman menu *Analysis* ini merupakan halaman yang digunakan untuk mengelola data sumur dengan perhitungan algoritma *greedy* dan *dominating set*. Pada halaman ini ditampilkan hasil dari perhitungan *greedy* dan *dominating set*. Halaman *Analysis* dapat dilihat pada gambar.



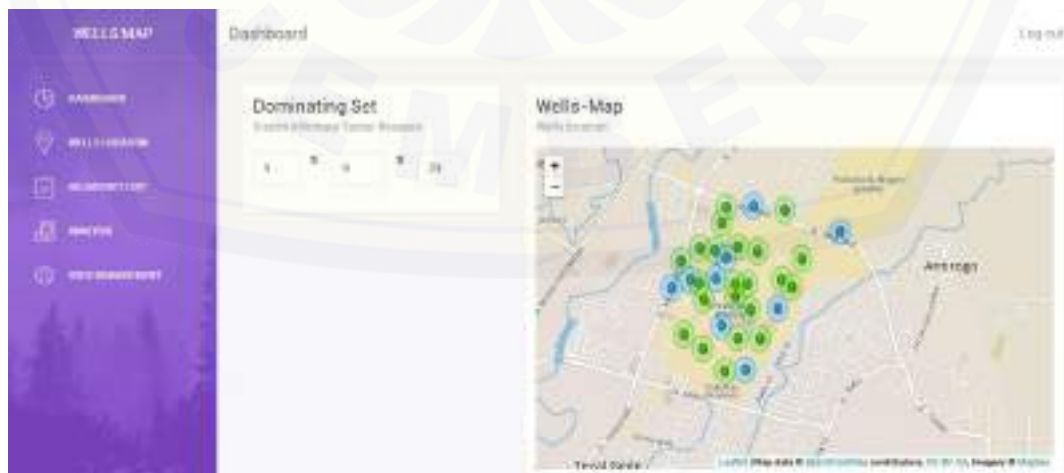
Gambar 5. 11 Tampilan Halaman *Analysis* Tahap 2



Gambar 5. 14 Tampilan Halaman *Analysis*



Gambar 5. 13 Tampilan Halaman *Analysis Tahap 3*



Gambar 5. 12 Halaman *Analysis Tahap 4*

4.5.6 Tampilan Halaman *User Management*

Halaman *User Management* ini merupakan halaman yang digunakan untuk *input*, *edit*, dan *delete* data *user*. Halaman ini hanya bisa diakses oleh super admin. Halaman *user management* dapat dilihat pada gambar, halaman *form edit user* dapat dilihat pada gambar, dan *delete user* dapat dilihat pada Gambar 5.15 sampai 5.17.



Gambar 5. 16 Tampilan Halaman *User Management*



Gambar 5. 15 Tampilan *Delete Data User*



Gambar 5. 17 Tampilan Halaman Edit Data User

4.6 Hasil Implementasi Algoritma *Greedy* Pada Sistem Informasi Penentuan *Dominating set* (Studi kasus: Penentuan Sumur Resapan di Universitas Jember)

Dominating set merupakan salah satu teorema dalam graf yang digunakan untuk menentukan titik yang mendominasi titik lainnya. Dalam penelitian ini *dominating set* digunakan untuk menentukan sumur resapan yang mendominasi sumur resapan lain yang akan dibangun di Universitas Jember. Penentuan letak sumur resapan menggunakan kombinasi model matematis *dominating set* yang disesuaikan dengan pertimbangan admin graf dan admin sumur. Penentuan titik dilakukan oleh admin sumur untuk kemudian dijadikan variabel titik. Pencarian titik yang mendominasi titik lain menggunakan algoritma *greedy*. Langkah pertama adalah menyeleksi titik dan tujuan berdasarkan titik dengan derajat terbesar yang ditunjukkan pada gambar 5.18.

```

foreach ($list_titik as $l)
{
    $tujuan = $this->a->get_tujuan($l["id_sumur"]);
    $available = false;
    foreach ($list_tujuan as $lt)
    {
        foreach ($tujuan as $t)
        {
            // echo $lt["tujuan"] . " | " . $t["titik"] . " | " . $t["tujuan"] . "<br />";
            if ($lt["list"] == $t["id_sumur"] || $lt["list"] == $t["id_sumur"])
            {
                $available = true;
            }
        }
    }
    if (!$available)
    {
        $anu = "";
        foreach ($tujuan as $t)
        {
            array_push($list_tujuan, array('id_sumur' => $t["id_sumur"], "list" => $t["list"]));
            $anu .= $t["list"].",";
            $this->dominatorSamaYangTerpakai[] = [
                'jenis' => 'terpakai',
                'isi' => $this->a->findVertex($t["list"])[0],
                'dominator' => $l["id_sumur"]
            ];
        }
        $this->dominatorSamaYangTerpakai[] = [
            'jenis' => 'dominator',
            'isi' => $l,
            'list' => $anu
        ];
    }
}
}

```

Gambar 5. 18 Kode Program Tahap 1 Algoritma *Greedy*

Setelah melakukan pencarian tahap 1, maka melakukan pencarian tahap 2 pada algoritma *greedy*, yaitu mencari titik yang terpakai dan menentukan titik yang tersisa seperti yang ditunjukkan pada gambar 5.19 dan gambar 5.20.

```

$indek = 0;
$titik = 0;
$l = true;
foreach($list_tujuan as $row)
{
    if($titik == $row['id_sumur'] && $l)
    {
        $temp[$indek++] = $row['id_sumur'];
        $l = false;
    }

    if($titik != $row['id_sumur'] && $titik != 0)
    {
        $titik = $row['id_sumur'];
        $l = true;
    }else if($titik == 0)
    {
        $titik = $row['id_sumur'];
    }

    $temp[$indek++] = $row['list'];
}

```

Gambar 5. 19 Kode Program tahap 2 *Algoritma greedy*

```

$tidak = array();
$list_tidak = array();
$count = 0;
foreach($list_titik as $list)
{
    foreach($temp as $t)
    {
        if($list['id_sumur'] == $t)
        {
            $count++;
        }
    }
    if($count == 0)
    {
        array_push($tidak,$list['id_sumur']);
        $this->listTidak[] = [
            'jenis' => 'tidak',
            'isi' => $this->a->findVertex($list['id_sumur'])[0]
        ];
    }
    $count = 0;
}
}

```

Gambar 5. 20 Kode Program tahap 2 *Algoritma Greedy*

Langkah selanjutnya adalah Menentukan titik serta tujuan yang tersisa (belum tercover) yang ditunjukkan pada gambar 5.21.

```

$titik_bantu = array();
foreach($list as $list)
{
    $kosongSemua = true;
    $tujuan = $this->get_tujuan($list);
    for($i=0;$i<count($tujuan);$i++)
    {
        if(!in_array($tujuan[$i]['list'], $steep))
        {
            $tujuan[$i]['list'] = null;
            $tujuan[$i]['id_sumur'] = null;
        }
    }
    for($i=0;$i<count($tujuan);$i++)
    {
        if($tujuan[$i]['id_sumur'] != "" || $tujuan[$i]['list'] != "")
        {
            array_push($list_tidak, array('id_sumur' => $tujuan[$i]['id_sumur'], 'list' => $tujuan[$i]['list']));
            $kosongSemua = false;
        }
    }
    if ($kosongSemua & count($tujuan) > 0) {
        $this->jobblo[] = [
            'jenis' => 'jobblo',
            'titik' => $this->findVertes($list)[0]
        ];
    }
}

```

Gambar 5. 21 Kode Program tahap 3 *Algoritma Greedy*

Langkah selanjutnya adalah menentukan *dominating set* pada titik dan tujuan yang tersisa yang ditunjukkan gambar 5.22.

```

$list_tidak_dominan = array();
$count = 0;
for($i=0;$i<count($list_tidak);$i++)
{
    if($i==0)
    {
        $titik_bantu = $list_tidak[$i]['id_sumur'];
    }
    if($list_tidak[$i]['id_sumur'] == $titik_bantu )
    {
        $count++;
    }
    else
    {
        $list_tidak_dominan[$titik_bantu] = $count;
        $count = 1;
        $titik_bantu = $list_tidak[$i]['id_sumur'];
    }
}

arsort($list_tidak_dominan);

```

Gambar 5. 22 Kode Program tahap 4 *Algoritma Greedy*

Langkah terakhir pada pencarian algoritma titik adalah mencari titik yang berdiri sendiri dan menyederhanakan hasil kedalam variabel dominan dan tujuan yang ditunjukkan pada gambar 5.23.

```

foreach($tidak as $t)
{
    if(!in_array($t, $temp))
    {
        array_push($list_tujuan, array('id_sumur' => $t, 'list' => $t));
        // echo $t." ";
    }
}

$data['dominan'] = array();

foreach($list_tujuan as $list)
{
    if(!in_array($list['id_sumur'], $data['dominan']))
    {
        array_push($data['dominan'], $list['id_sumur']);
    }
}
    
```

Gambar 5. 23 Kode Program tahap 5 Algoritma Greedy

Setelah pencarian titik *dominating set* menggunakan algoritma *greedy* selesai, maka tahap selanjutnya adalah perhitungan keoptimalah hasil menggunakan kode program penerapan rumus *dominating set* seperti yang ditunjukkan pada gambar 5.24.

```

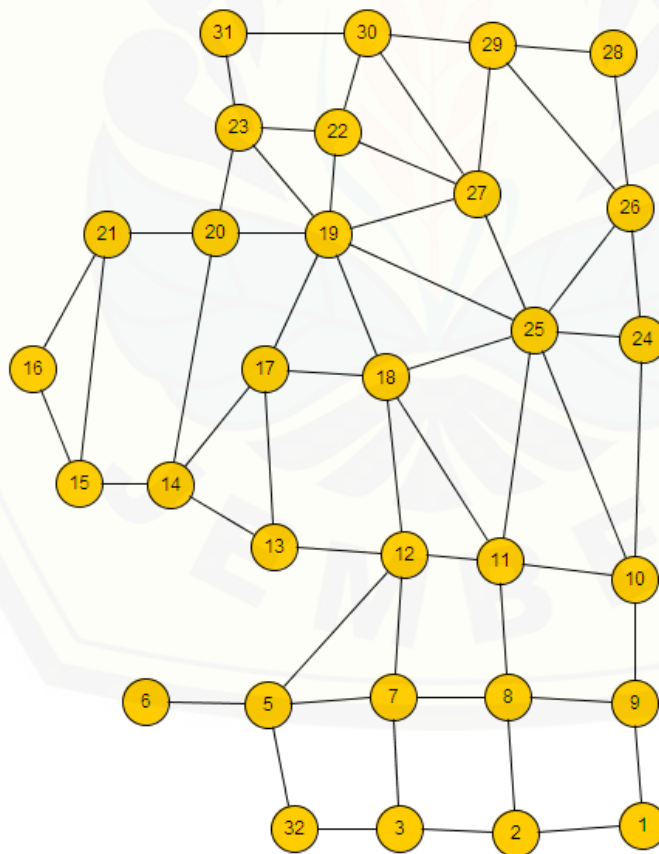
352 public function dominating(){
353     $this->load->library('../controllers/cMetode');
354     $this->index();
355     $hmm = array_merge($this->dominatorSamaYangTerpakai,$this->sisane,$this->jomblo);
356     $data['nodes'] = $hmm;
357     $titik = $this->index();
358     $where = "where id_sumur in (";
359     $p = $this->a->get_semua_titik();
360     $g = $this->a->getMax();
361     $data['min'] = round($p / (1+$g));
362     $data['ds'] = count($this->index());
363     $data['max'] = $p - $g;
364     for($i=0;$i<count($titik);$i++)
365     {
366         $where .= $titik[$i];
367         if($i < count($titik) - 1)
368         {
369             $where .= ",";
370         }
371         else
372         {
373             $where .= ")";
374         }
375     }
376     $this->load->view('sidebar');
377     $this->load->view('ldominating',$data);
378     $this->load->view('footer',$data);
379 }
    
```

Gambar 5. 24 Kode Program Dominating set

4.7 Pengujian Analisis Sistem Informasi Penentu *Dominating set* pada Sumur Resapan di Universitas Jember

Pengujian analisis sistem informasi penentu *dominating set* sumur resapan didapatkan dari hasil perhitungan manual dan hasil perhitungan sistem yang ada. Pengujian ini dilakukan untuk melihat keakurasian menggunakan metode *dominating set* dan algoritma *greedy*.

