

**SISTEM PENDUKUNG KEPUTUSAN MENENTUKAN JENIS
VEGETASI MANGROVE MENGGUNAKAN ELIMINATION
AND CHOICE EXPRESSING REALITY (ELECTRE) PADA
DINAS LINGKUNGAN HIDUP
KOTA MAKASSAR**

SKRIPSI



Oleh :

**ERWIN RIAN TO
2013020053**

JURUSAN TEKNIK INFORMATIKA

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER (STMIK) HANDAYANI
MAKASSAR
2017**

SISTEM PENDUKUNG KEPUTUSAN MENENTUKAN JENIS
VEGETASI *MANGROVE* MENGGUNAKAN *ELIMINATION AND
CHOICE EXPRESSING REALITY (ELECTRE)* PADA
DINAS LINGKUNGAN HIDUP
KOTA MAKASSAR

SKRIPSI

Diajukan sebagai salah satu syarat untuk mencapai gelar Sarjana Komputer
Jurusan Teknik Informatika

Disusun dan diajukan oleh :

ERWIN RIAN TO
2013020053

JURUSAN TEKNIK INFORMATIKA

SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN
KOMPUTER (STMIK) HANDAYANI
MAKASSAR
2017

HALAMAN PERSETUJUAN UJIAN SKRIPSI

Judul : SISTEM PENDUKUNG KEPUTUSAN MENENTUKAN
JENIS VEGETASI *MANGROVE* MENGGUNAKAN
ELIMINATION AND CHOICE EXPRESSING REALITY
(ELECTRE) PADA DINAS LINGKUNGAN HIDUP KOTA
MAKASSAR

ERWIN RIAN TO
2013020053
TEKNIK INFORMATIKA

Telah diperiksa dan disetujui untuk diajukan sebagai salah satu syarat untuk
menempuh Ujian Skripsi guna memperoleh gelar Sarjana Komputer.

Makassar, 8 Agustus 2017

Pembimbing 1

Pembimbing 2

Billy Eden William Asrul, S.Kom.,MT Respaty Namruddin, S.Kom., MT

Mengetahui,

Ketua Jurusan Teknik Informatika

Billy Eden William Asrul, S.Kom.,MT

HALAMAN PENGESAHAN

SISTEM PENDUKUNG KEPUTUSAN MENENTUKAN JENIS VEGETASI
MANGROVE MENGGUNAKAN *ELIMINATION AND CHOICE EXPRESSING*
REALITY (ELECTRE) PADA DINAS LINGKUNGAN HIDUP KOTA
MAKASSAR

Oleh

Nama : Erwin Rianto
NPM : 2013020053

Telah dipertahankan di hadapan sidang penguji Skripsi STMIK Handayani
pada tanggal 24 Agustus 2017
dan dinyatakan telah memenuhi syarat memperoleh gelar Sarjana Komputer.

Tim Penguji :

Tanda Tangan

- | | |
|---|---------|
| 1. Billy Eden William Asrul, S.Kom.,MT
Ketua sidang | 1. |
| 2. Respaty Namruddin, S.Kom., MT
Sekretaris Sidang | 2. |
| 3. Dr. Nasrullah M.Si
Anggota | 3. |
| 4. Dra.Najirah Umar, S.Kom.,MT
Anggota | 4. |
| 5. Herlinah, S.Kom., M.Si
Anggot | 5. |

Telah diperiksa dan disetujui untuk diajukan sebagai salah satu syarat untuk
menempuh Ujian Skripsi guna memperoleh gelar Sarjana Komputer.

Mengesahkan,

Pembantu Ketua I

Ketua Jurusan

Dr. Nasrullah M.Si

Billy Eden William Asrul, S.Kom.,MT

Mengetahui,

Ketua STMIK Handyani Makassar

Dr. Eng. Agussalim., MT

Sistem Pendukung Keputusan Menentukan Jenis Vegetasi *Mangrove* menggunakan *Elimination And Choice Expressing Reality* (Electre) pada Dinas Lingkungan Hidup Kota Makassar

Erwin Rianto, 2017

Dibimbing oleh : Billy Eden William Asrul dan Respaty Namruddin

ABSTRAK

Hutan *Mangrove* merupakan ekosistem peralihan yang memiliki keistimewaan, yakni *Mangrove* merupakan ekosistem yang sangat produktif dan mampu mendukung kelangsungan hidup dari ekosistem lain. Penelitian ini bertujuan untuk merancang dan mengimplementasikan Sistem Pendukung Keputusan menganalisa Kesesuaian Jenis Vegetasi *Mangrove* menggunakan Electre yang sesuai berdasarkan inputan faktor-faktor lingkungan.

Desain penelitian yang digunakan adalah UML yang didesain secara terstruktur yang terdiri dari rancangan model *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*. *Software* yang digunakan dalam membangun sistem ini adalah Delphi 7 dan MySQL untuk pengolahan database. Algoritma yang digunakan adalah *Elimination And Choice Expressing Reality* (Electre) dengan konsep *outranking* menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai. Dalam penelitian ini pengumpulan data diperoleh melalui *observasi*, wawancara dan kepustakaan.

Hasil dari penelitian ini adalah sistem mampu menentukan jenis *Mangrove* yang paling tepat untuk ditanam pada lahan dengan criteria yang dihitung, yakni salinitas, kekuatan ombak dan angin, kandungan pasir, lumpur, frekuensi penggenangan dan kepadatan penduduk. Berdasarkan pengujian *whitebox*, maka didapatkan nilai total Kompleksitas Siklomatis sebesar 21, *Independent Path* sebesar 21 dan *Graph* Matrik sebesar 21.

Kata kunci : *Elimination And Choice Expressing Reality*, *Mangrove* , *SPK*

KATA PENGANTAR

Assalamu Alaikum Wr.Wb.

Sesungguhnya segala puji bagi Allah, kami memuji-Nya, memohon pertolongan kepada-Nya, memohon ampunan kepada-Nya, bertaubat Kepada-Nya dan kami berlindung kepada Allah dari kejahatan diri kami dan keburukan perbuatan kami. Berkat rahmat dan karunia-Nya penulis dapat menyelesaikan penyusunan skripsi ini. Shalawat beserta salam semoga senantiasa terlimpah curahkan kepada Nabi Muhammad Shallallahu ‘alaihi wasallam, kepada keluarganya, para sahabatnya, hingga kepada ummatnya sampai akhir zaman.

Skripsi ini diajukan untuk memenuhi salah satu syarat memperoleh gelar sarjana (S1) pada STMIK Handayani Makassar dan juga merupakan laporan akhir yang wajib diselesaikan oleh setiap mahasiswa STMIK Handayani Makassar. Semoga dapat memberikan nilai tambah bagi semua pihak yang terkait didalamnya dan khususnya kepada penulis.

Dalam penyusunan dan penulisan skripsi ini tidak terlepas dari bantuan berupa sumbangan pemikiran, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu dalam kesempatan ini penulis mengucapkan terima kasih yang setulus-tulusnya kepada yang terhormat :

1. Kedua orang tua yang saya cintai, keluarga yang senantiasa memberikan nasihat, dukungan, sehingga penulis dapat menyelesaikan studi dengan baik.
2. Dr. Eng. Agussalim, MT selaku ketua STMIK Handayani Makassar.

3. Billy Eden W. Asrul, S.Kom, MT selaku ketua jurusan Teknik Informatika sekaligus Pembimbing I dalam penyusunan skripsi ini, sehingga dapat selesai dengan baik.
4. Respaty Namruddin, S.Kom., MT selaku pembimbing II dalam penyusunan skripsi ini, sehingga dapat selesai dengan baik.
5. Pimpinan dan Staf Dinas Lingkungan Hidup, khususnya Ervan Agustiar, ST selaku Kepala Seksi Pengendalian dan Kemitraan Ruang Terbuka Hijau.
6. Alumni STMIK Handayani Makassar yang telah memberikan arahan dan sumbangan pemikiran.
7. Kepada segenap pihak yang tidak sempat penulis sebutkan satu persatu dalam kesempatan singkat ini.

Penulis menyadari bahwa penulisan skripsi ini masih banyak kekurangan. Olehnya itu, kritik dan saran yang sifatnya mendidik, dan dukungan yang membangun, senantiasa penulis harapkan.

Saya berharap kepada Allah agar menjadikan kita termasuk orang-orang yang membimbing lagi dibimbing, menjaga batasan-batasanNya, dan meneladani sunnah Rasul-Nya. Akhir kata semoga skripsi ini dapat bermanfaat bagi pembaca, dan memberikan sumbangan ilmiah kepada Almamater dan masyarakat. Semoga Allah Subhanahu Wata'ala memberikan rahmat dan karunia-Nya bagi kita semua. Amin.

Wassalamu Alaikum Warahmatullahi Wabarakatu.

Makassar, 12 Juli 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGANTAR.....	ii
HALAMAN PENGESAHAN	ii
ABSTRAK	v
KATA PENGANTAR.....	vi
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB I PENDAHULUAN	1
A. Latar Belakang	1
B. Batasan Masalah.....	3
C. Rumusan Masalah	3
D. Tujuan Penelitian	4
E. Manfaat Penelitian	4
F. Penegasan Konsep	4
BAB II TINJAUAN PUSTAKA	7
A. Teori Sistem Pendukung Keputusan	7
B. Teori <i>Mangrove</i>	11
C. <i>Elimination and Choice Expressing Reality</i> (Electre)	23
D. <i>Unified Modelling Language</i> (UML)	27
E. Kamus Data	38

F. Konsep Database	41
G. MySQL	51
H. Software yang Digunakan	52
I. Metode Pengujian.....	59
BAB III METODE PENELITIAN.....	67
A. Penggambaran Sistem	67
B. Analisis Kebutuhan	69
C. Sumber Data dan Teknik Pengumpulan Data	71
D. Waktu dan Lokasi Penelitian	72
D. Gambaran Umum Lokasi Penelitian	72
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM.....	75
A. Pemodelan Sistem	75
B. Perancangan Database.....	80
C. Perancangan Antar Muka.....	82
D. Implementasi	85
E. Pengujian Sistem.....	99
BAB V.....	119
KESIMPULAN	119
A. Kesimpulan	119
B. Saran	119
DAFTAR PUSTAKA	120
LAMPIRAN	122

DAFTAR TABEL

Tabel 2.1 Kesesuaian Jenis Vegetas <i>Mangrove</i>	14
Tabel 2.2 Simbol <i>Use Case</i>	30
Tabel 2.3 Simbol <i>Activity Diagram</i>	33
Tabel 2.4 Simbol <i>Sequence Diagram</i>	34
Tabel 2. 5 Simbol <i>Class Diagram</i>	36
Tabel 4.1 tbanalisa.....	80
Tabel 4.2 tbdatlingkungan.....	80
Tabel 4.3 tbkec	81
Tabel 4.4 tbranking	81
Tabel 4.5 tbvegetasi.....	81
Tabel 4.6 login.....	82
Tabel 4.7 Contoh data lingkungan	85
Tabel 4.8 Konversi data lingkungan	85

DAFTAR GAMBAR

Gambar 2.1 Sistem pemrosesan bekas	42
Gambar 2.2 <i>flowgraph</i>	62
Gambar 2.3 <i>flowchart</i> dan <i>flowgraph</i>	63
Gambar 2.4 <i>Translating</i>	64
Gambar 3.1 <i>Usecase diagram</i> sistem yang berjalan	67
Gambar 3.2 <i>Usecase diagram</i> sistem yang di usulkan.....	68
Gambar 4.1 <i>Use Case Diagram</i>	75
Gambar 4.2 <i>Class Diagram</i>	77
Gambar 4.3 <i>Activity Diagram</i>	78
Gambar 4.4 <i>Sequence Diagram</i>	79
Gambar 4.5 Rancangan <i>Ouput</i>	82
Gambar 4.6 Rancangan <i>Input</i> Login	83
Gambar 4.7 Rancangan <i>Input</i> Data Kecematan	83
Gambar 4.8 Rancangan <i>Input</i> Data Vegetasi <i>Mangrove</i>	83
Gambar 4.9 Rancangan <i>Input</i> Data Analisa Lingkungan.....	84
Gambar 4.10 Rancangan <i>Inpu</i> Analisa Kesesuaian	84
Gambar 4.11 Rancangan <i>Input</i> Pengguna.....	84
Gambar 4.12 Tampilan Login	94
Gambar 4.13 Menu Utama Admin	94
Gambar 4.14 Menu Data Kecematan	95
Gambar 4.15 Menu Data Vegetasi <i>Mangrove</i>	95

Gambar 4.16 Menu Data Analisa Lingkungan.....	96
Gambar 4.17 Menu Data Analisa Kesesuaian Jenis Lingkungan	97
Gambar 4.18 Menu Data Pengguna	97
Gambar 4.19 Laporan Hasil Analisa	98

BAB I

PENDAHULUAN

A. Latar Belakang

Hutan *Mangrove* merupakan ekosistem peralihan yang memiliki keistimewaan, yakni *Mangrove* merupakan ekosistem yang sangat produktif dan mampu mendukung kelangsungan hidup dari ekosistem lain. Hal tersebut terjadi karena *Mangrove* merupakan pelindung dan sekaligus sumber *nutrien* bagi organisme yang hidup di tengahnya. Selain hal tersebut, keistimewaan lain yang dimiliki *Mangrove* adalah kemampuan untuk mengikat polutan berupa logam berat maupun senyawa racun yang lain.

Ekosistem *Mangrove* merupakan salah satu ekosistem yang harus dijaga kelestariannya. Akan tetapi, pada masa sekarang ini banyak hutan *Mangrove* telah mengalami kerusakan yang disebabkan oleh faktor Manusia dan Alam. Strategi yang dapat digunakan untuk mengatasi hal tersebut adalah dengan melakukan *reboisasi* atau penanaman kembali bagi hutan *Mangrove* yang mengalami kerusakan.

Namun *reboisasi* tanpa persiapan dan perencanaan yang matang dapat mengakibatkan gagalnya usaha *reboisasi*, yakni tanaman akan tumbuh kerdil, rendah dan batang seringkali bengkok, sehingga *Mangrove* yang ditanam tetap menyebabkan abrasi pantai dan lain-lain. Salah satu upaya untuk meminimalisasi kegagalan tersebut yakni dengan pengelolaan secara

terencana dengan tahap perhitungan tingkat salinitas, penyesuaian terhadap kekuatan ombak, angin, lumpur, kandungan pasir, frekuensi penggenangan dan kepadatan penduduk. Dinas Lingkungan Hidup Kota Makassar yang terletak di jalan Urip Sumoharjo No. 8 kompleks gabungan dinas Makassar merupakan instansi yang menangani pertumbuhan ekosistem *Mangrove* di wilayah kota Makassar.

Di Makassar sendiri, *Mangrove* hanya menyisahkan kurang lebih 30% hutan bakau yang sehat, dimana telah terdapat sekitar 70% hutan bakau yang rusak dari total lahan seluas 26.000 hektar dengan 10 jenis vegetasi.

Mengacu pada hal diatas, maka penulis mengangkat judul '**Sistem Pendukung Keputusan Menentukan Jenis Vegetasi *Mangrove* menggunakan Algoritma *Elimination And Choice Expressing Reality (Electre)***' yang diharapkan dapat membantu dalam menentukan jenis vegetasi *Mangrove* yang sesuai berdasarkan inputan faktor-faktor lingkungan.

Peneliti menggunakan metode Electre untuk menyelesaikan permasalahan pengambilan keputusan memalui perankingan berdasarkan kelebihan dan kekurangan melalui perbandingan berpasangan pada kriteria yang sama. Electre merupakan salah satu metode pengambilan keputusan multikriteria berdasarkan pada konsep outranking dengan menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai.

B. Batasan Masalah

A. Untuk mempertajam sasaran dan membatasi ruang lingkup, maka perlu membatasi masalah tentang hal yang akan dirancang. Penulis membatasi masalah sebagai berikut:

1. Sistem yang penulis Rancang adalah *Sistem Pendukung Keputusan Menentukan Jenis Vegetasi Mangrove menggunakan Algoritma Elimination And Choice Expressing Reality (Electre)*.
2. Kriteria yang diolah yaitu salinitas, kekuatan ombak dan angin, kandungan pasir, lumpur, frekuensi penggenangan dan kepadatan penduduk.
3. Hanya mencakup jenis vegetasi *Mangrove* yang ada di wilayah makassar.
4. Pengujian sistem menggunakan pengujian *white box testing*.
5. Pengujian Sistem hanya ke metode yang digunakan yaitu *Elimination And Choice Expressing Reality (Electre)*.

C. Rumusan Masalah

1. Bagaimana merancang Sistem Pendukung Keputusan untuk menganalisa kesesuaian jenis vegetasi *Mangrove* menggunakan Electre ?
2. Bagaimana mengimplementasikan Sistem Pendukung Keputusan untuk menganalisa kesesuaian jenis vegetasi *Mangrove* menggunakan Electre ?

D. Tujuan Penelitian

B.Adapun tujuan dari penelitian ini adalah :

1. Merancang Sistem Pendukung Keputusan menganalisa Kesesuaian Jenis Vegetasi *Mangrove* menggunakan Electre.
2. Mengimplementasikan Sistem Pendukung Keputusan menganalisa Kesesuaian Jenis Vegetasi *Mangrove* menggunakan Electre.

E. Manfaat Penelitian

Manfaat yang akan dicapai dalam penelitian ini adalah sebagai berikut:

1. Untuk mempermudah proses reboisasi hutan *Mangrove*.
2. Memudahkan dalam pemilihan jenis *Mangrove* yang akan ditanam berdasarkan faktor lingkungan.

F. Penegasan Konsep

Sistem pendukung keputusan menganalisa kesesuaian jenis vegetasi *Mangrove* menggunakan Electre merupakan sistem yang akan membantu *user* mengambil keputusan untuk memilih jenis *Mangrove* yang cocok untuk ditanam ataupun proses *reboisasi* dengan memperhitungkan faktor-faktor dari hasil analisa lingkungan yakni :

1. Salinitas

Salinitas atau tingkat keasinan dapat diukur dengan *Refraktometer* dengan langkah sebagai berikut :

- a. Buka penutup *refraktometer* lalu teteskan *refraktometer* dengan aquadest.

- b. Bersihkan tetesan aquadest tadi dengan tisyu dan jangan sampai ada sisa aquadest yang tertinggal.
- c. Teteskan air sampel yang ingin diketahui kadar salinitasnya.
- d. Lalu arahkan refrakto meter ke arah cahaya matahari langsung.
- e. Akan tampak sebuah bidang berwarna biru dan putih.
- f. Garis batas antara kedua bidang itulah yang menunjukkan kadar salinitasnya.
- g. Catat hasil nilai salinitasnya.

2. Kekuatan Ombak dan Angin

Kekuatan angin yang mempengaruhi ombak dapat diukur menggunakan *Anemometer* yang diletakkan di tempat yang terbuka. Aliran udara atau angin yang bertiup akan memutarakan mangkuk atau baling-baling yang ada pada *Anemometer*. Semakin besar kekuatan atau kecepatan angin yang berhembus, maka akan semakin cepat pula putaran baling-baling tersebut. Kecepatan angin dapat diketahui dengan cara menghitung jumlah putaran baling-baling per detiknya.

3. Kandungan Lumpur dan Pasir

Menghitung kandungan lumpur dan pasir dapat dilakukan dengan langkah-langkah sebagai berikut :

- a. Isi gelas ukur dengan pasir yang telah disediakan sampai 450 cc kemudian ditambah dengan air sampai 900 cc.
- b. Tutup gelas ukur sampai rapat kemudian dikocok-kocok.

- c. Diamkan selama kurang lebih 1 jam.
- d. Catat berapa endapan lumpur dan pasir.

4. Frekuensi Penggenangan

Frekuensi penggenangan dapat diukur dari lama penggenangan per bulannya.

5. Kepadatan Penduduk

Kepadatan penduduk dapat dilihat dari wilayah sekitar Ekosistem *Mangrove* itu sendiri.

BAB II

TINJAUAN PUSTAKA

A. Teori Sistem Pendukung Keputusan

1. Pengambilan Keputusan

Menurut Herdino Ambargo. 2013, Keputusan adalah hasil pemecahan masalah yang dihadapinya dengan tegas. Hal ini berkaitan dengan jawaban atas pertanyaan-pertanyaan tentang apa yang harus dilakukan dan mengenai unsur-unsur perencanaan. Dapat juga dikatakan bahwa keputusan itu sesungguhnya merupakan hasil proses pemikiran yang berupa pemilihan satu diantara beberapa alternatif yang dapat digunakan untuk memecahkan masalah yang dihadapinya.

Sedangkan menurut Prajudi Atmosudirjo. 1979, mengatakan Keputusan adalah suatu pengakhiran daripada proses pemikiran tentang suatu masalah atau problem untuk menjawab pertanyaan apa yang harus diperbuat guna mengatasi masalah tersebut, dengan menjatuhkan pilihan pada suatu alternatif.

Selanjutnya menurut AF Stoner dalam Hasan (2002), “Keputusan adalah pemilihan di antara alternatif – alternatif.”

Pengertian ini mengandung tiga pengertian, yaitu :

- a. Ada pilihan atas dasar logika atau pertimbangan.
- b. Ada beberapa alternatif yang harus dan dipillih salah satu yang terbaik.

- c. Ada tujuan yang ingin dicapai dan keputusan itu semakin mendekati pada tujuan tersebut.