Data yang diuji adalah data titik lokasi sumur resapan di Universitas Jember yang telah diinputkan kedalam peta hasil *tracking*. Pencarian titik untuk mendapatkan himpunan pendominasi menggunakan algoritma *greedy*. Representasi titik dan sisi dalam bentuk graf dapat dilihat pada Gambar 5.25.



Gambar 5. 25 Graf Data Sumur

Tahap pertama yang dilakukan adalah mengurutkan data sumur berdasarkan jumlah derajat dari terbesar ke terkecil, hasil perhitungan tahap pertama dapat dilihat pada Tabel 5.1.

Tabel 5. 1 Tabel hasil perhitungan tahap 1

TAHAP 1			
id_sumur	jmlderajat	Latitude	Longitude
19	7	-8.162.768.102.998.220	11.371.589.899.063.100
25	7	-8.163.893.836.151.750	11.371.888.160.705.500
12	6	-8.164.817.784.576.750	11.371.639.251.708.900
7	5	-8.166.315.213.691.120	11.371.564.149.856.500
11	5	-8.165.508.089.476.070	11.371.720.790.863.000
17	5	-8.164.127.478.484.140	1.137.164.032.459.250
18	5	-816.411.685.838.109	11.371.702.551.841.700
27	5	-8.162.407.018.107.110	11.371.759.414.672.800
5	4	-8.164.977.085.813.270	11.371.471.881.866.400
8	4	-8.166.994.895.975.120	11.371.649.980.545.000
10	4	-816.542.312.893.748	11.371.866.703.033.400
14	4	-8.164.074.377.966.050	11.371.446.132.659.900
20	4	-8.162.396.397.958.310	11.371.530.890.464.700
22	4	-8.162.152.134.458.060	11.371.651.053.428.600
23	4	-8.160.941.434.904.230	11.371.564.149.856.500
26	4	-8.162.799.963.414.120	11.371.995.449.066.100
29	4	-816.026.174.231.863	11.371.906.399.726.800
30	4	-8.160.017.477.513.250	1.137.173.902.988.430
2	3	-8.168.757.816.506.060	11.371.583.461.761.400
3	3	-8.167.568.377.002.210	11.371.467.590.332.000
9	3	-8.167.058.616.129.930	11.371.772.289.276.100
13	3	-8.163.846.045.657.840	11.371.537.864.208.200
15	3	-8.163.925.696.477.860	11.371.399.998.664.800
21	3	-8.162.247.715.845.510	11.371.451.497.077.900
24	3	-8.164.276.159.897.220	11.371.945.023.536.600
32	3	-8.166.803.735.449.710	1.137.136.459.350.580
1	2	-8.168.683.476.640.910	11.371.692.895.889.200
16	2	-81.629.274.050.523	11.371.345.818.042.700
28	2	-8.161.376.862.359.240	11.372.199.296.951.200
31	2	-8.160.081.198.781.230	1.137.159.311.771.390

6	1	-8.164.339.880.485.880	11.371.300.220.489.500
---	---	------------------------	------------------------

Tahap selanjutnya adalah menentukan titik dominator berdasarkan list titik yang dimilikinya, apabila list titik yang dimilikinya belum didominasi oleh titik lain maka titik itu menjadi titik dominator. Proses pemilihan titik dapat dilihat pada Tabel 5.2.

Tabel 5. 2 Hasil Penhitungan Tahap 2

TAHAP 2			
id_sumur	jmlderajat	List	Keterangan
19	7	17,18,25,20,23,22,27	dominator
25	7	24,10,11,18,19,27,26	tidak
12	6	17,18,7,5,13,11	tidak
7	5	3,5,8,12,32	dominator
11	5	8,10,12,25,18	tidak
17	5	12,13,18,19,14	tidak
18	5	17,12,25,19,11	tidak
27	5	25,22,19,29,30	tidak
5	4	6,12,7,32	tidak
8	4	2,9,11,7	tidak
10	4	9,11,24,25	tidak
14	4	13,15,20,17	tidak
20	4	21,19,14,23	tidak
22	4	23,27,19,30	tidak
23	4	22,20,19,31	tidak
26	4	24,25,28,29	tidak
29	4	26,27,28,30	tidak
30	4	22,27,29,31	tidak
2	3	1,3,8	tidak
3	3	2,7,32	tidak
9	3	1,8,10	tidak
13	3	12,14,17	tidak
15	3	14,16,21	dominator
21	3	15,16,20	tidak
24	3	10,25,26	tidak
32	3	3,5,7	tidak
1	2	2,9	dominator

16	2	15,21	tidak
28	2	26,29	dominator
31	2	30,23	tidak
6	1	5	tidak

Pada Tabel 5.2 list yang ditandai dengan warna merah merupakan titik yang telah dicover oleh titik lain yang menjadi dominator. Pada tahap selanjutnya, dilakukan pencarian titik sisa yang belum tercover. Proses pemilihan titik sisa dapat dilihat pada tabel 5.3.

Tabel 5. 3 Hasil Perhitungan Tahap 3

TAHAP 3			
id_sumur	jmlderajat	List	Keterangan
19	7	17,18,25,20,23,22,27	dominator
25	7	24,10,11,18,19,27,26	tidak
12	6	17,18,7,5,13,11	tidak
7	5	3,5,8,12,32	dominator
11	5	8,10,12,25,18	tidak, sisa
17	5	12,13,18,19,14	tidak
18	5	17,12,25,19,11	tidak
27	5	25,22,19,29,30	tidak
5	4	6,12,7,32	tidak
8	4	2,9,11,7	tidak
10	4	9,11,24,25	tidak, sisa
14	4	13,15,20,17	tidak
20	4	21,19,14,23	tidak
22	4	23,27,19,30	tidak
23	4	22,20,19,31	tidak
26	4	24,25,28,29	tidak
29	4	26,27,28,30	tidak
30	4	22,27,29,31	tidak, sisa
2	3	1,3,8	tidak
3	3	2,7,32	tidak
9	3	1,8,10	tidak
13	3	12,14,17	tidak, sisa
15	3	14,16,21	dominator

21	3	15,16,20	tidak
24	3	10,25,26	tidak, sisa
32	3	3,5,7	tidak
1	2	2,9	dominator
16	2	15,21	tidak
28	2	26,29	dominator
31	2	30,23	tidak, sisa
6	1	5	tidak, sisa

Pada tabel 5.3 dapat dilihat id_sumur yang berwarna biru merupakan titik yang belum tercover oleh titik dominasi. Tahap selanjutnya adalah mencari titik dominasi dari titik sisa yang belum terdominasi. Proses pemilihan dominasi titik sisa dapat dilihat pada Tabel 5.4.

Tabel 5. 4 Hasil Perhitungan Tahap 4

Tahap 4			
id_sumur	jmlderajatsisa	list	keterangan
11	1	8,10,12,25,18	tidak
10	2	9,11,24,25	sisa dominator
30	1	22,27,29,31	sisa dominator
13	0	12,14,17	sendiri
24	1	10,25,26	tidak
31	1	30,23	tidak
6	0	5	sendiri

Pada tabel 5.4 dapat dilihat bahwa yang menjadi titik dominator adalah id_sumur 10 dan 30. Id_sumur yang berwarna kuning adalah titik yang telah dicover oleh dominator sisa. Id_sumur 13 dan 6 tidak memiliki list anggota maka ditetapkan sebagai dominator tanpa list. Tahap Selanjutnya adalah menggabungkan semua titik yang menjadi dominator kedalam himpunan dominasi. Hasil yang didapatkan dari pencarian menggunakan algoritma *greedy* terdapat 9 titik dominasi. Tahap selanjutnya adalah menghitung optimalisasi hasil *dominating set* menggunakan rumus.

Diketahui:

$$p = 31, \Delta(G) = 7, \gamma(G) = 9$$

Hasil:

$$\left\lceil \frac{p}{1+\Delta(G)} \right\rceil \leq \gamma(G) \leq p - \Delta(G) = \left\lceil \frac{31}{1+7} \right\rceil \leq 9 \leq 31 - 7 = 24 \leq 9 \leq 24$$

Keterangan:

p : jumlah titik-titik yang ada pada sebarang graf

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

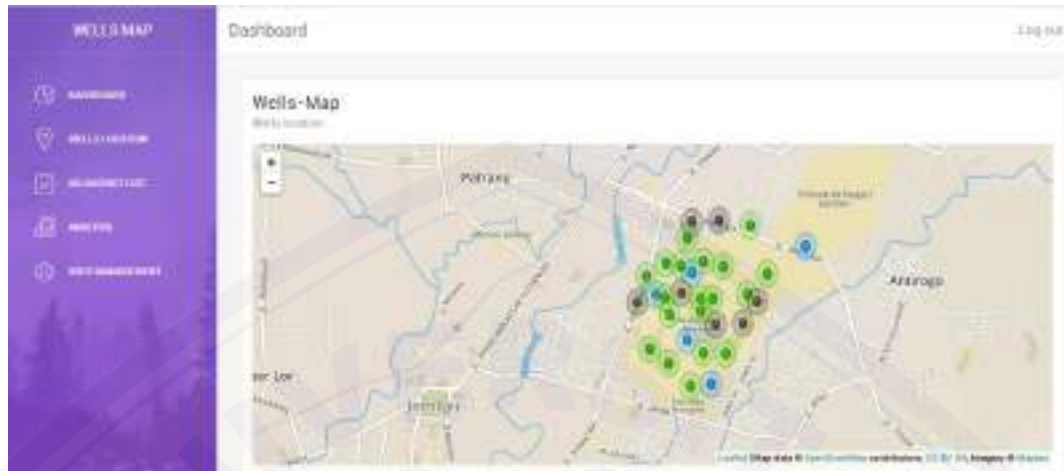
$\gamma(G)$: jumlah perkiraan *dominating set*

Tabel-tabel diatas merupakan hasil proses penghitungan menggunakan metode *dominating set* dan algoritma *greedy* secara manual, sedangkan perhitungan dalam sistem ditunjukkan pada gambar 5.26 sampai dengan gambar 5.29.



Gambar 5. 26 Hasil Perhitungan Tahap 1 dan Tahap 2

Gambar 5.26 merupakan proses dari tahap pertama pencarian titik dengan mengurutkan titik berdasarkan derajat dari besar ke kecil dan memilih titik yang dijadikan *dominating set* dilihat dari list titik yang dimiliki.



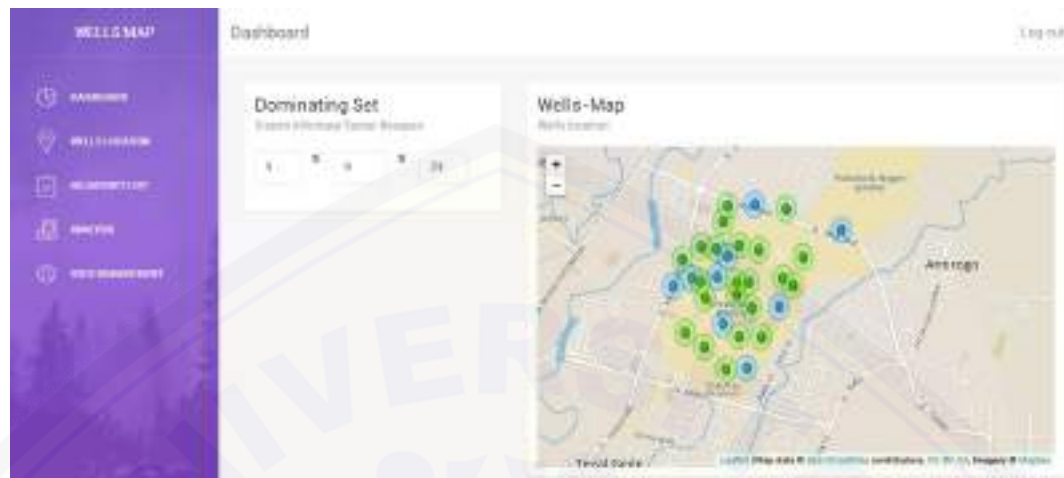
Gambar 5. 28 Hasil Perhitungan Tahap 3

Gambar 5.27 merupakan proses perhitungan untuk mencari titik yang belum tercover.



Gambar 5. 27 Hasil Perhitungan Tahap 4

Gambar 5.28 merupakan proses perhitungan untuk mencari titik yang mendominasi dari titik sisa.



Gambar 5. 29 Hasil Perhitungan Tahap 5

Gambar 5.29 merupakan hasil perhitungan algoritma *greedy* dan hasil penilaian *dominating set*.

4.8 Pembahasan

4.8.1 Pembahasan Hasil Implementasi Algoritma *Greedy* Pada Sistem Informasi Penentuan *Dominating set* (Studi kasus: Penentuan Sumur Resapan di Universitas Jember)

Implementasi metode *dominating set* pada sistem informasi penentu letak sumur resapan terkait tujuan dari penelitian ini, yaitu untuk mengetahui titik pendominasi pada sumur resapan di Universitas Jember. Untuk mencapai tujuan tersebut sistem ini memiliki fitur utama yaitu fitur yang menerapkan perhitungan *dominating set* dengan pencarian menggunakan algoritma *greedy*. Data yang digunakan dalam fitur ini adalah data sumur dan data *adjacency list*.

Data sumur didapatkan dari data lokasi peletakan marker pada peta universitas jember sesuai lokasi yang akan dibangun sumur resapan. Kriteria penempatan titik untuk data sumur adalah satu lahan satu titik, satu lahan merupakan representasi hasil *tracking* yang membentuk lahan dengan batas garis. Sisi untuk menghubungkan setiap

titik dibuat berdasarkan batas garis lahan yang berhimpitan dengan batas garis lahan lain. Sisi merupakan derajat dari sebuah titik. Data *adjacency list* didapat dari titik sumur yang berhubungan dengan titik sumur lain.

Kemampuan sistem dalam mencapai tujuan yang telah ditentukan di awal masih terdapat banyak bagian yang harus diperbaiki dan dikembangkan lagi. Sistem yang dibangun seharusnya memperhatikan parameter lain yang mempengaruhi penentuan lokasi pembuatan sumur resapan, seperti tingkat permeabilitas tanah, luas lahan, jarak antar sumur, kondisi lahan, dan lain lain. Pada sistem ini, peletakan lokasi sumur resapan dilakukan oleh pakar sumur, dimana pakar sumur telah mempertimbangan titik lokasi yang akan dibangun sesuai dengan kebutuhan. Pertimbangan yang dilakukan pakar sumur tidak dilakukan oleh sistem, maka hal ini dapat menjadi kekurangan dari sistem yang dibuat dan dapat ditambahkan pada pengembangan yang selanjutnya.

Pada pembangunan sistem informasi penentu letak sumur resapan, perbandingan perhitungan manual dan sistem adalah sama. Baik perhitungan pencarian titik algoritma *greedy* maupun perhitungan *dominating set* memiliki hasil yang sama dan optimal. Berikut adalah hasil perhitungan *dominating set* untuk menilai keoptimalan hasil pencarian titik.

Diketahui:

$$p = 31, \Delta(G) = 7, \gamma(G) = 9$$

Hasil:

$$\left\lceil \frac{p}{1+\Delta(G)} \right\rceil \leq \gamma(G) \leq p - \Delta(G) = \left\lceil \frac{31}{1+7} \right\rceil \leq 9 \leq 31 - 7 = 24 \leq 9 \leq 24$$

Dari perhitungan tersebut dapat disimpulkan bahwa hasil pencarian titik menggunakan algoritma *greedy* telah optimal, karena mendekati hasil minimum *dominating set* pada data sumur akan dibangun. Lokasi titik sumur yang menjadi *dominating set* dapat dilihat pada tabel 5.5.

Tabel 5. 5 Tabel Lokasi Sumur Hasil *Dominating set*

sumur	List	Latitude	Longitude
19	17,18,25,20,23,22,27	-8.162.768.102.998.220	11.371.589.899.063.100
7	3,8,32,5,12	-8.166.315.213.691.120	11.371.564.149.856.500
15	16,14,21	-8.163.925.696.477.860	11.371.399.998.664.800
1	2,9	-8.168.683.476.640.910	11.371.692.895.889.200
28	29,26	-8.161.376.862.359.240	11.372.199.296.951.200
10	11,24	-816.542.312.893.748	11.371.866.703.033.400
30	31	-8.160.017.477.513.250	1.137.173.902.988.430
13	-	-8.163.846.045.657.840	11.371.537.864.208.200
6	-	-8.164.339.880.485.880	11.371.300.220.489.500

Hasil pencarian titik menggunakan algoritma *greedy* masih belum di bandingkan dengan hasil pencarian menggunakan algoritma lain yang dapat memberikan hasil yang lebih optimal. Hasil *dominating set* dikatakan optimal jika mendekati nilai *minimum dominating set*.

Dengan adanya sistem prediksi ini, maka mempermudah pengguna memperoleh hasil penentuan lokasi sumur resapan yang akan dibangun. Admin sumur dapat mengetahui lokasi sumur mana saja yang akan dibangun melalui peta yang disediakan oleh sistem. Admin graf dapat menganalisis secara matematis dari graf yang terbentuk menggunakan *dominating set* yang disediakan oleh sistem.

4.8.2 Pembahasan Hasil Pengujian Sistem

Pengujian program dalam penelitian ini dilakukan dengan menggunakan dua metode pengujian sistem, yaitu metode pengujian *white box* dan *black box*. Pada pengujian *white box* menggunakan *cyclometric complexity* untuk menghitung kompleksitas suatu method. Hasil dari pengujian *white box* dengan *listing* program sesuai dengan rumusan pada metode *dominating set*. Pengujian *white box* yang dilakukan pada penelitian ini meliputi *listing* program, grafik alir, kompleksitas siklomatis, basis set dan *test case*. Metode sudah berjalan dan menghasilkan respon yang diharapkan. Kompleksitas suatu method dapat diuji dengan mengetahui berapa

jumlah *test case* yang diperlukan untuk mengevaluasi *method* tersebut sehingga setiap percabangan didalam *method* tersebut pernah dilalui. Semakin banyak jumlah *test case* maka *method* dapat dikatakan semakin kompleks.

Sistem informasi penentu letak sumur resapan ini jumlah *test case* yang dihasilkan adalah 1 buah. Sehingga sistem ini tingkat kompleksitasnya dapat dikatakan cukup rendah. Metode pengujian *black box* digunakan untuk mengetahui apakah fungsi-fungsi, inputan, dan keluaran sistem sesuai dengan spesifikasi yang dibutuhkan oleh pengguna. Maka berdasarkan pengujian *black box* yang telah dilakukan, dapat diketahui bahwa sistem informasi penentu letak sumur resapan menggunakan metode *dominting set* telah berhasil dibangun sesuai dengan kebutuhan fungsional.

BAB 6. PENUTUP

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

6.1 Kesimpulan

Kesimpulan dari hasil penelitian yang telah dilakukan oleh peneliti adalah sebagai berikut:

1. Sistem Informasi penentu lokasi sumur resapan ini mampu memberikan hasil pencarian lokasi menggunakan algoritma *greedy* dan hasil perhitungan optimalisasi menggunakan *dominating set* sebagai.
2. Hasil perhitungan metode *dominating set* diperoleh dari hasil pencarian menggunakan algoritma *greedy* pada data sumur dan data *adjacency list*.
3. Pencarian titik menggunakan algoritma *greedy* dilakukan dalam beberapa tahap. Tahap pertama adalah mengurutkan data sumur berdasarkan jumlah derajatnya dari besar ke kecil. Tahap kedua adalah mencari titik yang mendominasi dengan mempertimbangan titik listnya, pada proses ini menghasilkan 5 titik dominator. Tahap ketiga adalah mencari titik yang belum tercover, pada tahap ini menghasilkan 7 titik sisa. Tahap keempat adalah mencari titik yang mendominasi dari titik sisa, pada tahap ini menghasilkan 4 titik dominator dimana 2 titiknya tidak memiliki list anggota. Selanjutnya adalah menggabungkan seluruh hasil dominator kedalam himpunan dominasi, hasil dari seluruh proses adalah 9 titik dominator yang menjadi lokasi sumur resapan.

4. Hasil perhitungan metode *dominating set* memperlihatkan bahwa himpunan dominasi yang dihasilkan telah optimal dan mendekati nilai minimum *dominating set*.

6.2 Saran

Beberapa saran dan masukan berikut diharapkan dapat memberikan perbaikan sistem dalam penelitian selanjutnya, antara lain:

1. Pengembangan lebih lanjut pada penelitian ini diharapkan dapat menambahkan fitur untuk mempertimbangan lokasi sumur resapan yang sesuai pakar ke dalam sistem berdasarkan parameter jarak, luas lahan, kerapatan tanah, dan lain lain.
2. Menggunakan algoritma pencarian yang lain sebagai pembanding dari hasil yang diperoleh pada penelitian ini.
3. Membuat sistem berbasis android agar lebih memudahkan pengguna untuk melakukan *inputing* data secara *mobile*.

DAFTAR PUSTAKA

- Agustin, I. H. (2014). *Penerapan Teori Dominating set dalam Instalasi Client Hub Untuk Jaringan Intranet di Universitas Jember*. Universitas Jember.
- Alanko, S. C. (2011). Computing the Domination Number of Grid Graphs. *The Electronic Journal of Combinatorics*, 18 (141).
- Alfarisi, R. (2014). *Penerapan teknik konstruksi graf, rainbow connection, dan dominating set dalam analisis morfologi jalan*. Universitas Jember.
- Ardiansyah, d. (2010). Implementasi Algoritma *Greedy* untuk Melakukan Graph Coloring: Studi Kasus Peta Propinsi Jawa Timur. *Jurnal Informatika Vol. 4*, No. 2.
- Ayuliana. (2009). Teknik Pengujian Perangkat Lunak.
- Haynes, T. H. (1998). *Fundamentals of Domination in Graphs*. New York: Marcel Dekker, Inc.
- Henning, M. (2008). On Matching and Total Domination in Graphs. *Discrete Mathematics*, 308.
- Iswadi, H. (2011). Batas Atas Bilangan Dominasi Lokasi Metrik Dari Graf Hasil Operasi Korona.
- L. Ruan, e. a. (2004). A *greedy* approximation for minimum connected *dominating sets*. *Theoretical Computer Science* 329, 325 - 330.
- Passa, F. (2010). Aplikasi Algoritma *Greedy* Pada Persoalan Pewarnaan Graf. *MAKALAH IF3051 STRATEGI ALGORITMA*.

Sommerville. (2003). *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.

Zulhidayati, I. (2013). *Aplikasi Algoritma Greedy dan Program Dinamis (Dynamic Programming) Pada Permainan Greedy Spider*. Universitas Pendidikan Indonesia.