Setelah mengetahui definisi dari keputusan maka selanjutnya akan dikemukakan definisi dari pengambilan keputusan :

Menurut George R. Terry (2008), pengambilan keputusan adalah pemilihan alternatif perilaku (kelakuan) tertentu dari dua atau lebih alternatif yang ada

2. Dasar Pengambilan Keputusan

Menurut George R. Terry (2008), dasar-dasar pengambilan keputusan adalah :

- a. Intuisi

Suatu proses bawah sadar/tdk sadar yang timbul atau tercipta akibat pengalaman yang terseleksi. Pengambilan keputusan yang berdasarkan atas intusi atau perasaan memiliki sifat subjektif, sehingga mudah terkena pengaruh.

- b. Pengalaman

Pengambilan keputusan berdasarkan pengalaman memiliki manfaat bagi pengetahuan praktis. Karena pengalaman seseorang dapat mempekira-kan keadaan sesuatu, dapat memperhitungkan untung ruginya, baik-buruknya keputusan yang akan dihasilkan. Karena pengalaman, seseorang yang menduga masalahnya walaupun hanya dengan melihat sepintas saja mungkin sudah dapat menduga cara penyelesaiannya.

c. Fakta

Pengambilan keputusan berdasarkan fakta dapat memberikan keputusan yang sehat, solid, dan baik. Dengan fakta, maka tingkat kepercayaan terhadap pengambilan keputusan dapat lebih tinggi, sehingga orang dapat menerima keputusan-keputusan yang dapat dibuat dengan rela dan lapang dada.

d. Wewenang

Pengambilan keputusan berdasarkan wewenang biasanya dilakukan oleh pemimpin terhadap bawahannya atau orang yang lebih tinggi kedudukannya kepada orang lebih rendah kedudukannya.

e. Rasional

Pada pengambilan keputusan yang berdasarkan rasional, keputusan yang dihasilkan bersifat objektif, logis, lebih transparan, konsisten untuk memaksimalkan hasil atau nilai dalam batas kendala tertentu, sehingga dapat dikatakan mendekati kebenaran atau sesuai dengan apa yang diinginkan.

3. Pengertian Sistem Pendukung Keputusan

Menurut Turban (2005), Sistem Pendukung Keputusan merupakan sistem yang dapat membantu para pengambil keputusan dengan memanfaatkan komputer dalam pengambilan keputusan. Secara lebih mendalam, para ahli yaitu Keen dan Scott- Marton mendefinisikan

sistem pendukung keputusan sebagai sistem pendukung keputusan yang selanjutnya disebut SPK yang digabungkan dengan sumber daya intelektual manusia dan kemampuan komputer untuk meningkatkan kualitas keputusan. Sistem ini merupakan sistem pendukung keputusan berbasis komputer untuk pembuat keputusan manajemen yang berhubungan dengan masalah semi – terstruktur.

4. Komponen Sistem Pendukung Keputusan

Sistem pendukung keputusan terdiri dari 3 komponen utama atau subsistem yaitu (Dadan Umar Daihani, 2001) :

a. Sub Sistem Data (*Database*)

Subsistem data merupakan komponen sistem pendukung keputusan penyedia data bagi sistem. Data dimaksud disimpan dalam suatu pangkalan data (*database*) yang diorganisasikan suatu sistem yang disebut sistem manajemen pangkalan data (*Data Base Manajemen System/DBMS*).

b. Subsistem Model (*Model Subsistem*)

c. Subsistem Dialog (*User Sistem Interface*)

Sistem pendukung keputusan memiliki fasilitas yang mampu mengintegrasikan system terpasang dengan pengguna secara interaktif. Umar Daihani (2001).

B. Teori *Mangrove*

Menurut Kusmana (2002), mengemukakan bahwa *Mangrove* adalah suatu komunitas tumbuhan atau suatu individu jenis tumbuhan yang membentuk komunitas tersebut di daerah pasang surut. Hutan *Mangrove* adalah tipe hutan yang secara alami dipengaruhi oleh pasang surut air laut, tergenang pada saat pasang naik dan bebas dari genangan pada saat pasang rendah. Ekosistem *Mangrove* adalah suatu sistem yang terdiri atas lingkungan biotik dan abiotik yang saling berinteraksi di dalam suatu habitat *Mangrove*.

Hutan *Mangrove* memberikan perlindungan kepada berbagai organisme baik hewan darat maupun hewan air untuk bermukim dan berkembang biak. Hutan *Mangrove* dipenuhi pula oleh kehidupan lain seperti mamalia, amfibi, reptil, burung, kepiting, ikan, primata, serangga dan sebagainya. Selain menyediakan keanekaragaman hayati (*biodiversity*), ekosistem *Mangrove* juga sebagai plasma nutfah (*geneticpool*) dan menunjang keseluruhan sistem kehidupan di sekitarnya. Habitat *Mangrove* merupakan tempat mencari makan (*feeding ground*) bagi hewan-hewan tersebut dan sebagai tempat mengasuh dan membesarkan (*nursery ground*), tempat bertelur dan memijah (*spawning ground*) dan tempat berlindung yang aman bagi berbagai ikan-ikan kecil serta kerang (*shellfish*) dari predator. Beberapa manfaat hutan *Mangrove* dapat dikelompokkan sebagai berikut:

A. Manfaat / Fungsi Fisik :

1. Menjaga agar garis pantai tetap stabil
2. Melindungi pantai dan sungai dari bahaya erosi dan abrasi.

3. Menahan badai/angin kencang dari laut
4. Menahan hasil proses penimbunan lumpur, sehingga memungkinkan terbentuknya lahan baru.
5. Menjadi wilayah penyangga, serta berfungsi menyaring air laut menjadi air daratan yang tawar
6. Mengolah limbah beracun, penghasil O₂ dan penyerap CO₂.

B. Manfaat / Fungsi Biologik :

1. Menghasilkan bahan pelapukan yang menjadi sumber makanan penting bagi plankton, sehingga penting pula bagi keberlanjutan rantai makanan.
2. Tempat memijah dan berkembang biaknya ikan-ikan, kerang, kepiting dan udang.
3. Tempat berlindung, bersarang dan berkembang biak dari burung dan satwa lain.
4. Sumber plasma nutfah & sumber genetik.
5. Merupakan habitat alami bagi berbagai jenis biota.

C. Manfaat / Fungsi Ekonomik :

1. Penghasil kayu : bakar, arang, bahan bangunan.
2. Penghasil bahan baku industri : pulp, tanin, kertas, tekstil, makanan, obat-obatan, kosmetik, dll
3. Penghasil bibit ikan, nener, kerang, kepiting, bandeng melalui pola tambak silvofishery
4. Tempat wisata, penelitian & pendidikan.

D. Faktor-faktor yang dapat mempengaruhi pertumbuhan *Mangrove* adalah :

1. Gerakan gelombang yang minimal, agar jenis tumbuhan *Mangrove* dapat menancapkan akarnya
2. Salinitas payau (pertemuan air laut dan tawar)
3. Endapan Lumpur
4. *Zona intertidal* (pasang surut) yang lebar

Kerusakan *Mangrove* dapat terjadi secara alamiah atau melalui tekanan masyarakat. Secara alami umumnya kadar kerusakannya jauh lebih kecil daripada kerusakan akibat ulah manusia. Kerusakan alamiah timbul karena peristiwa alam seperti adanya topan badai atau iklim kering berkepanjangan yang menyebabkan akumulasi garam dalam tanaman. Banyak kegiatan manusia di sekitar kawasan hutan *Mangrove* yang berakibat perubahan karakteristik fisik dan kimiawi di sekitar habitat *Mangrove* sehingga tempat tersebut tidak lagi sesuai bagi kehidupan dan perkembangan flora dan fauna di hutan *Mangrove*. Tekanan tersebut termasuk kegiatan reklamasi, pemanfaatan kayu *Mangrove* untuk berbagai keperluan, misalnya untuk pembuatan arang dan sebagai bahan bangunan, pembuatan tambak udang, reklamasi dan tempat pembuangan sampah di kawasan *Mangrove* yang menyebabkan polusi dan kematian pohon.

Akibat yang terjadi bila hutan *Mangrove* rusak adalah :

1. Abrasi pantai
2. Mengakibatkan intrusi air laut lebih jauh ke daratan
3. Potensi perikanan menurun

4. Kehidupan satwa liar terganggu

5. Sumber mata pencaharian penduduk setempat berkurang

E. Kesesuaian Jenis Vegetasi *Mangrove*

Table 2.1 Kesesuaian Jenis Vegetasi *Mangrove*

Jenis Mangrove	Salinitas	Toleransi Kekuatan Ombak dan Angin	Toleransi Kandungan Pasir	Toleransi Lumpur	Frekuensi Penggenangan
Api-api	10-30	MD	TS	S	-
Nipah	0-10	STS	TS	S	20 Hari / Bulan
Bayur Laut	10-30	STS	MD	MD	9 Hari / Bulan
Nyirih	10-30	TS	MD	MD	9 Hari / Bulan
Tanjang Merah	10-30	TS	TS	MD	10-19 Hari / Bulan
Tancang	10-30	TS	MD	S	10-19 Hari / Bulan
Bius	10-30	TS	MD	S	10-19 Hari / Bulan
Tinjang	10-30	MD	MD	S	20 Hari / Bulan
Tongke Besar	10-30	MD	S	S	20 Hari / Bulan
Bakau	10-30	S	MD	S	20 Hari / Bulan

Keterangan

S	: Sesuai	Konversi Nilai	4
MD	: Moderat	Konversi Nilai	3
TS	: Tidak Sesuai	Konversi Nilai	2
STS	: Sangat Tidak Sesuai	Konversi Nilai	1

F. Jenis – jenis *Mangrove*

Jenis – jenis *Mangrove* diantaranya :

1. Api-api

Api-api adalah nama sekelompok tumbuhan dari marga *Avicennia*, suku *Acanthaceae*. Api-api biasa tumbuh di tepi atau dekat laut sebagai bagian dari komunitas hutan bakau. *Avicennia* merupakan pohon *Mangrove* pionir, jadi mudah sekali dikenal. Tumbuhnya selalu di tepi laut maupun di tepi sungai. *Avicennia* merupakan pohon tinggi yang berukuran sedang sampai besaf.

Getah yang keluar dari kulit batangnya dilaporkan mempunyai khasiat sebagai *aphrodisiac* (pembangkit gairah), kontraseptif dan obat sakit gigi. Biji mudanya digunakan sebagai obat untuk mematangkan bisul. Buah dan bijinya apabila direbus dapat dimakan. Apabila ditumbuk halus dan dicampur dengan salep dapat menjadi obat luka yang manjur, terutama luka karena terbakar. Daun muda dan pucuk atau sirung rasanya sangat enak sebagai lalap atau dibuat sayur lodeh. Selain itu, abu dari kayu jenis-jenis *Avicennia* dapat digunakan sebagai sabun.

2. Nipah

Nipah adalah sejenis palem (palma) yang tumbuh di lingkungan hutan bakau atau daerah pasang-surut dekat tepi laut/Habitatnya adalah daerah rawa yang berair payau atau daerah pasang surut di dekat pantai.

Pemanfaatan Nipah. Berbagai bagian tumbuhan nipah (*Nypa fruticans*) telah dimanfaatkan manusia sejak lallma. Daun nipah dapat

dimanfaatkan untuk membuat atap rumah, anyaman dinding rumah, dan berbagai kerajinan seperti tikar, topi dan tas keranjang. Pada zaman dulu, daun nipah juga dimanfaatkan sebagai media tulis di samping daun lontar.

Batang, dan tangkai daun nipah dapat digunakan sebagai kayu bakar. Lidinya dimanfaatkan sebagai sapu lidi, dan berbagai anyaman. Tandan bunga yang belum mekar dapat disadap untuk diambil air niranya. Air nira ini dapat dijadikan gula nira, difermentasi menjadi cuka dan tuak, juga sebagai bahan baku etanol yang dapat dijadikan bahan bakar nabati pengganti bahan bakar minyak bumi.

Tunas nipah dapat dimakan dan buah nipah yang masih muda dapat dijadikan semacam kolang-kaling untuk campuran minuman, kolak, maupun dijadikan manisan. Sedangkan bijinya yang telah tua dapat ditumbuk untuk diambil tepungnya

3. Bayur Laut

Nama sejenis pohon penghuni hutan pantai, rawa-rawa pantai, dan hutan bakau, anggota suku Malvaceae. Pohon ini menghasilkan kayu yang berkualitas baik, yang dalam perdagangan dikenal sebagai kayu dungun. Dipencarkan oleh aliran air dan gelombang laut, dungun kecil tersebar luas dari pesisir timur Afrika, India, kawasan Malaysia, hingga pulau-pulau di Pasifik.

4. Nyirih

Dapat sebagai bedak dingin, obat dan lulur yang sehat untuk kulit.

yang tumbuh pada ekosistem bakau yang berlumpur karena terkena semburan air laut, khususnya pada tempat yang mengarah di bagian atas ekosistem ini. Di ekosistem bakau di Indonesia, *Xylocarpus granatum* telah diketahui toleran terhadap lingkungan dengan salinitas 0.1-3%. Seringkali tumbuh mengelompok dalam jumlah besar. Individu yang telah tua seringkali ditumbuhi oleh epifit dan tumbuhan ini termasuk kedalam tumbuhan perintis/reklamasi

5. Tanjung Merah

Salah satu spesies *Mangrove* yang paling penting dan tersebar luas di Pasifik. *Mangrove* ini ditemukan di daerah pasang surut daerah tropis Pasifik dari Asia Tenggara ke Kepulauan Ryukyu Jepang selatan. *Mangrove* berdaun besar ini tumbuh subur di berbagai kondisi intertidal, termasuk tingkat salinitas yang rendah sampai tingkat salinitas tinggi, dan mentolerir kondisi saat terjadi banjir dan jenis tanah lainnya. Kebanyakan *Mangrove* jenis ini terletak di tengah dan di atas zona pasang surut. *Mangrove* pada umumnya diyakini mempunyai peranan yang sangat penting dalam perlindungan garis pantai, meningkatkan kualitas air di lingkungan dekat pantai (terumbu karang), tempat berlindung ikan karang dan spesies lainnya, dan mendukung rantai makanan laut.

6. Tancang

Tancang jenis Lindur (*Bruguiera gymnorrhiza*), merupakan salah satu jenis *Mangrove* yang memiliki nilai ekonomis tinggi, karena buahnya dapat dijadikan sebagai tepung yang kemudian bisa diolah menjadi panganan atau jajanan *Mangrove* yang sangat lezat dan bergizi. Salah satu spesies *Mangrove* yang paling penting dan tersebar luas di Pasifik..

7. Bius

Tumbuh mengelompok dalam jumlah besar, biasanya pada tanah liat di belakang zona *Avicennia*, atau di bagian tengah vegetasi *Mangrove* kearah laut. Jenis ini juga memiliki kemampuan untuk tumbuh pada tanah/substrat yang baru terbentuk dan tidak cocok untuk jenis lainnya. Kemampuan tumbuhnya pada tanah liat membuat pohon jenis ini sangat bergantung kepada akar nafas untuk memperoleh pasokan oksigen yang cukup, dan oleh karena itu sangat responsif terhadap penggenangan yang berkepanjangan. Memiliki buah yang ringan dan mengapung sehingga penyebarannya dapat dibantu oleh arus air, tapi pertumbuhannya lambat. Perbungaan terjadi sepanjang tahun.

8. Tinjang

Tinjang memiliki Ketinggian pohon dapat mencapai 30 m dengan diameter batang mencapai 50 cm, memiliki perakaran yang khas hingga dapat mencapai 5 m, dan kadang-kadang memiliki akar udara yang

keluar dari cabang. Kulit kayu berwarna abu-abu tua dan berubah-ubah. Tinjang memiliki ciri-ciri warna daun berwarna hijau tua, bentuk elips meruncing. pucuk daun berwarna merah. Bunga berwarna merah kecoklatan dengan formasi 2-4 bunga per kelompok.³ Batang agak mengkilap.

9. Tongke Besar

Kulit kayu halus, bercelah, berwarna abu-abu hingga hitam. Memiliki akar tunjang dengan panjang hingga 3 m, dan akar udara yang tumbuh dari cabang bawah. Daun berkulit, berbintik teratur di lapisan bawah.

10. Bakau

Tumbuhan ini memiliki ciri-ciri yang menyolok berupa akar tunjang yang besar dan berkayu, pucuk yang tertutup daun penumpu yang meruncing, serta buah yang berkecambah serta berakar ketika masih di pohon (*vivipar*).

Pohon tinggi dengan akar tunggang yang biasanya abortif, akar lateral atau banyak. Batang menyilinder, warna hampir hitam atau kemerahan, permukaan kasar atau kadang-kadang bersisik, dengan retak-retak melintang yang menonjol hampir melingkari batang.

Daun melonjong, dengan titik-titik hitam yang terlihat pada permukaan bawah, warna hijau mengkilap di atas dan lebih pudar di bawah permukaan daun. Perbungaan aksiler, menggarpu, berwarna kuning muda sampai hampir putih; daun mahkota melanset,

kekuningan muda. Buah matang bani membulat telur memanjang, coklar-hijau pudar.

G. Faktor penunjang keberhasilan penanaman *Mangrove*

Sebelum melakukan penanaman, harus diperhatikan beberapa faktor fisik penunjang keberhasilan penanaman: keadaan pasang surut, musim ombak dan kesesuaian jenis tanaman dengan lingkungannya. Selain itu, faktor pelibatan masyarakat (termasuk perempuan dan anak-anak) dalam kegiatan penanaman juga menentukan keberhasilan penanaman. Dengan keterlibatan ini akan timbul rasa memiliki dan keinginan menjaga dan memelihara tanaman.

Penanaman sebaiknya dilakukan pada saat air laut surut agar memudahkan penanaman dan jarak antar tanaman dapat segera diketahui apakah seragam atau tidak. Untuk mengetahui kondisi pasang surut air laut ini, beberapa hari sebelum penanaman perlu diamati waktu dan lama pasang dan surut.

Informasi dari masyarakat tentang kondisi ini akan sangat bermanfaat. Untuk penanaman dipinggir laut, terutama di daerah pantai yang menghadap laut terbuka, musim ombak besar perlu diketahui agar setelah penanaman bibit/benih tidak hilang diterjang ombak. Untuk daerah-daerah pantai penanaman sebaiknya tidak dilakukan pada musim barat karena saat tersebut ombaknya besar.

Penanaman pada musim timur akan lebih baik karena ombaknya relatif kecil sehingga resiko bibit/benih hilang diterjang gelombang laut kecil. Waktu penanaman ini sebaiknya didiskusikan dan disepakati bersama dengan masyarakat karena merekalah yang lebih menguasai kondisi setempat. Kesesuaian jenis tanaman dengan lingkungannya perlu diperhatikan karena akan mempengaruhi tingkat keberhasilan penanaman.

Faktor-faktor yang perlu diperhatikan untuk kesesuaian jenis ini adalah salinitas, frekuensi penggenangan, kandungan pasir dan lumpur, dan kekuatan ombak dan angin (M. Khazali, 1999).

H. Penentuan jarak tanaman *Mangrove*

Jarak tanam tergantung lokasi dan tujuan penanaman. Penanaman di pinggir laut dengan tujuan melindungi pantai dari abrasi atau sebagai jalur hijau, jarak tanamnya adalah 1 x 1 meter. Jumlah baris tanaman tergantung kondisi pantai, namun diusahakan sebanyak mungkin.

Dengan semakin banyaknya tegakan tanaman akan semakin besar kemampuannya untuk melindungi pantai dari abrasi, semakin besar kemampuannya menyuburkan pantai, dan semakin banyak ruang untuk perlindungan dan tumbuh bagi biota air seperti ikan dan udang.

Penanaman di pinggir sungai atau saluran-saluran air menuju tambak dengan tujuan melindungi tanggul atau jalur hijau, apabila hanya 1 baris, jarak antar tanaman dapat 1 meter atau 1.5 meter. Apabila lebih dari 1 baris, jarak tanam dapat 1 x 1 meter atau 1.5 x 1.5 meter. Apabila dilokasi penanaman banyak penjala, pencari udang atau kepiting, maka

jarak antar tanaman sebaiknya diperbesar menjadi 2 meter atau 2 x 2 meter. Hal ini untuk memberi ruang bagi mereka dan alat yang digunakan agar tidak merusak tanaman.

Jarak antar tanaman di tambak dengan tujuan untuk melindungi tanggul dapat 1 meter, 1.5 meter atau 2 meter. Setelah tanaman membesar dan dirasakan terlalu rapat, dapat dilakukan penjarangan sehingga jarak antar tanaman menjadi 2 meter atau 3 meter.

Penanaman di tengah tambak (terutama tambak bandeng) jarak tanaman dapat 1.5 x 1.5 meter, 2 x 2 meter atau 2 x 3 meter. Setelah tanaman membesar, dapat dijarangkan menjadi 3 x 3 meter, 2 x 4 meter atau 4 x 3 meter (M. Khazali, 1999).

C. *Elimination and Choice Expressing Reality (Electre)*

Metode ELECTRE termasuk pada metode analisis pengambilan keputusan multikriteria yang berasal dari Eropa pada tahun 1960an. ELECTRE adalah akronim dari *Elimination Et Choix Traduisant la Réalité* atau dalam bahasa Inggris berarti *Elimination and Choice Expressing Reality* (wikipedia).

Menurut Janko dan Bernoider (2005:11), ELECTRE merupakan salah satu metode pengambilan keputusan multikriteria berdasarkan pada konsep outranking dengan menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai. Metode ELECTRE digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi dan alternatif yang sesuai dapat dihasilkan.

Dengan kata lain, ELECTRE digunakan untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria yang dilibatkan (Setiyawati). Suatu alternatif dikatakan mendominasi alternatif lainnya jika satu atau lebih kriterianya melebihi (dibandingkan dengan kriteria alternatif yang lain) dan sama dengan kriteria lain yang tersisa (Kusumadewi, 2006).