LAMPIRAN

LAMPIRAN A

A.1 Skenario Login

Tabel 1 Skenario Use case Login Super Admin

Nama Use Case	Login
Aktor	Super Admin
Deskripsi Singkat	Super admin akan mengakses aplikasi
Prekondisi	Username dan password yang akan digunakan untuk <i>login</i>
Prakondisi	Login berhasil
Flow Events	
Skenario Normal : login	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman <i>login</i>
3. Menginputkan <i>username</i> , <i>password</i> dan <i>captcha</i>	
4. Klik tombol <i>login</i>	
	5. Mengecek <i>captcha</i> dan mengecek ke database
	6. Menampilkan halaman <i>home</i>
Skenario Alternatif : Username dan Password salah	
Aksi Aktor	Reaksi Sistem

4a. Klik tombol login	
	6a. Menampilkan pesan “username atau password salah”
Skenario Alternatif : <i>Captcha</i> salah	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol login	
	6a. Menampilkan pesan “captcha salah”

Tabel 2 Skenario Use Case login Admin sumur

Nama Use Case	Login
Aktor	Admin sumur
Deskripsi Singkat	Super admin akan mengakses aplikasi
Prekondisi	<i>Username</i> dan <i>password</i> yang akan digunakan untuk <i>login</i>
Prakondisi	Login berhasil
Flow Events	
Skenario Normal : login	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	
	2. Menampilkan halaman <i>login</i>
3. Menginputkan <i>username</i> , <i>password</i> dan <i>captcha</i>	
4. Klik tombol <i>login</i>	

	5. Mengecek captcha dan mengecek ke database
	6. Menampilkan halaman <i>home</i>
Skenario Alternatif : Username dan Password salah	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol login	
	6a. Menampilkan pesan “username atau password salah”
Skenario Alternatif : Captcha salah	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol login	
	6a. Menampilkan pesan “captcha salah”

Tabel 3 Skenario Use Case login admin Graf

Nama Use Case	Login
Aktor	Admin Graf
Deskripsi Singkat	Super admin akan mengakses aplikasi
Prekondisi	Username dan password yang akan digunakan untuk <i>login</i>
Prakondisi	Login berhasil
Flow Events	
Skenario Normal : login	
Aksi Aktor	Reaksi Sistem
1. Membuka halaman website	

	2. Menampilkan halaman <i>login</i>
3. Menginputkan <i>username</i> , <i>password</i> dan <i>captcha</i>	
4. Klik tombol <i>login</i>	
	5. Mengecek <i>captcha</i> dan mengecek ke database
	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”
Skenario Alternatif : <i>Captcha</i> salah	
Aksi Aktor	Reaksi Sistem
4a. Klik tombol <i>login</i>	
	6a. Menampilkan pesan “ <i>captcha</i> salah”

A.2 Skenario Pengelolaan Data User

Tabel 4 Pengelolaan Data User – Super Admin

Nama Use Case	Pengelolaan data <i>User</i>
Aktor	Super Admin
Deskripsi Singkat	Super admin akan mengelola data <i>user</i>
Prekondisi	Data <i>User</i>
Prakondisi	1. Data <i>user</i> berhasil ditambahkan

	2. Data <i>user</i> berhasil diubah 3. 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> meliputi tabel data <i>user</i> yang telah diinputkan beserta tombol <i>input</i> data baru, <i>edit</i> dan <i>delete</i>
3. Klik tombol <i>input user</i>	
	4. Menampilkan halaman form tambah <i>user</i>
5. Mengisi form tambah <i>user</i>	
6. Klik tombol <i>save</i>	
	7. Menyimpan ke database
	8. Menampilkan halaman <i>user</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>save</i>	
	6a. Menampilkan pesan “please fill out this field”
Skenario Normal : Edit User	

Aksi Aktor	Reaksi Sistem
3. Klik tombol <i>edit user</i>	
	4. Menampilkan halaman form <i>edit user</i>
5. Mengisi form <i>edit user</i>	
6. Klik tombol <i>save</i>	
	7. Menyimpan ke database
	8. Menampilkan halaman user
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>save</i>	
	6a. Menampilkan pesan “please fill out this field”
Skenario Normal : Delete User	
Aksi Aktor	Reaksi Sistem
3. Klik tombol <i>delete user</i>	
	4. Menampilkan pesan “apakah anda yakin akan menghapus data ini?”
5. Klik ya	
	6. Menghapus data pada database
	7. Menampilkan halaman <i>user</i>
Skenario Alternatif : klik tidak	

Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan halaman <i>user</i>

A.3 Skenario Pengelolaan Data Sumur

Tabel 5 Pengelolaan Data Sumur - Superadmin

Nama Use Case	Pengelolaan data sumur
Aktor	Super admin
Deskripsi Singkat	Super admin akan mengelola data sumur
Prekondisi	Data sumur
Prakondisi	<ol style="list-style-type: none"> 1. Data sumur berhasil ditambahkan 2. Data sumur berhasil diubah 3. Data sumur berhasil dihapus
Flow Events	
Skenario Normal : View Sumur	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>wells location</i>	
	2. Menampilkan halaman <i>wells location</i> meliputi peta dengan data sumur yang telah diinputkan beserta form data sumur, tombol <i>find me</i> , tombol <i>reset</i> , dan tombol <i>delete</i> .
3. Klik lokasi pada peta	

	4. Menampilkan peta dengan <i>marker</i> pada lokasi yang dipilih
	5. Menampilkan form dengan <i>longitude</i> dan <i>latitude</i> lokasi sumur
6. Klik tombol <i>save</i>	
	7. Menyimpan ke database
	8. Menampilkan halaman <i>wells location</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>save</i>	
	6a. Menampilkan pesan “please fill out this field”
Skenario Alternatif : klik reset	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>reset</i>	
	7a. Menampilkan halaman <i>wells location</i> dan menghilangkan marker yang dibuat
Skenario normal : klik find me	
Aksi Aktor	Reaksi Sistem
3a. Klik tombol <i>find me</i>	
	4a. Membuat marker sesuai lokasi <i>user</i> berdasarkan <i>GPS</i>

5a. Klik tombol <i>save</i>	
	6a. Menyimpan ke database
	7a. Menampilkan halaman <i>wells location</i>
Skenario Normal : Edit Sumur	
Aksi Aktor	Reaksi Sistem
3. Klik salah satu marker pada peta	
	4. Menampilkan pop up detail data sumur
5. Klik edit	
	6. Menampilkan peta dengan marker yang bisa di geser
7. Geser marker sesuai lokasi yg diinginkan	
	8. Menampilkan longitude dan latitude marker
9. Klik save	
	10. Menyimpan ke database
	11. Menampilkan halaman <i>well location</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
9a. Klik tombol <i>save</i>	

	10a. Menampilkan pesan “please fill out this field”
Skenario Normal : Delete sumur	
Aksi Aktor	Reaksi Sistem
3. Klik pada sala satu marker	
	4. Menampilkan pop up detail data sumur dan tombol delete
5. Klik delete	
	6. Menghapus dari database
	12. Menampilkan halaman <i>wells location</i>
Skenario Altenatif : klik tidak	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan halaman <i>user</i>

Tabel 6 Pengelolaan Data Sumur - admin sumur

Nama Use Case	Pengelolaan data sumur
Aktor	Admin sumur
Deskripsi Singkat	Admin sumur akan mengelola data sumur
Prekondisi	Data sumur
Prakondisi	1. Data sumur berhasil ditambahkan 2. Data sumur berhasil diubah

	3. Data sumur berhasil dihapus
Flow Events	
Skenario Normal : View Sumur	
Aksi Aktor	Reaksi Sistem
9. Klik menu <i>wells location</i>	
	10. Menampilkan halaman <i>wells location</i> meliputi peta dengan data sumur yang telah diinputkan beserta form data sumur, tombol <i>find me</i> , tombol <i>reset</i> , dan tombol <i>delete</i> .
11. Klik lokasi pada peta	
	12. Menampilkan peta dengan <i>marker</i> pada lokasi yang dipilih
	13. Menampilkan form dengan <i>longitude</i> dan <i>latitude</i> lokasi sumur
14. Klik tombol <i>save</i>	
	15. Menyimpan ke database
	16. Menampilkan halaman <i>wells location</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem

6a. Klik tombol <i>save</i>	
	6a. Menampilkan pesan “please fill out this field”
Skenario Alternatif : klik reset	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>reset</i>	
	7a. Menampilkan halaman <i>wells location</i> dan menghilangkan marker yang dibuat
Skenario normal : klik find me	
Aksi Aktor	Reaksi Sistem
3a. Klik tombol <i>find me</i>	
	4a. Membuat marker sesuai lokasi <i>user</i> berdasarkan <i>GPS</i>
5a. Klik tombol <i>save</i>	
	6a. Menyimpan ke database
	7a. Menampilkan halaman <i>wells location</i>
Skenario Normal : Edit Sumur	
Aksi Aktor	Reaksi Sistem
13. Klik salah satu marker pada peta	
	14. Menampilkan pop up detail data sumur
15. Klik edit	

	16. Menampilkan peta dengan marker yang bisa di geser
17. Geser marker sesuai lokasi yg diinginkan	
	18. Menampilkan longitude dan latitude marker
19. Klik save	
	20. Menyimpan ke database
	21. Menampilkan halaman <i>well location</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
9a. Klik tombol <i>save</i>	
	10a. Menampilkan pesan “please fill out this field”
Skenario Normal : <i>Delete</i> sumur	
Aksi Aktor	Reaksi Sistem
7. Klik pada sala satu marker	
	8. Menampilkan pop up detail data sumur dan tombol delete
9. Klik delete	
	10. Menghapus dari database

	22. Menampilkan halaman <i>wells location</i>
Skenario Alternatif : klik tidak	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan halaman <i>user</i>
	4.

A.4 Skenarion *View Data Sumur*

Tabel 7 Skenario Use Case View data sumur admin graf

Nama Use Case	<i>View data sumur</i>
Aktor	Admin graf
Deskripsi Singkat	Admin graf akan melihat data sumur
Prekondisi	Data sumur yang telah diinputkan
Prakondisi	Menampilkan data sumur
Flow Events	
Skenario Normal : <i>View Sumur</i>	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>wells location</i>	
	2. Menampilkan halaman <i>wells location</i> meliputi peta dengan data sumur yang telah diinputkan
3. Klik salah satu marker	

	4. Menampilkan detail data sumur
--	----------------------------------

A.5 Skenario Pengelolaan Data Adjacency List

Tabel 8 skenario pengelolaan data adjacency list super admin

Nama Use Case	Pengelolaan data adjacency list
Aktor	Super admin
Deskripsi Singkat	Super admin akan mengelola data adjacency list
Prekondisi	Data adjacency list
Prakondisi	<ol style="list-style-type: none"> 1. Data sumur berhasil ditambahkan 2. Data sumur berhasil diubah 3. Data sumur berhasil dihapus
Flow Events	
Skenario Normal : View adjacency list	
Aksi Aktor	Reaksi Sistem
1. Klik menu <i>adjacency list</i>	
	2. Menampilkan halaman <i>adjacency list</i> meliputi peta, dan tabel data <i>adjacency list</i> yang telah diinputkan beserta tombol <i>edit</i> dan <i>delete</i> .
3. Klik marker	
	4. Menampilkan form input data
5. Klik add	

6. Input form adjacency	
7. Klik tombol save	
	8. Menyimpan ke database
	9. Menampilkan halaman adjacency list
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
7a. Klik tombol <i>save</i>	
	7b. Menampilkan pesan “please fill out this field”
Skenario Normal : Edit User	
Aksi Aktor	Reaksi Sistem
3. Klik tombol <i>edit</i>	
	4. Menampilkan halaman form <i>edit adjacency list</i>
5. Mengisi form <i>edit</i>	
6. Klik tombol <i>save</i>	
	7. Menyimpan ke database
	8. Menampilkan halaman <i>adjacency list</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol <i>save</i>	

	5b. Menampilkan pesan “please fill out this field”
Skenario Normal : <i>Delete data adjacency list</i>	
Aksi Aktor	Reaksi Sistem
3. Klik tombol <i>delete</i>	
	4. Menampilkan pesan “apakah anda yakin akan menghapus data ini?”
5. Klik ya	
	6. Menghapus data pada database
	7. Menampilkan halaman <i>adjacency list</i>
Skenario Alternatif : klik tidak	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan halaman <i>adjacency list</i>

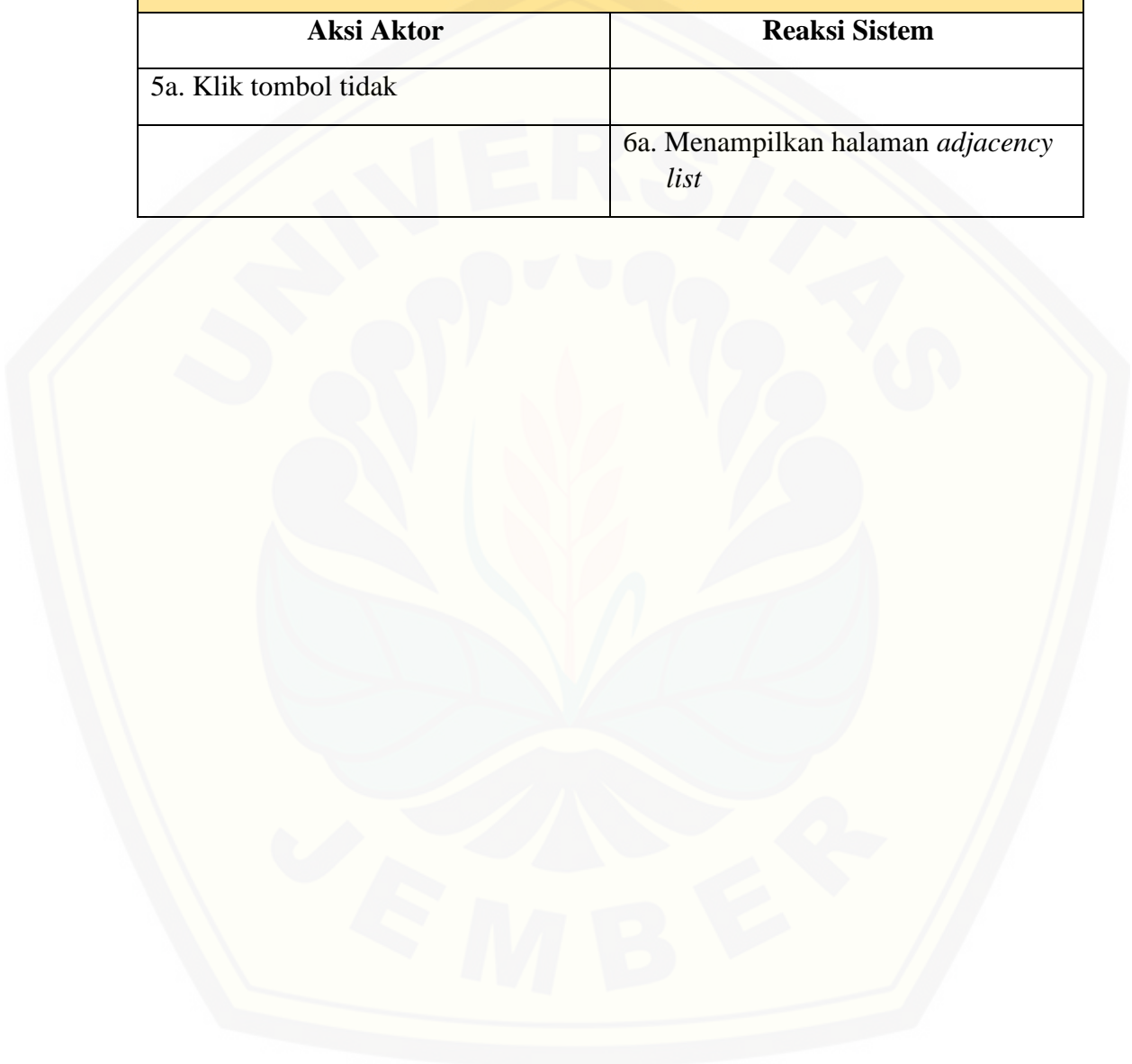
Tabel 9 skenario pengelolaan data adjacency list admin graf

Nama Use Case	Pengelolaan data adjacency list
Aktor	Admin graf
Deskripsi Singkat	admin graf akan mengelola data adjacency list
Prekondisi	Data adjacency list

Prakondisi	<ol style="list-style-type: none"> 1. Data sumur berhasil ditambahkan 2. Data sumur berhasil diubah 3. Data sumur berhasil dihapus
Flow Events	
Skenario Normal : View adjacency list	
Aksi Aktor	Reaksi Sistem
10. Klik menu <i>adjacency list</i>	
	11. Menampilkan halaman <i>adjacency list</i> meliputi peta, dan tabel data <i>adjacency list</i> yang telah diinputkan beserta tombol <i>edit</i> dan <i>delete</i> .
12. Klik marker	
13. Klik add	
14. Input form adjacency	
15. Klik tombol save	
	16. Menyimpan ke database
	17. Menampilkan halaman adjacency list
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
6a. Klik tombol <i>save</i>	
	6b. Menampilkan pesan “please fill out this field”
Skenario Normal : Edit User	

Aksi Aktor	Reaksi Sistem
9. Klik tombol <i>edit</i>	
	10. Menampilkan halaman form <i>edit adjacency list</i>
11. Mengisi form <i>edit</i>	
12. Klik tombol <i>save</i>	
	13. Menyimpan ke database
	14. Menampilkan halaman <i>adjacency list</i>
Skenario Alternatif : Form data tidak lengkap	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol <i>save</i>	
	5b. Menampilkan pesan “please fill out this field”
Skenario Normal : Delete data adjacency list	
Aksi Aktor	Reaksi Sistem
8. Klik tombol <i>delete</i>	
	9. Menampilkan pesan “apakah anda yakin akan menghapus data ini?”
10. Klik ya	
	11. Menghapus data pada database

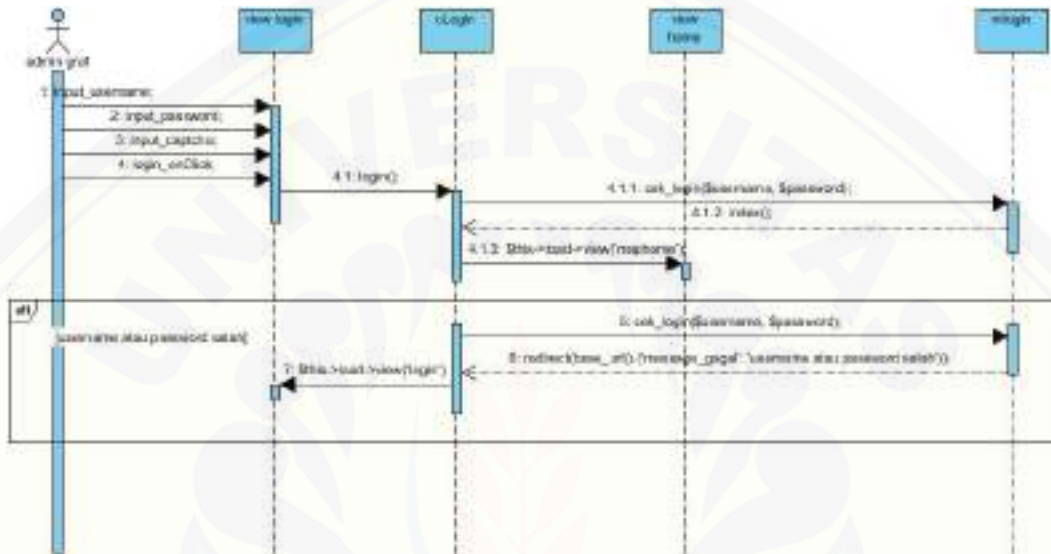
	12. Menampilkan halaman <i>adjacency list</i>
Skenario Alternatif : klik tidak	
Aksi Aktor	Reaksi Sistem
5a. Klik tombol tidak	
	6a. Menampilkan halaman <i>adjacency list</i>



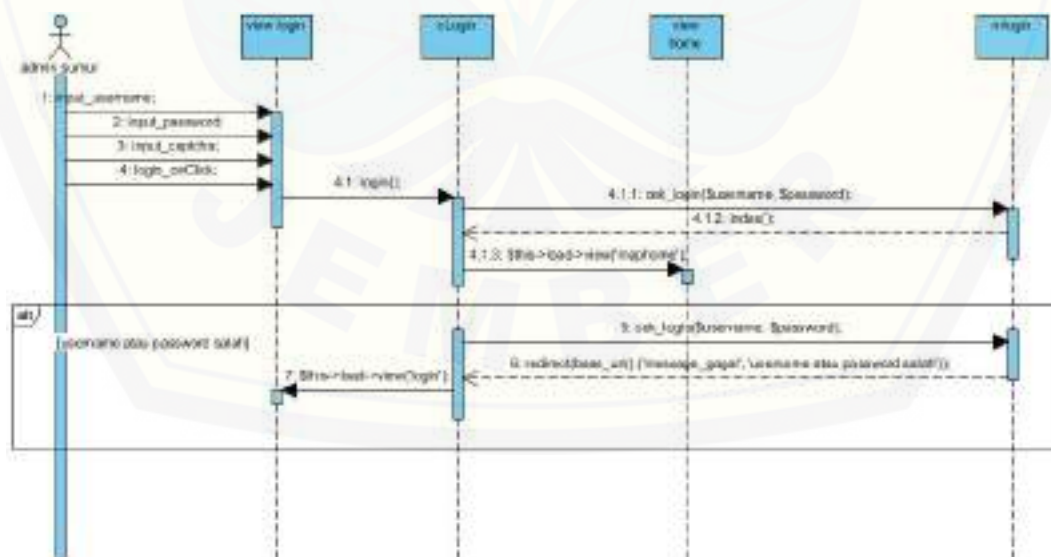
LAMPIRAN B

B.1 Sequence Diagram Login

Penggambaran *Sequence Diagram* login digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat.