Langkah-langkah yang dilakukan dalam penyelesaian masalah menggunakan metode Electre adalah sebagai berikut:

a. Normalisasi matriks keputusan.

Dalam prosedur ini, setiap atribut diubah menjadi nilai yang comparable. Setiap normalisasi dari nilai x_{ij} dapat dilakukan dengan rumus :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \text{ untuk } i = 1, 2, 3, \dots, m \text{ dan } j = 1, 2, 3, \dots, n.$$

Sehingga didapat matriks R hasil normalisasi,

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1n} \\ r_{21} & r_{22} & \dots & r_{2n} \\ \vdots & \vdots & \dots & \vdots \\ r_{m1} & r_{m2} & \dots & r_{mn} \end{bmatrix}$$

R adalah matriks yang telah dinormalisasi, dimana m menyatakan alternatif, n menyatakan kriteria dan r_{ij} adalah normalisasi pengukuran pilihan dari alternatif ke- i dalam hubungannya dengan kriteria ke- j .

b. Pembobotan pada matriks yang telah dinormalisasi

Setelah dinormalisasi, setiap kolom dari matriks R dikalikan dengan bobot-bobot (w_j) yang ditentukan oleh pembuat keputusan. Sehingga, *weighted normalized matrix* adalah yang ditulis sebagai:

$$V = R.W$$

$$\begin{bmatrix} v_{11} & v_{12} & \dots & v_{1n} \\ v_{21} & v_{22} & \dots & v_{2n} \\ \vdots & \vdots & \dots & \vdots \\ v_{m1} & v_{m2} & \dots & v_{mn} \end{bmatrix} = \begin{bmatrix} w_1 r_{11} & w_2 r_{12} & \dots & w_n r_{1n} \\ w_1 r_{21} & w_2 r_{22} & \dots & w_n r_{2n} \\ \vdots & \vdots & \dots & \vdots \\ w_1 r_{m1} & w_2 r_{m2} & \dots & w_n r_{mn} \end{bmatrix}$$

dimana W adalah

$$W = \begin{bmatrix} w_1 & 0 & \dots & 0 \\ 0 & w_2 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & w_n \end{bmatrix}$$

c. Menentukan himpunan *concordance* dan *discordance*

Untuk setiap pasang dari alternatif k dan l ($k, l = 1, 2, 3, \dots, m$ dan $k \neq l$) kumpulan J kriteria dibagi menjadi dua himpunan bagian, yaitu *concordance* dan *discordance*. Sebuah kriteria dalam suatu alternatif termasuk *concordance* jika:

$$C_{kl} = \{j, v_{kj} \geq v_{lj}\}, \text{ untuk } j = 1, 2, 3, \dots, n.$$

Sebaliknya, komplementer dari himpunan bagian *concordance* adalah himpunan *discordance*, yaitu bila:

$$D_{kl} = \{j, v_{kj} < v_{lj}\}, \text{ untuk } j = 1, 2, 3, \dots, n.$$

d. Menghitung matriks *concordance* dan *discordance*

a. Menghitung matriks *concordance*

Untuk menentukan nilai dari elemen-elemen pada matriks *concordance* adalah dengan menjumlahkan bobot-bobot yang termasuk pada himpunan *concordance*, secara matematisnya adalah sebagai berikut:

$$C_{kl} = \sum_{j \in C_{kl}} w_j$$

b. Menghitung matrix *discordance*

Untuk menentukan nilai dari elemen-elemen pada matriks *discordance* adalah dengan membagi maksimum selisih kriteria yang termasuk ke dalam himpunan bagian *discordance* dengan maksimum selisih nilai seluruh kriteria yang ada, secara matematisnya adalah sebagai berikut:

$$d_{ki} = \frac{\max\{|v_{kj} - v_{lj}|\}_{j \in D_{ki}}}{\max\{|v_{kj} - v_{lj}|\}_{\Psi_j}}$$

e. Menentukan matriks dominan *concordance* dan *disordance*

a. Menghitung matriks dominan *concordance*

Matriks F sebagai matriks dominan *concordance* dapat dibangun dengan bantuan nilai *threshold*, yaitu dengan membandingkan setiap nilai elemen matriks *concordance* dengan nilai *threshold*.

$$C_{kl} \geq c$$

Dengan nilai *threshold* (c) adalah :

$$\underline{c} = \frac{\sum_{k=1}^m \sum_{l=1}^m c_{kl}}{m(m-1)}$$

Sehingga elemen matriks F ditentukan sebagai berikut :

$$f_{kl} = \begin{cases} 1, & \text{jika } c_{kl} \geq \underline{c} \\ 0, & \text{jika } c_{kl} < \underline{c} \end{cases}$$

b. Menghitung matriks dominan *disordance*

Matriks G sebagai matriks dominan *disordance* dapat dibangun dengan bantuan nilai *threshold* d :

$$\underline{d} = \frac{\sum_{k=1}^m \sum_{l=1}^m d_{kl}}{m(m-1)}$$

dan elemen matriks G ditentukan sebagai berikut:

$$g_{kl} = \begin{cases} 1, & \text{jika } d_{kl} \geq \underline{d} \\ 0, & \text{jika } d_{kl} < \underline{d} \end{cases}$$

f. Menentukan *aggregate dominance matrix*

Matriks E sebagai *aggregate dominance matrix* adalah matriks yang setiap elemennya merupakan perkalian antara elemen matriks F

dengan elemen matriks G yang bersesuaian, secara matematis dapat dinyatakan sebagai :

$$e_{kl} = f_{kl} \times g_{kl}$$

g. Eliminasi alternatif yang *less favourable*

Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila $e_{kl} = 1$ maka alternatif A_k merupakan alternatif yang lebih baik daripada A_l . Sehingga, baris dalam matriks E yang memiliki jumlah $e_{kl}=1$ paling sedikit dapat dieliminasi. Dengan demikian, alternative terbaik adalah alternatif yang mendominasi alternatif lainnya (Setiyawati).

D. Unified Modelling Language (UML)

1. Pengertian *Unified Modelling Language* (UML)

Menurut Sholih (2006) UML merupakan singkatan dari *Unified Modelling Language* yaitu sekumpulan pemodelan konvensi yang digunakan untuk menentukan atau menggambarkan sebuah sistem perangkat lunak dalam kaitannya dengan objek.

a. Konsep Objek

Objek dalam *software analysis design* adalah sesuatu berupa konsep, benda atau sesuatu yang membedakannya dengan lingkungannya. Secara sederhana objek adalah mobil, manusia, *alarm* dan lain-lainnya. Tapi objek dapat pula merupakan sesuatu yang abstrak yang hidup didalam sistem seperti tabel, *database*, *even*, *system messages*.

Objek dikenali dari keadaannya dan juga operasinya. Sebagai contoh sebuah mobil dikenali dari warnanya, bentuknya, sedangkan manusia dari suaranya. Alasan mengapa saat ini pendekatan dalam pengembangan *software* dengan *object-oriented*, pertama adalah *scalability* dimana objek lebih mudah dipakai untuk menggambarkan sistem yang besar dan kompleks. Kedua *dynamic modeling*, adalah dapat dipakai untuk pemodelan sistem dinamis atau *real time*.

b. Teknik dasar *object-object analysis/design*

Dalam dunia pemodelan, metodologi implementasi objek walaupun terkait kaidah-kaidah standar, namun teknik pemilihan objek tidak terlepas pada subjektivitas *software analyst & designer*. Beberapa objek akan diabaikan dan beberapa objek menjadi perhatian untuk diimplementasikan didalam sistem. Hal ini sah-sah saja karena kenyataan bahwa suatu permasalahan sudah tentu memiliki lebih dari satu solusi. Ada 3 (tiga) teknik/konsep dasar dalam *object-oriented Analysis/Design*, yaitu pemodulan (*encapsulation*), peniruan (*inheritance*), dan *polymorphism*.

2. Sejarah singkat *Unified Modelling Language* (UML)

Unified Modelling Language (UML) adalah sebuah bahasa yang berdasarkan pada grafik/gambar yang berguna untuk memvisualisasi, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan *software* berbasis *object-oriented*. UML sendiri juga memberikan standar penulisan sebuah sistem *blue print*, yan meliputi

konsep bisnis proses, penulisan kelas-kelas dalam bahasa program yang spesifik, skema *database*, dan komponen yang diperlukan dalam sistem *software*. Pendekatan analisa dan rancangan dengan menggunakan model *object-oriented* mulai diperkenalkan sekitar pertengahan 1970 hingga akhir 1980 dikarenakan pada saat itu aplikasi software sudah meningkan dan mulai kompleks. Jumlah yang menggunakan *object-oriented* mulai diuji cobakan diaplikasikan antara 1989 hingga 1994, seperti halnya oleh Geradi Booch dari *Rational Software CO*, dikenalkan dengan *object-oriented software engenering*, serta Jems Rumbaugh dari *general electric*, dikenal dengan *object modeling technique*.

Kelamahan saat itu disadari oleh Booch maupun Rumbaugh adalah tidak adanya standar penggunaan model yang berbasis *object-oriented*, ketika mereka bertemu ditemani rekan lainnya Ivar Jacobson dari *Objectory* mulai mendiskusikan untuk mengadopsi masing-masing pendekatan metode *object-oriented* untuk membuat sebuah model bahasa yan *uniform/seragam* yang disebut UML (*Unified Modeling Language*) dan dapat digunakan diseluruh dunia.

Secara resmi bahasa UML dimulai pada bulan oktober 1994, ketika Rumbaugh bergabung Booch untuk membuat sebuah *project* pendekan metode yang *uniform/seragam* dari masing-masing metode mereka. Saat itu baru dikembangkan draft metode UML versi 0.8 dan diselesaikan serta di *release* pada bulan oktober 1995. Bersama dengan itu, Jacobson bergabung dan UML tersebut diperkaya ruang lingkupnya dengan metode

object-oriented software engenering sehingga muncul *release 0.9* pada bulan juni 1996.

Hingga saat ini sejak Juni 1998 UML versi 1.3 telah diperkaya dan direspon oleh *Object Management Group, Anderson Consulting, Ericsson, Platinum Technology, Object Time Limeted*, dll serta dipelihara oleh *Object Management Group* yang dipimpin oleh Cris Kobryn.

UML adalah standar yang dibuat oleh *Object Management Group*, sebuah badan yang bertugas mengeluarkan standar-standar teknologi *object-oriented* dan *software component*.


3. Dokumentasi *Unified Modelling Language (UML)*

Dokumentasi analisis sistem dengan menggunakan UML adalah sebagai berikut.


a. *Use Case Diagram*




Use Case diagram merupakan teknik untuk merekam persyaratan fungsionalitas sebuah sistem. *Use case diagram* mendeskripsikan interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri, dengan member sebuah narasi tentang bagaimana sistem tersebut digunakan.