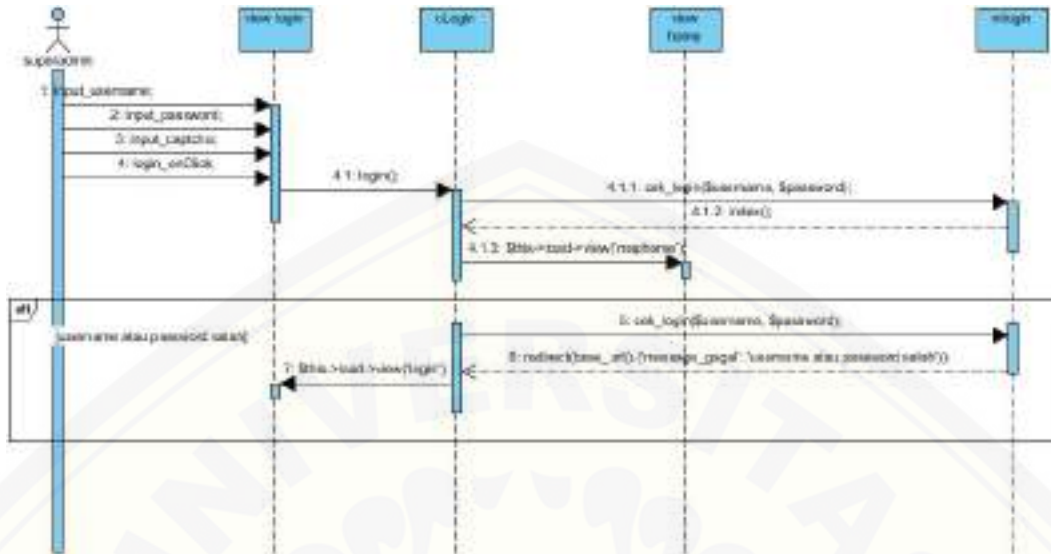


Gambar 1 Sequence Diagram Login admin graf



Gambar 2 Sequence Diagram Login admin sumur

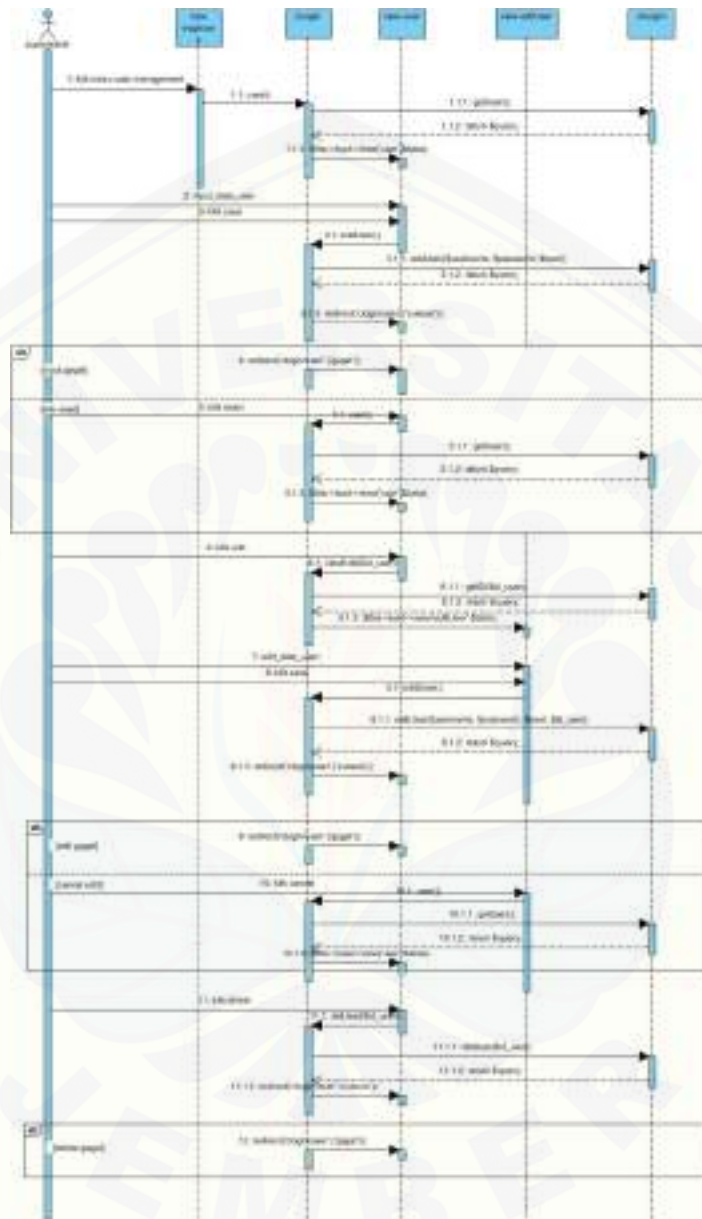




Gambar 3 Sequence Diagram Login Superadmin

B.2 Sequence Diagram Pengelolaan Data User

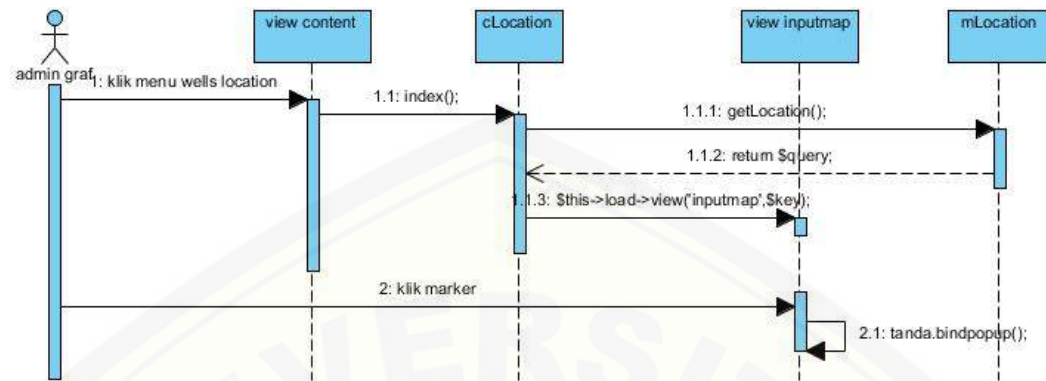
Penggambaran *Sequence Diagram* pengelolaan data *user* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat.



Gambar 4 Pengelolaan data sumur - super admin

B.3 Sequence Diagram view Data Sumur

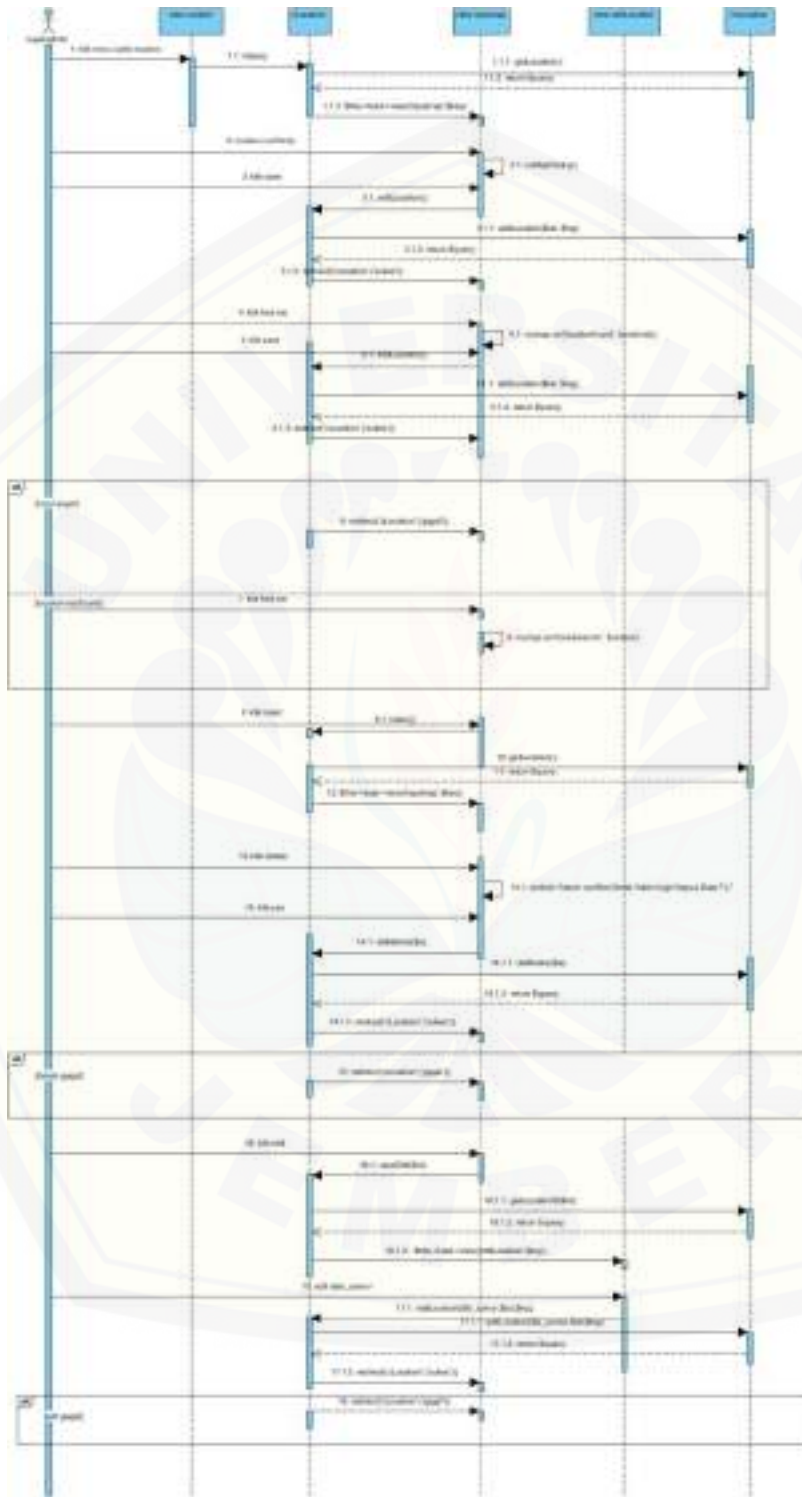
Penggambaran *Sequence Diagram* view data sumur digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat.



Gambar 5 Sequence Diagram View Data Sumur

B.4 Sequence Diagram Pengelolaan Data Sumur

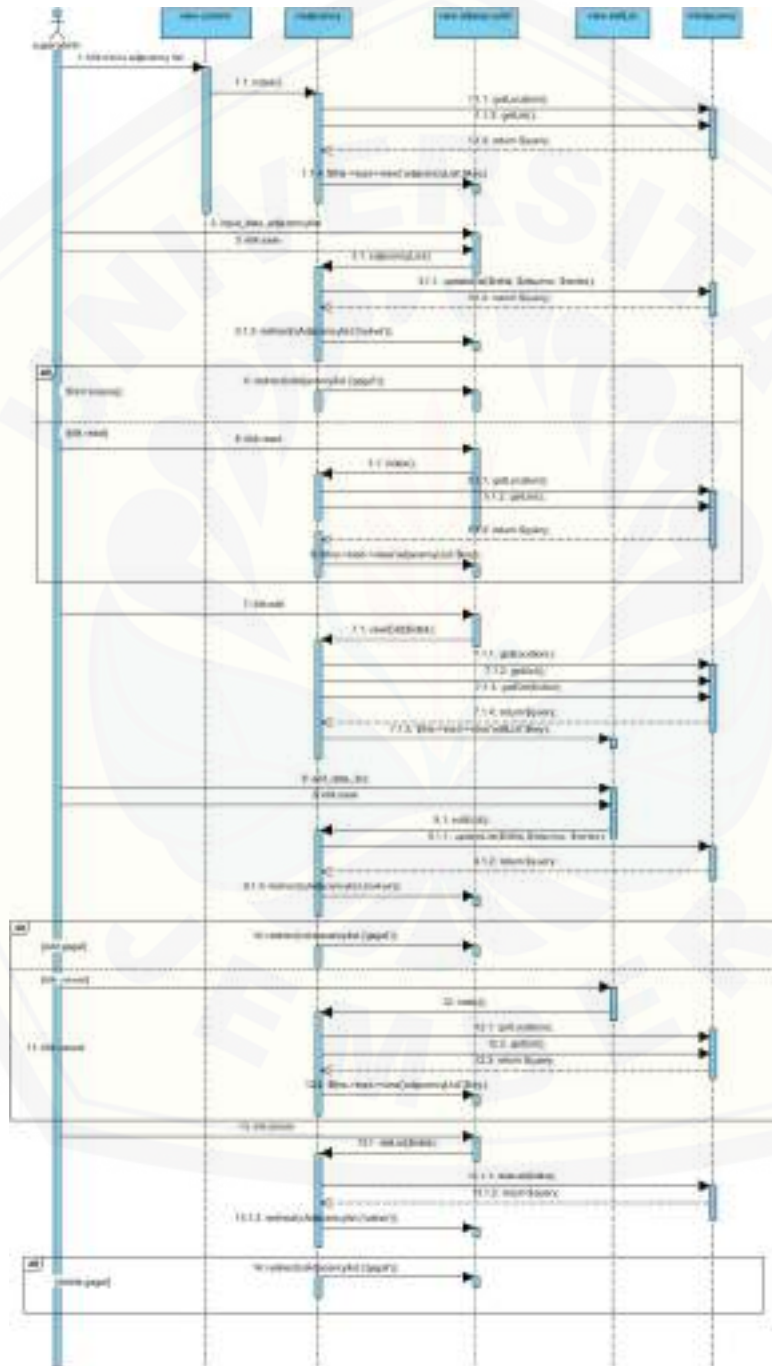
Penggambaran *Sequence Diagram* pengelolaan data sumur digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat.



Gambar 6 Pengelolaan Data Sumur

B.5 Sequence Diagram Pengelolaan Data Adjacency List

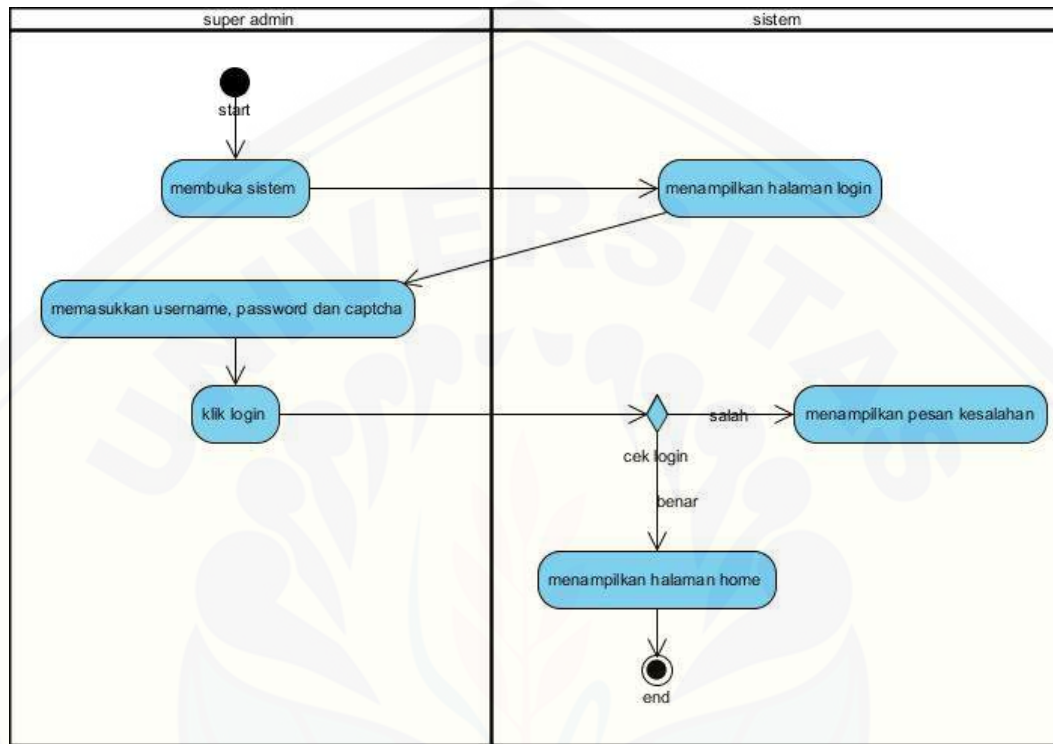
Penggambaran *Sequence Diagram* pengelolaan data *Adjacency List* digunakan untuk menjelaskan fungsi atau *method* yang akan dibuat.



Gambar 7 Pengelolaan Data Adjacency List

LAMPIRAN C

C.1 Activity Diagram Login



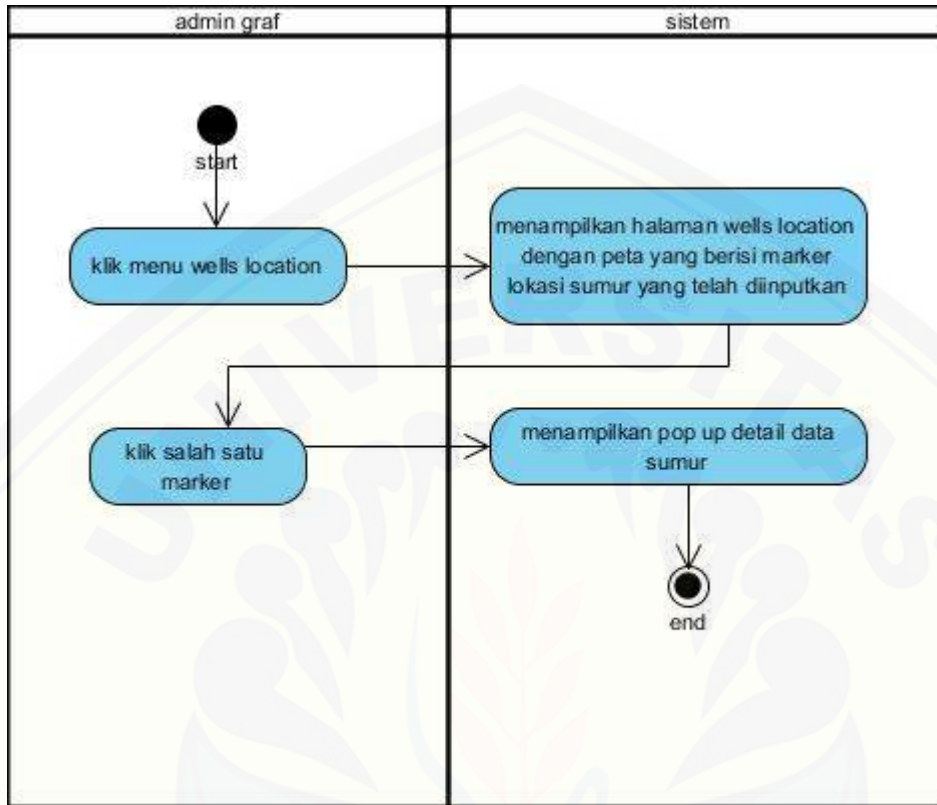
Gambar 8 Activity Diagram Login

C.2 Activity Diagram Pengelolaan Data User



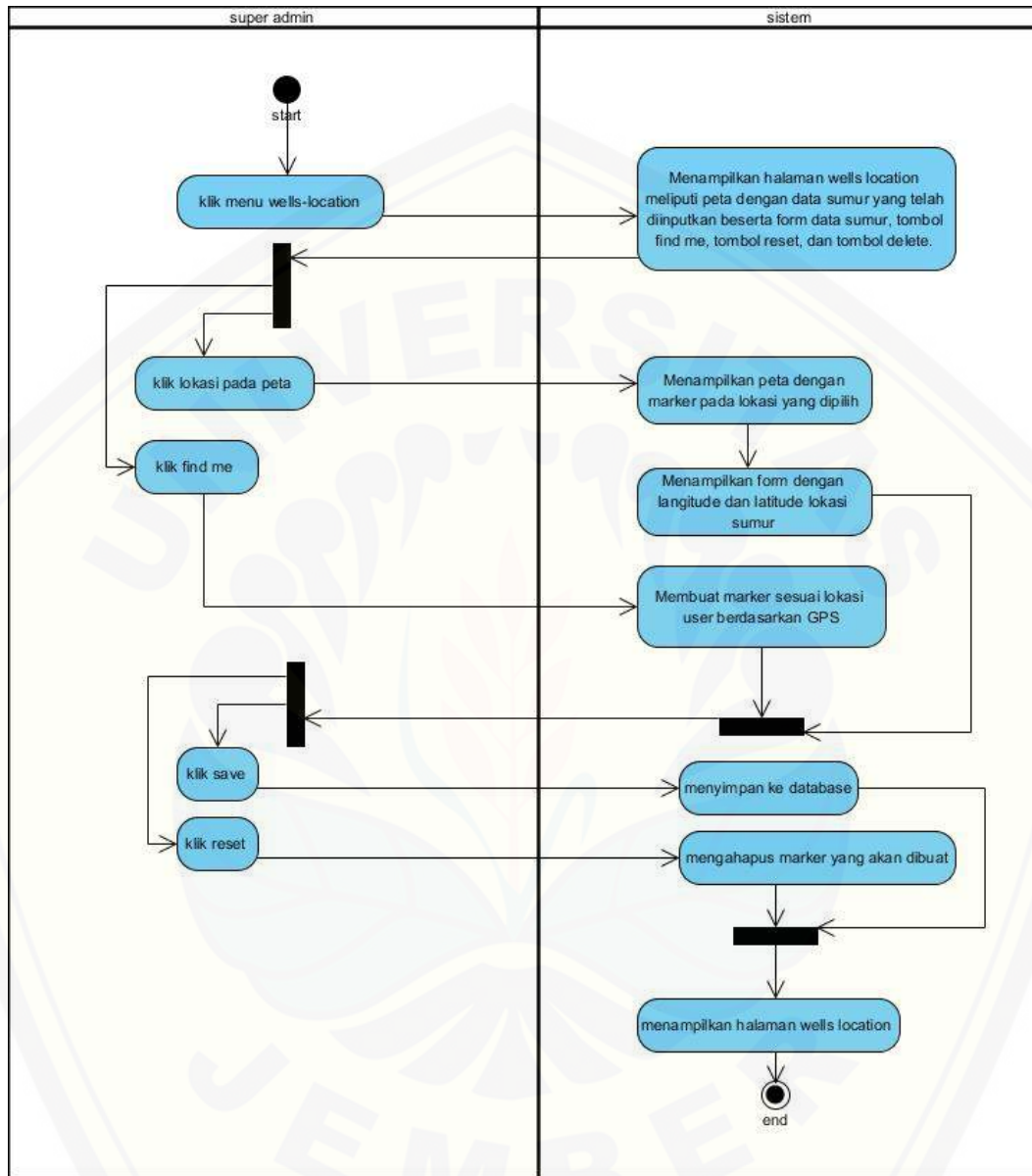
Gambar 9 Activity Diagram Pengelolaan Data User

C.3 Activity Diagram View Data Sumur



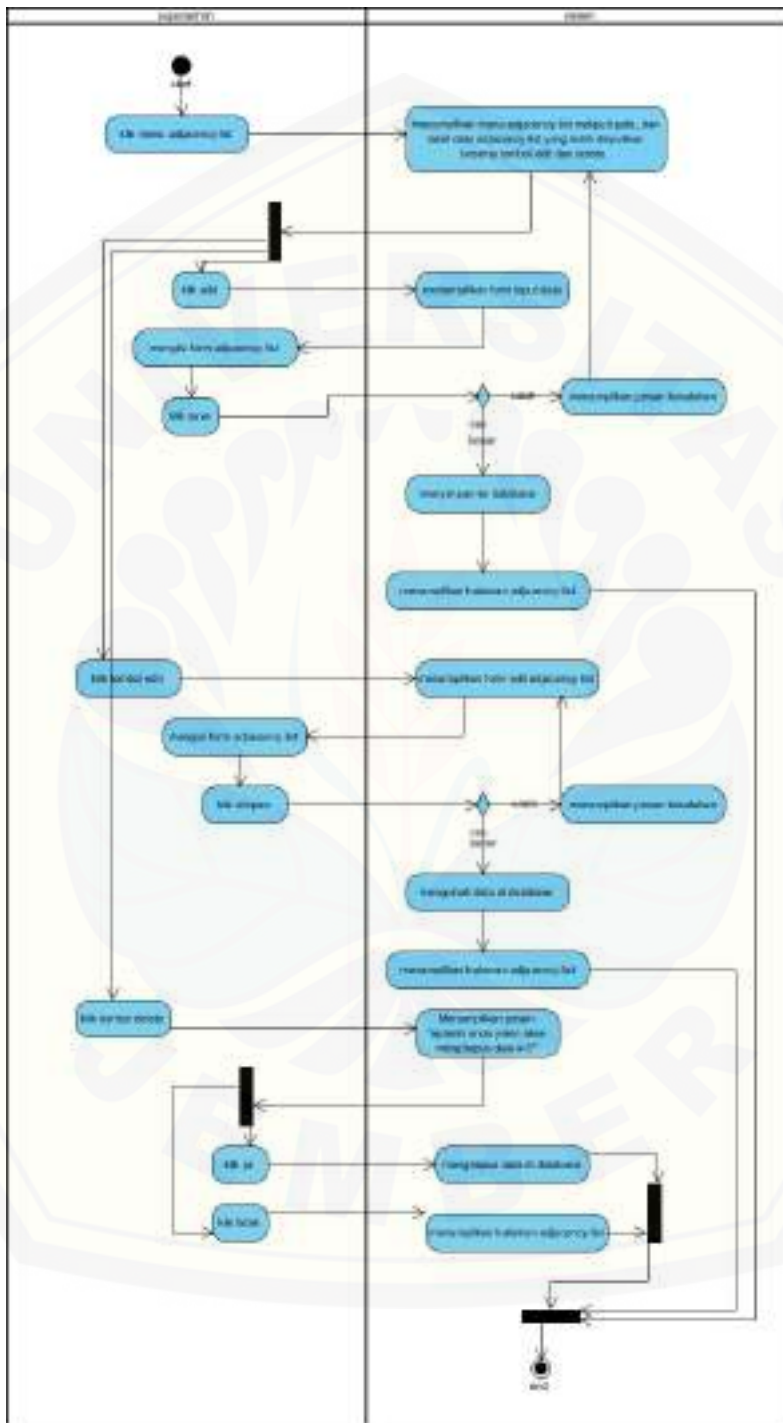
Gambar 10 Activity Diagram View Data Sumur