Tabel 2.2 Simbol *Use Case*

No.	Simbol	Nama	Penjelasan
.		<i>Actor</i>	Menspesifikasikan himpunan

			peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	----->	<i>Dependancy</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).
	←	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek

			induk (<i>ancestor</i>).
.		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
.		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
.		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
.		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara

			terbatas.
.		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor.
.		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemennya (sinergi).
0.		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan



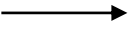
			mencerminkan suatu sumber daya komputasi.
--	--	--	---


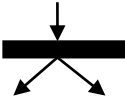


Sumber : Sholiq (2006)

b. *Activity Diagram*

Diagram aktivitas adalah teknik untuk mendeskripsikan logika procedural, proses bisnis, dan aliran kerja dalam suatu program. Diagram aktivitas mempunyai peran seperti halnya bagan alir (*flowchart*), akan tetapi perbedaannya dengan bagan alir adalah diagram aktivitas dapat mendukung perilaku parallel sedangkan bagan alir tidak bias.

Tabel 2.3 Simbol *Activity Diagram*

No	Symbol	Nama	Keterangan
1		<i>Initial state</i>	Proses dimulai pertama kali didalam activity
		<i>State</i>	Aktivitas yang terjadi dalam activity
3		<i>Control flow</i>	Urutan perpindahan suatu aktivitas

4		<i>Decision</i>	Menggambarkan cabang suatu keputusan
5		<i>Transition (fork)</i>	Kegiatan yang dilakukan secara paralel
6		<i>Transition (join)</i>	Menunjukkan kegiatan yang digabungkan
7		<i>Final state</i>	Proses terakhir didalam activity

Sumber : Sholih (2006)


c. *Sequence Diagram*



Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan use case diagram. Memerlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case .

Tabel 2.4 Simbol *Sequence Diagram*

N O	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.

2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

Sumber : Sholiq (2006)

d. *Class diagram*

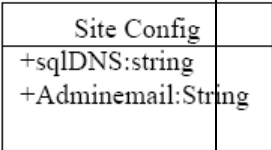
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek.

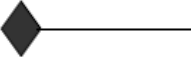

Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).


Class diagram menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok : Nama (dan stereotype), Atribut, Metoda.

Tabel 2.5 Simbol *Class Diagram*

SIMBOL	NAMA	KETERANGAN
 <p>The symbol is a rectangular box representing a class. It is divided into three horizontal sections. The top section contains the text "Site Config". The middle section contains the text "+sqlDNS:string". The bottom section contains the text "+Adminemail:String".</p>	<p><i>Class</i></p>	<p><i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi obyek. Sebuah class digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian.</p>
<p><u>1..n</u> Owned by <u>1</u></p>	<p><i>Assosiation</i></p>	<p>Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum</p>

		antara 2 class, dan dilambangkan oleh sebuah garis yang terhubung antara 2 class.
	Compo sition	Jika sebuah class tidak bisa berdiri sendiri dan harus merupakan bagian dari class yang lain, maka class tersebut memiliki relasi <i>Composition</i> terhadap class tempat dia bergantung tersebut.
	Depen dency	Kadangka la sebuah class

		<p>menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i>. Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.</p>
	<p style="text-align: center;">Aggregation</p>	<p style="text-align: center;"><i>Aggregation</i> menunjukkan</p>

		keseluruhan bagian <i>relationship</i>
←	Generalization	Sebuah relasi <i>generalization</i> sepadan dengan sebuah relasi <i>inheritance</i> pada konsep berorientasi obyek.

Sumber : Sholiq (2006)

E. Kamus Data

Menurut Yakub (2012), kamus data merupakan daftar elemen data yang terorganisir dengan definisi yang tepat dan sesuai dengan sistem sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan data *storage*. Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari sistem informasi.

Selain digunakan untuk dokumentasi dan mengurangi redundansi, kamus data juga dapat digunakan untuk :

- a. Memvalidasi diagram arus data dalam hal kelengkapan dan keakuratan.
- b. Menyediakan suatu titik awal untuk menggambarkan layar dan laporan-laporan.
- c. Menentukan muatan data yang disimpan dalam file-file.
- d. Mengembangkan logika untuk proses-proses diagram arus data.

Pada tahap analisis sistem, kamus data digunakan sebagai alat komunikasi antara analisis sistem dengan *user*. Kamus data mendefinisikan elemen data dengan fungsi sebagai berikut :

1. Menjelaskan arti aliran data dan penyimpanan data dalam data *flow diagram*.
2. Mendeskripsikan komposisi paket yang bergerak melalui aliran data.
3. Mendeskripsikan komposisi penyimpanan data
4. Mendefinisikan nilai dan satuan yang relevan bagi penyimpanan dan aliran data
5. Mendeskripsikan hubungan detail antara penyimpanan data yang akan menjadi titik perhatian dalam *entity relationship diagram*.

Menurut Yakub (2012), kamus data berisi seharusnya mencerminkan keterangan yang jelas tentang data yang akan dicatat sehingga kamus data dapat menjelaskan hal-hal sebagai berikut :

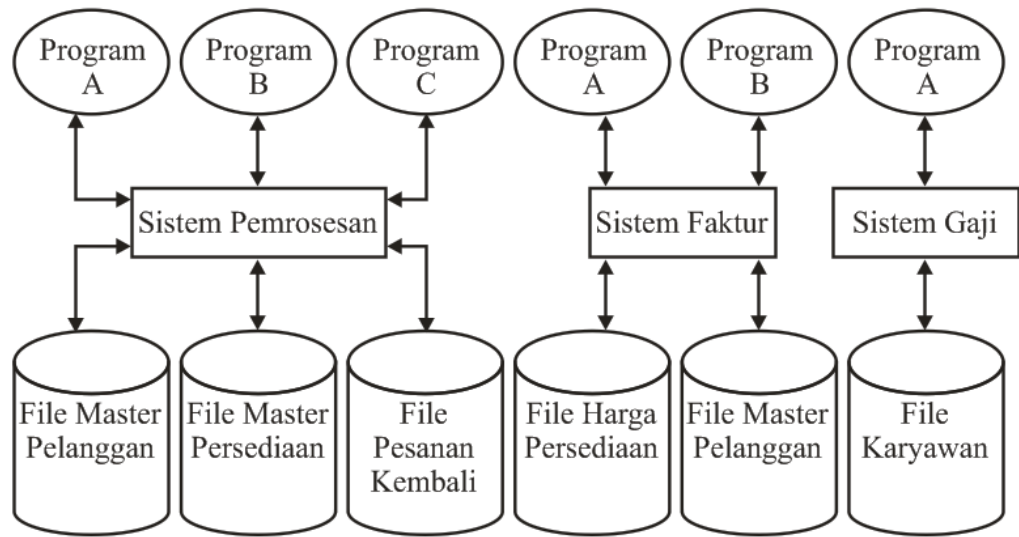
1. Nama arus data, dibuat berdasarkan arus data yang mengalir di *data flow diagram* dan memerlukan penjelasan tentang arus data dapat langsung mencarinya dengan mudah.

2. Struktur data, struktur data menunjukkan arus data yang dicatat di kamus data terdiri dari data item
3. Alias, nama lain dari data dapat dituliskan bila nama lain ada, misalnya, bagian pembuat faktur dan langganan menyebut bukti penjualan sebagai faktur, sedangkan bagian gudang menyebutnya sebagai tembusan permintaan persediaan.
4. Bentuk data, bentuk data perlu dicatat di kamus data, karena dapat digunakan untuk mengelompokkan kamus data ke dalam kegunaanya sewaktu perancangan sistem.
5. Arus Data, arus data menunjukkan dari mana data mengalir dan ke mana data akan menuju. Keterangan arus data ini perlu dicatat di *data flow diagram* agar mudah mencarinya.
6. Penjelasan, penjelasan dapat diisi dengan keterangan – keterangan tentang arus data, misalnya nama arus data adalah tembusan permintaan persediaan, maka dapat lebih dijelaskan sebagai tembusan dari faktur penjualan untuk meminta barang dari gudang
7. Periode, periode ini menunjukkan kapan terjadinya arus data ini dicatat di kamus data, kapan *input* data harus dimasukkan, kapan proses dari program harus dilakukan, dan kapan laporan harus dihasilkan
8. *Volume*, *volume* yang harus dicatat adalah tentang volume rata-rata dan *volume* puncak dari arus data.

F. Konsep Database

Nugroho (2010) bahwa database atau basis data merupakan suatu bentuk pengorganisasian data yang dilakukan sedemikian rupa dengan tujuan agar data tersebut menjadi lebih efektif dan efisien baik dari sisi penyimpanan maupun dari sisi pengaksesannya. Pada awal pengolahan data berbasis komputer, belum dikenal apa yang disebut database. Database merupakan hal yang sangat penting disebabkan komputer hanya digunakan secara eksklusif untuk perhitungan *scientific*. Secara perlahan, komputer memasuki dunia bisnis, dan agar dapat bermanfaat komputer harus dapat menyimpan, memanipulasi, serta memanggil berkas/*file* data yang berukuran sangat besar. Sistem pemrosesan berkas (*file*) berbasis komputer dikembangkan untuk kebutuhan tersebut.

Pada sistem pemrosesan berkas, data-data tersimpan secara terpisah pada masing-masing bagian yang ingin menggunakannya, dan data-data tersebut tidak saling berhubungan satu dengan yang lainnya. Akibatnya, masing-masing bagian memiliki data yang belum tentu sama dengan bagian yang lainnya, meskipun mereka memiliki jenis data yang sama. Perhatikan gambar berikut :



Gambar 2.1 Sistem pemrosesan berkas di perusahaan 'X'
 Sumber : Nugroho (2010:7)

Nugroho (2010:7) memberikan contoh sistem pemrosesan berkas pada perusahaan 'X'(gambar 2.1). dimana pendekatan sistem berfokus pada pemrosesan data yang diperlukan untuk setiap departemen secara individual. Setiap departemen diperlakukan seolah-olah mereka merupakan unit bisnis mandiri tanpa keterkaitan antara satu dengan yang lainnya. Setiap perogram dikembangkan untuk memenuhi kebutuhan sebagian departemen yang membutuhkannya.

Nugroho (2010:8) juga mengemukakan bahwa berdasarkan bagan sistem pemrosesan berkas diperusahaan 'X' (gambar 2.1), ada berkas misalnya (*file master* pelanggan) yang harus digandakan untuk memenuhi kebutuhan pemrosesan data di dua sistem yang berbeda, yaitu sistem pemesanan dan sistem faktur. Akibatnya, apabila ada perubahan di satu berkas, perubahan tersebut mungkin tidak tercermin pada berkas yang

sama yang digunakan pada sistem lainnya, sehingga data dan informasi yang sama tidak konsisten di satu berkas dan di berkas lainnya.

Database merupakan sebuah koleksi informasi yang terorganisir dengan cara-cara tertentu sehingga sebuah program komputer dapat dengan cepat memilih potongan data yang diinginkan. Ditambah juga bahwa kita dapat membayangkan sebuah database sebagai sebuah sistem *file elektronik*.

Sebuah database adalah sebuah koleksi informasi yang terorganisir sehingga dapat dengan mudah diakses, diatur, dan diperbaharui. Dalam pandangan tertentu, database dapat dibedakan menurut tipe isinya, yaitu *bibliographic, full text, numeric, dan image*.

1. Keunggulan Database

Pendekatan database memberikan tekanan pada integrasi dan berbagi data dalam organisasi. Nigroho (2010:20) mengemukakan bahwa pendekatan database/sistem basis data menawarkan keunggulan dibandingkan sistem pemrosesan berkas tradisional. Keunggulan tersebut adalah :

a. Kemandirian perogram dan data

Pemisahan deskripsi data dari perogram aplikasi yang menggunakan data tersebut dinamakan kemndirian data (*data independence*). Dengan pendekatan basis data, deskripsi data disimpan dilokasi terpusat yang dinamakan *repository*. Karakteristik sistem basis data ini mengijinkan organisasi data berubah (samapai

batas tertentu) tanpa mempengaruhi program aplikasi yang memproses data tersebut (tidak membutuhkan pemrograman ulang/*reprogramming*).

b. Mengurangi pengulangan data (*rerundansi*) yang tidak perlu

Sasaran perancangan dengan pendekatan basis data adalah menyatukan berkas-berkas data pada suatu struktur logika yang tunggal. Setiap fakta primer direkam hanya pada satu tempat di basis data.

c. Memperbaiki konsistensi data

Dengan mengendalikan redundansi, secara dramatis dapat mengurangi kesempatan terjadinya ketidak konsistenan data.

d. Memperbaiki kesempatan berbagi data (*data sharing*)

Basis data dirancang untuk berbagai sumber daya data dalam organisasi. Pengguna dengan hak tertentu dapat mengakses bagian tertentu dalam basis data, dimanapun pengguna tersebut berada dalam organisasi.

e. Merambah produktivitas pengembangan program aplikasi

Salah satu keunggulan pendekatan basis data adalah pengurangan waktu dan biaya untuk mengembangkan aplikasi bisnis baru. Ada dua alasan yang memungkinkan aplikasi basis data dapat dikembangkan dengan cara yang lebih cepat dibandingkan aplikasi yang mengakses berkas data pada pendekatan tradisional, yaitu :

1. Dengan mengasumsikan bahwa bisnis data telah dirancang dan diimplementasikan, pemrogram cukup berkonsentrasi pada fungsi spesifik yang dibutuhkan oleh aplikasi baru tanpa perlu mengasumsikan perancangan berkas data atau rincian implementasi aras rendah.
 2. Sistem manajemen basis data menjadikan sejumlah *tool* seperti pembuatan *form* atau genertor laporan dan bahasa tingkat tinggi dapat mengotomatisasi beberapa aktivitas perancangan basis data dan implementasinya.
- f. Memperbaiki akses data

Dengan basis data bertipe rasional, pengguna tanpa pengetahuan dan pengalaman dan pemrograman dapat memanggil dan menampilkan data-data tertentu sesuai dengan hak yang dimilikinya, meskipun cara mendapatkannya mungkin melewati batas departemen dimana prnggua itu berada.

- g. Mengurangi biaya pemeliharaan program

Data yang tersimpan harus diubah karena beberapa alasan yaitu tipe data baru ditambahkan, format data berubah, dan sebagainya. Pada sistem pemrosesan berkas, deskripsi data dan logika untuk mengakses data bersatu dengan perogram aplikasi. Akibatnya, perubahan format data dan metode akses akan mengakibatkan pemrogram harus merubah program aplikasinya. Pada sistem basis data, data bersifat mandiri dari program aplikasi yang

menggunakannya, sehingga sampai batas-batas tertentu format data maupun aplikasi data dapat diubah secara mandiri tanpa harus mengubah yang lainnya. Akhirnya, pemeliharaan program secara signifikan akan berjalan lebih mudah pada lingkungan basis data modern.

2. Tahapan Perancangan Database

Dalam proses perancangan database ada dua tahapan yang harus dilalui, yaitu perancangan database secara logika dan perancangan database secara fisik. Nugroho (2010:38) mengemukakan bahwa pada tahap perancangan database secara logika, pertama-tama yang dilakukan adalah menerjemahkan model data konseptual ke bentuk relasi-relasi berdasarkan teori tentang database rasional dan teori tentang objek atau kelas.

Setiap program komputer dalam sistem informasi dirancang termasuk didalamnya bagaimana format *input* dan *output* yang diinginkan. Kemudian, dilakukan peninjauan ulang terhadap transaksi-transaksi, laporan, tampilan di layar monitor dan lain-lainnya yang didukung oleh database.

Selama tahap ini juga dilakukan verifikasi data apa yang sebenarnya harus disimpan dalam database. Langkah terakhir pada perancangan database secara logika adalah mentransformasikan spesifikasi data ke bentuk-bentuk elemenelemen yang mengikuti aturan-aturan untuk spesifikasi data yang terstruktur dengan baik. Untuk kebanyakan database

saat ini, aturan-aturan datang dari teori database relasional dan prosesnya disebut normalisasi. Hasil dari perancangan database secara logika adalah gambaran lengkap database tanpa terpengaruh pada tipe *database manajement system* (DBMS) yang akan digunakan untuk mengelolah data-data yang bersangkutan.

Lebih lanjut Nugroho (2010:40) juga mengemukakan bahwa pada perancangan database secara fisik, yang dilakukan adalah memutuskan bagaimana organisasi database di tempat penyimpanan komputer, serta mendefenisikan struktur fisik dari DBMS. Dalam tahapan ini juga, mulai dilakukan penulisan program-program untuk memproses transaksi-transaksi dan laporan-laporan. Sasarannya adalah merancang database yang secara efisien dan aman menangani pemrosesan data.

a) Database Relasional

Konsep perancangan database yang banyak digunakan saat ini adalah konsep model data relasional. Model data relasional pertama kali diperkenalkan pada tahun 1970 oleh E.F. Codd, kemudian dilanjutkan oleh IBM. Model data relational menggambarkan data dalam bentuk tabel-tabel. Model data relasional berbasis pada teori matematika, yaitu kalkulus relasional. Bunafit Nugroho (2010:46) mengemukakan bahwa komponen inti pada model data relational, yaitu :

1) Struktur data

Data-data yang diorganisasikan dalam bentuk tabel dengan baris dan kolom.

2) Manipulasi data

Operasi yang sangat berguna digunakan untuk memanipulasi data yang disimpan dalam relasi-relasi.

3) Integrasi data

Fasilitas-fasilitas untuk mengspesifikasi aturan bisnis yang memelihara integritas data saat mereka dimanipulasi.

Relasi adalah tabel data dua dimensi. Setiap tabel memuat himpunan kolom dan himpunan baris. Sebuah atribut/*field*, konsisten dengan definisinya digunakan sebagai nama kolom relasi. Setiap baris relasi berhubungan dengan satu *record* untuk satu entitas tunggal. Setiap relasi harus memiliki kunci primer (*primary key*) yang dapat digunakan sebagai basis pencarian. Kunci primer adalah atribut ataupun kombinasi atribut yang secara unik mengidentifikasi setiap baris pada relasi. Apabila ada kunci primer yang memiliki lebih dari satu atribut maka disebut sebagai kunci komposit (*composite key*). Apabila kunci primer sebuah tabel berada pada tabel lain maka disebut sebagai kunci tamu (*foreign key*).

Perancangan database relasional bertujuan untuk menghasilkan database dengan struktur yang dikatakan normal. Pencapaian struktur normal tersebut dilakukan melalui tahapan normalisasi.

Nugroho (2010:50) memaparkan prinsip yang mendasari normalisasi, yaitu prinsip ketergantungan fungsional (*functional dependency*) serta prinsip dekomposisi tanpa kehilangan informasi yang dibutuhkan. Normalisasi dapat dipahami sebagai tahapan yang masing-masing berujung dengan bentuk normal. Bentuk normal adalah keadaan relasi yang dihasilkan dengan menerapkan aturan sederhana berkaitan dengan ketergantungan rasional yang bergantung pada relasi yang bersangkutan.

b) Normalisasi

Nugroho (2010:48) mengemukakan bahwa ada beberapa bentuk normal dalam perancangan database yaitu:

1) Bentuk normal pertama (1NF/*first normal form*)

Bentuk normal pertama adalah suatu bentuk relasi dimana atribut bernilai banyak telah dihilangkan sehingga diperoleh nilai tunggal pada perpotongan setiap baris dan kolom pada tabel.

2) Bentuk normal kedua (2NF/*second normal form*)

Semua ketergantungan fungsional yang bersifat sebagian telah dihilangkan. Suatu relasi berada dalam bentuk normal kedua jika dia berada dalam bentuk normal pertama dan setiap atribut bukan kunci bergantung penuh terhadap kunci primer. Relasi yang berada dalam bentuk normal pertama akan menjadi bentuk normal kedua jika salah satu kondisi dibawah ini terpenuhi.

(a) Kunci primer hanya mengandung satu atribut.

(b) Tidak ada atribut bukan kunci hadir pada relasi sehingga semua atribut pada relasi adalah komponen dari kunci primer.

(c) Setiap atribut bukan kunci bergantung penuh pada sekumpulan atribut yang berfungsi sebagai kunci primer.

3) Bentuk normal ketiga (3NF/*third normal form*)

Semua bentuk ketergantungan transitif telah dihilangkan.

Ketergantungan transitif adalah ketergantungan fungsional antar dua atau lebih *field* bukan kunci.

4) *Boyce-cood normal form* (BCNF)

Semua anomali yang tersisa dari hasil penyempurnaan ketergantungan fungsional telah dihilangkan.

5) Bentuk normal keempat (4NF/*fourth normal form*)

Semua ketergantungan bernilai banyak telah hilang.

6) Bentuk normal kelima (5NF/*fifth normal form*)

Tujuan tahapan normalisasi adalah untuk menghasilkan tabel yang terstruktur dengan baik. Tabel yang terstruktur dengan baik adalah tabel yang mengandung jumlah redundansi (duplikat data yang tidak perlu) dalam jumlah minimal dan menyajikan pengguna untuk menyisipkan, mengubah, serta menghapus baris-baris tabel tanpa menimbulkan kesalahan atau inkosistensi data.

Nugroho (2010:50) menyebutkan bahwa tidak ada patokan dasar sejauh mana normalisasi harus dilakukan. Pada dasarnya normalisasi harus tetap dilakukan hingga tidak ditemukan lagi

anomali-anomali dalam tabel. Lebih lanjut Nugroho (2010:53) juga menyebutkan bahwa pada kenyataannya, seringkali normalisasi hingga bentuk normal ketiga sudah cukup memadai untuk menghilangkan anomali-anomali.

7) Relasi/hubungan antar tabel

Dalam model database relasional dikenal beberapa hubungan antar tabel yang ada dalam database, yaitu :

(a) Hubungan satu ke satu

Merupakan bentuk relasi / hubungan dimana satu *record* pada sebuah tabel hanya berelasi/berhubungan dengan satu *record* pada tabel lain.