C.4 Activity Diagram Pengelolaan Data Sumur



Gambar 11 Activity Diagram Pengelolaan Data Sumur

C.5 Activity Diagram Pengelolaan Data Adjacency List



Gambar 12 Activity Diagram Pengelolaan Data Adjacency List

LAMPIRAN D

D.1 Kode Program *Login*

Kode program login terletak pada kelas *view* login, controller clogin, dan model mlogin.

```

public function index() {
if ($this->session->userdata('level') == 'superadmin') {
$this->load->view('sidebar');
$this->load->view('content');
$this->load->view('footer');
} else if ($this->session->userdata('level') == 'admin sumur') {
$this->load->view('sidebar');
$this->load->view('content');
$this->load->view('footer');
} else if ($this->session->userdata('level') == 'admin graf') {
$this->load->view('sidebar');
$this->load->view('content');
$this->load->view('footer');
} else {
$data['cap'] = $this->captcha(320);
$this->load->view('login',$data);
}}
public function login() {
$username = $this->input->post('username');
$password = $this->input->post('password');
$captcha = $this->input->post('captcha-code');
$type = $this->input->post('captcha-input');
$filename = $this->input->post('captcha-filename');
$path = 'captcha/'.$filename;
if($type === $captcha) {
$result = $this->mlogin->cek_login($username, $password);
if($result) {
$this->deletefile($path);
$this->session->set_flashdata('sukses');
redirect('clogin/index');
} else{
redirect(base_url());
$this->session->set_flashdata('gagal');
}
} else{
redirect(base_url());
$this->session->set_flashdata('gagal');
}}
public function Captcha($width) {
if(empty($width)) {
$width = 450;
}
$prevCap = $this->session->userdata('captcha');
if(!empty($prevCap)){
if(file_exists('captcha/'.$prevCap)){
$this->deleteFile('captcha/'.$prevCap);
}
}
$this->session->unset_userdata('captcha');
}

```

```

}
$this->load->helper('captcha');
$vals = array(
'word' => rand(1001,9999),
'img_path' => './captcha/',
'img_url' => base_url().'captcha/',
'img_width' => $width,
'img_height' => 40,
'expiration' => 7200,
'font_size' => 50,
'colors' => array(
'background' => array(255, 255, 255),
'border' => array(255, 255, 255),
'text' => array(0, 0, 0),
'grid' => array(255, 140, 40)
);
$cap = create_captcha($vals);
$this->session->set_userdata('captcha', $cap['filename']);
return $cap;
}
public function deleteFile($path){
unlink($path);
}
public function logout(){
$this->session->sess_destroy();
redirect(base_url());
}
}

```

Gambar 13 Kode Program 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 sumur') {
                $this->session->set_userdata('admin sumur', TRUE);
            }
            return TRUE;
        }
    } else {
        return FALSE;
    }
}
}

```

Gambar 14 Kode Program mlogin

D.2 Kode Program Pengelolaan Data User

Kode program pengelolaan data user terletak pada kelas *view* user, *editUser*, *controller* *clogin*, dan *model* *mlogin*.

```

public function user(){
    $this->load->view('sidebar');
    $data['user'] = $this->mlogin->getUser();
    $this->load->view('user',$data);
    $this->load->view('footer');
}
public function viewEdit($id_user){
    $this->load->view('sidebar');
    $data['user'] = $this->mlogin->getUser();
    $data['userID'] = $this->mlogin->getID($id_user);
    $this->load->view('editUser',$data);
    $this->load->view('footer');
}
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 15 Kode Program pengelolaan Data User controller *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 16 Kode Program pengelolaan Data User moedel mlogin

D.3 Kode Program View Data Sumur

Kode program view data sumur terletak pada kelas view inputmap, controller cLocation dan model mLocation.

```

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

```

Gambar 17 Kode Program view data sumur controller cLocation

```

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

```

Gambar 18 Kode Program view data sumur model mLocation

D.4 Kode Program Pengelolaan Data Sumur

Kode program pengelolaan data sumur terletak pada kelas view inputmap, editSumur, controller cLocation, dan model mLocation.

```
<?php
```

```

if(!defined('BASEPATH')) exit ('No direct script allowed');
class CLocation extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->model('mLocation');
    }
    public function index() {
        $this->load->view('sidebar');
        $key['lokasi'] = $this->mLocation->getLocation();
        $this->load->view('inputmap',$key);
        $this->load->view('footer');
    }
    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('sidebar');
        $key['detail'] = $this->mLocation->getLocationID($id);
        $this->load->view('editLocation',$key);
        $this->load->view('footer');
    }
    public function editLocation(){
        $id_sumur = $this->input->post('id_sumur');
        $lat = $this->input->post('lat');
        $lng = $this->input->post('lng');
        $input = $this->mLocation->editLocation($id_sumur, $lat, $lng);
        if($input==true){
            $this->session->set_flashdata('sukses', 'edit success');
            redirect('cLocation');
        } else {
            $this->session->set_flashdata('gagal', 'edit failed');
            redirect('cLocation');
        }
    }
    public function delMarker($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 19 Kode Program Pengelolaan Data Sumur controller cLocation

```

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

```

Gambar 20 Kode Program Pengelolaan Data Sumur model mLocation

D.5 Kode Program Pengelolaan Data *Adjacency List*

Kode program pengelolaan data sumur terletak pada kelas *view adjacencyList*, *editList*, *controller cAdjacency*, dan *model mAdjacency*.

```

<?php

if(!defined('BASEPATH')) exit ('No direct script allowed');

class CAdjacency extends CI_Controller{

    public function __construct(){
        parent::__construct();

        // Membuka Model
        $this->load->model('mAdjacency');
        $this->load->model('mLocation');
    }

    public function index() {
        $this->load->view('sidebar');
        $key['lokasi'] = $this->mLocation->getLocation();
        $key['list'] = $this->mAdjacency->getList();
        $key['latlng'] = $this->mAdjacency->getLatLng();
        $this->load->view('adjacencyList',$key);
        $this->load->view('footer');
    }
    public function detailList($id_sumur) {
        $this->load->view('sidebar');
        $key['lokasi'] = $this->mLocation->getLocation();
        $key['list'] = $this->mAdjacency->detailList($id_sumur);
        $this->load->view('detailList',$key);
        $this->load->view('footer');
    }
    public function viewEdit($idlist) {
        $this->load->view('sidebar');
        $key['lokasi'] = $this->mLocation->getLocation();
        $key['list'] = $this->mAdjacency->getList();
        $key['idlist'] = $this->mAdjacency->getEdit($idlist);
        $this->load->view('editList',$key);
        $this->load->view('footer');
    }
    public function adjacencyList()
    {
        $id = $this->input->post('id');
        $id_sumur = $this->input->post('id_sumur');
        $input = $this->mAdjacency->adjacencyList($id, $vertex);
        if($input==true){
            $this->session->set_flashdata('sukses', 'add success');
            redirect('cAdjacency');
        } else {
            $this->session->set_flashdata('gagal', 'add failed');
            redirect('cAdjacency');
        }
    }
    public function editList() {
        $idlist = $this->input->post('idlist');
        $id_sumur = $this->input->post('id_sumur');
    }
}

```



```

$list = $this->input->post('list');
$lat = $this->input->post('lat');
$lng = $this->input->post('lng');
$input = $this->mAdjacency->updateList($idlist, $id_sumur, $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 21 Kode Program pengelolaan data adjacency list controller cAdjacency

```

<?php
if(!defined('BASEPATH')) exit ('No direct script access allowed');
class MAdjacency extends CI_Model{
    public function adjacencyList($id, $vertex) {
        $query = $this->db->query("INSERT into tb_list (id_sumur, list) values ('".$id."','".$vertex."");");
        return $query;
    }
    public function getList() {
        $query = $this->db->query("select * from tb_list");
        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, $idsumur, $vertex) {
        $query = $this->db->query("UPDATE `tb_list` SET `id_list`='".$idlist."', `id_sumur`='".$idsumur."', `list`='".$vertex.'" 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;
        }
    }
}
}
?>

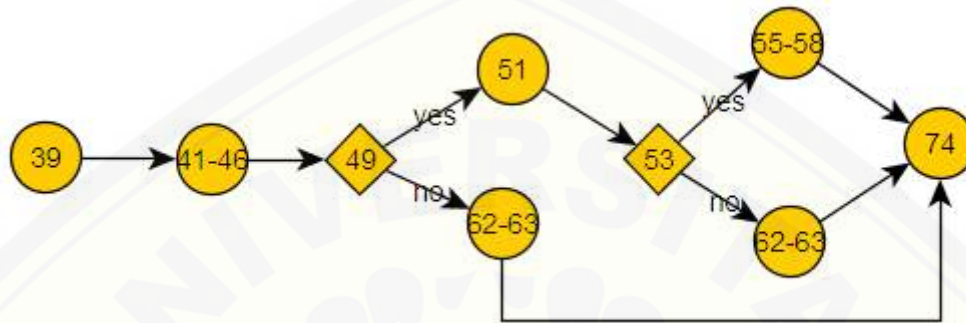
```

Gambar 22 Kode Program pengelolaan data adjacency list model mAdjacency

LAMPIRAN E

E.1 Kelas *Controller*

1. *Class* clogin



Gambar 23 Diagram Alir function login()

$$CC = \text{EDGE} - \text{NODE} + 2$$

$$CC = 10 - 9 + 2$$

$$CC = 3$$

Jalur 1 = 39-41-42-43-44-45-46-49-51-53-55-56-57-58-74 (jika login sukses)

Jalur 2 = 39-41-42-43-44-45-46-49-62-63-74 (jika captcha salah)

Jalur 3 = 39-41-42-43-44-45-46-49-51-53-62-63-74 (jika login gagal)

Tabel 10 Test case function login()

Jalur 1	
<i>Test Case</i>	Jika berhasil melakukan login
Target yang Diharapkan	Menampilkan Halaman Dashboard
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	39-41-42-43-44-45-46-49-51-53-55-56-57-58-74
Jalur 2	
<i>Test Case</i>	Jika gagal memasukkan captcha
Target yang Diharapkan	Menampilkan Halaman login dan pesan kesalahan
Hasil Pengujian	Benar

<i>Path/ Jalur</i>	39-41-42-43-44-45-46-49-62-63-74
Jalur 2	
<i>Test Case</i>	Jika gagal melakukan login
Target yang Diharapkan	Menampilkan Halaman login dan pesan kesalahan
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	39-41-42-43-44-45-46-49-51-53-62-63-74



Gambar 24 Diagram Alir function index()

2. *Class cLocation*

$$CC = \text{EDGE-NODE}+2$$

$$CC = 2-3+2$$

$$CC = 1$$

$$\text{Jalur 1} = 18-19-20-21-22-23$$

Tabel 11 Test case function index()

Jalur 1	
<i>Test Case</i>	Jika berhasil menampilkan halaman <i>wells location</i>
Target yang Diharapkan	Menampilkan Halaman <i>wells location</i>
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	18-19-20-21-22-23

CC = EDGE-NODE+2

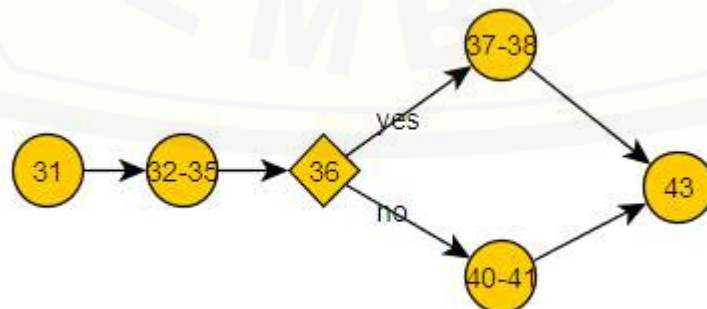
CC = 6-6+2

CC = 2

Jalur 1 = 31-32-33-34-35-36-37-38-43

Jalur 2 = 31-32-33-34-35-36-40-41-43

Jalur 1	
<i>Test Case</i>	Jika berhasil menginputkan data sumur
Target yang Diharapkan	Menampilkan Halaman <i>wells location</i> dan pesan sukses
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	31-32-33-34-35-36-37-38-43
Jalur 2	
<i>Test Case</i>	Jika gagal menginputkan data sumur
Target yang Diharapkan	Menampilkan Halaman <i>wells location</i> dan pesan gagal
Hasil Pengujian	Benar
<i>Path/ Jalur</i>	31-32-33-34-35-36-40-41-43



Gambar 25 Diagram Alir function addLocation()