(b) Hubungan satu ke banyak

Merupakan bentuk relasi dimana satu *record* pada sebuah tabel berelasi dengan banyak *record* pada tabel lainnya.

(c) Hubungan banyak ke banyak

Merupakan bentuk relasi dimana banyak *record* pada sebuah tabel berelasi dengan banyak *record* pada tabel lainnya.

G. MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia.

MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. (Wikipedia, 2017).

MySQL termasuk jenis RDBMS (*Relational Database Management System*). Itulah sebabnya istilah seperti tabel, baris dan kolom mengandung satu atau sejumlah tabel. Tabel terdiri dari sejumlah baris dan setiap baris mengandung satu atau beberapa kolom. Keunggulan MySQL yaitu :

1. Mampu menangani jutaan *user* dalam waktu yang bersamaan
2. Mampu menampung lebih dari 50.000.000 *record*

Sangat cepat mengeksekusi perintah.

H. Software Yang digunakan

1. Borland Delphi 7

Menurut Abdul Kadir (2004) “Delphi merupakan sebuah piranti pengembangan aplikasi berbasis Windows yang dikeluarkan oleh Borland Internasional”. Borland Delphi dirilis pertama kali pada bulan Februari

1995, dan sekarang sudah mencapai perkembangannya sampai dengan Delphi 7. Delphi merupakan bahasa pemrograman yang mudah, karena Delphi adalah bahasa perograman tingkat tinggi (*high level*) sehingga sangat memudahkan user untuk bermain-main di tingkat ini.

Pemrograman Delphi sangatlah mudah, hanya tinggal click and drag, dan jadilah program aplikasi yang diinginkan.

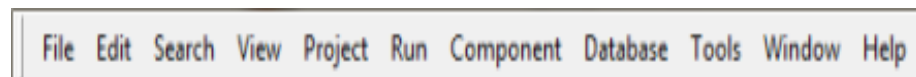
Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa pemrograman ini terletak pada Kualitas, Produktivitas, Kwantitas, Pengembangan perangkat lunak, Desain, Kecepatan eksekusi program.

Khusus untuk pemrograman *database*, Borland Delphi menyediakan fasilitas objek yang kuat dan lengkap yang memudahkan programmer dalam membuat program. Format *database* yang dimiliki Delphi adalah format *database* Paradox, dBase, Ms. Access, ODBC, SyBASE, *MySql*, Oracle dan lain-lain.

Adapun komponen Delphi 7 dibawah ini :

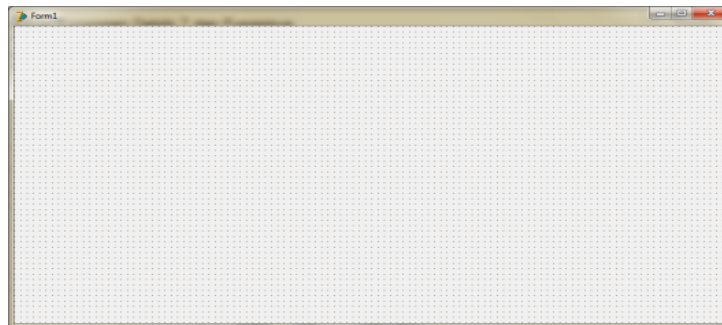
a. *MenuBar*

MenuBar berfungsi memilih tugas-tugas tertentu seperti: memulai,memilih, dan mengakhiri suatu aplikasi.



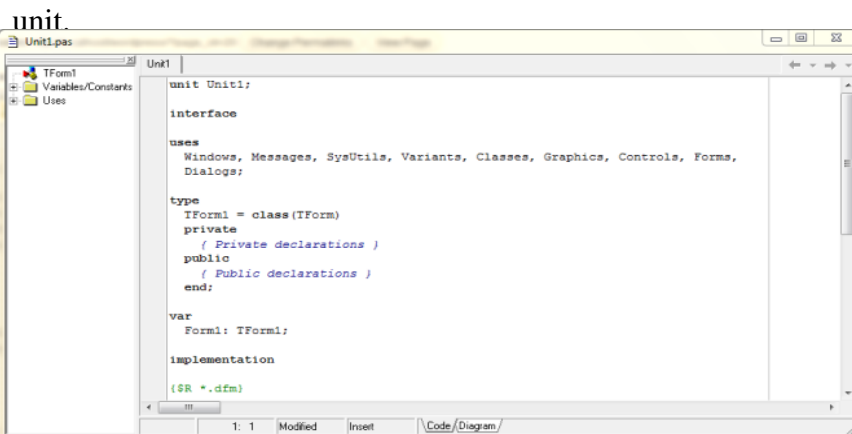
d. Jendela *Form*

Jendela ini digunakan untuk merancang dan mendesain user interface. Di sini pula kontrol-kontrol komponen dari komponen palet diletakkan.



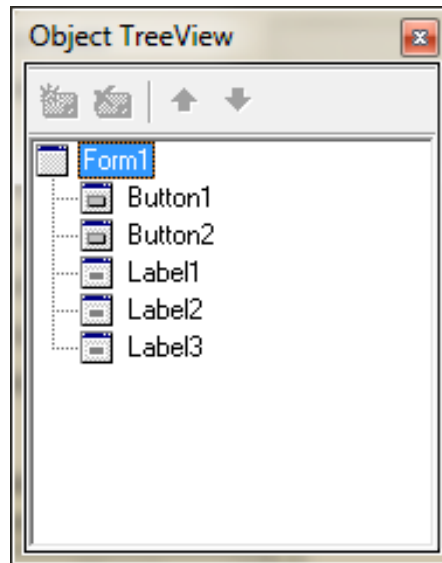
e. Jendela Unit

Jendela ini digunakan secara umum untuk menuliskan listing program dalam suatu aplikasi. Di dalamnya terdapat sebuah struktur



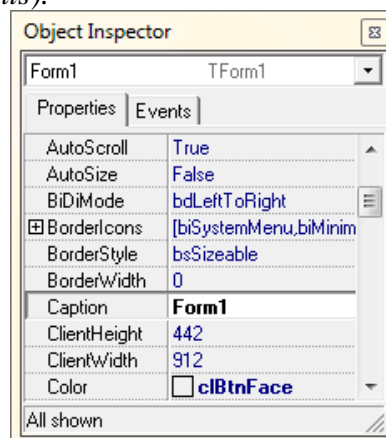
f. Jendela *Object Tree View*

Jendela ini digunakan untuk melihat kontrol-kontrol apa saja yang berada di dalam form tersebut secara hierarki seperti Windows Explorer.



g. Jendela *Object Inspector*

Jendela ini digunakan untuk memanipulasi kontrol-kontrol yang sudah ada dalam form. Jendela *Object Inspector* ini memiliki dua halaman, yang masing-masing disebut Properti (*Properties*) dan kejadian (*Events*).



2. MySQL

Menurut Andri Kristanto (2010) *MySQL* adalah sebuah perangkat lunak sistem manajemen basis data *SQL* atau *DBMS* yang *multithread* dan *multi_user*. *MySQL* dibangun, didistribusikan dan didukung oleh *MySQL AB*. *MySQL AB* merupakan perusahaan komersial yang dibiayai oleh pengembang *MySQL*. *MySQL* dapat didefinisikan sebagai sistem manajemen *database* sendiri yang merupakan struktur penyimpanan data.

Untuk menambah, mengakses dan memproses data yang disimpan dalam sebuah *database* komputer, diperlukan sistem manajemen *database* seperti *MySQL Server*. Selain itu *MySQL* dapat dikatakan sebagai basis data terhubung (*RDBMS*). *Database* terhubung menyimpan data pada tabel-tabel terpisah.

Server database MySQL mempunyai kecepatan akses tinggi mudah digunakan dan handal. *MySQL* dikembangkan untuk menangani *database* yang besar secara cepat dan telah sukses digunakan selama bertahun-tahun sehingga membuat *server MySQL* cocok untuk mengakses *database* di *internet*. *MySQL* juga merupakan sistem *client-server* yang terdiri atas *multithread SQL Server* yang mendukung *software client* dan *library* yang berbeda.

SQL (Structure Query Language) adalah bahasa standar yang digunakan untuk mengakses *serverdatabase*. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi dan digunakan sebagai

standar industri. Dengan menggunakan *SQL*, proses akses *database* menjadi lebih *user-friendly*.

Dalam konteks bahasa *SQL*, pada umumnya informasi tersimpan dalam tabel-tabel yang secara logik merupakan dua dimensi yang terdiri dari dua baris data (*row dan record*) yang berada dalam satu atau lebih kolom (*coloumn*). Baris data tabel sering disebut sebagai *instanisce* dari data sedangkan kolom sering disebut sebagai *atributes* atau *field*. Keseluruhan tabel itu dihimpun dalam satu kesatuan yang disebut *database*.

Menurut wahana komputer (2010), kelebihan dan keuntungan memakai *SQL* adalah :

- a. Banyak ahli berpendapat *MySQL* merupakan *server* tercepat
- b. *MySQL* merupakan sistem manajemen *database* yang *Opensource* (kode sumbernya terbuka), yaitu software ini bersifat free atau bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.
- c. *MySQL* mempunyai performa yang tinggi tapi simpel
- d. *Database MySQL* mengerti bahasa *SQL*
- e. *MySQL* dapat di akses melalui protocol *ODBC* (*Open Database Connectivity*) buatan microsoft. Ini menyebabkan *MySQL* dapat diakses oleh banyak software.
- f. Semua klien dapat mengakses *server* dalam satu waktu, tanpa harus menunggu yang lain untuk mengakses *database*

- g. *Database MySQL* dapat diakses dari semua tempat di internet dengan hak akses tertentu
- h. *MySQL* merupakan *database* yang mampu menyimpan data berkapasitas besar, sampai berukuran *Gigabyte*
- i. *MySQL* dapat berjalan di berbagai *operating* sistem seperti Linux, Windows, Solaris, dan lain-lain.

3. Koneksi Port

Langkah – langkah untuk mengkoneksikan port timbangan digital ke *system* :

1. Driver timbangan digital diinstal ke komputer sistem
2. Pada bahasa pemrograman Delphi, diinstal komponen Cport yang berfungsi membaca port computer system,
3. kemudian dibuat pengaturan pada aplikasi untuk dapat mengatur koneksi port timbangan dengan port computer system. Sehingga user dapat mengubah koneksi port timbangan sesuai dengan tempat USB yang diinginkan.

I. Metode Pengujian

1. White Box Testing

Ujicoba *Whitebox* merupakan metode desain uji kasus yang menggunakan struktur kontrol dari desain prosedural untuk menghasilkan kasus-kasus uji. Dengan menggunakan metode ujicoba *whitebox*, para pengembang *software* dapat menghasilkan kasus-kasus uji yang :

1. Menjamin bahwa seluruh independent paths dalam modul telah dilakukan sedikitnya satu kali,
2. Melakukan seluruh keputusan logikal baik dari sisi benar maupun salah,
3. Melakukan seluruh perulangan sesuai batasannya dan dalam batasan operasionalnya
4. Menguji struktur data internal untuk memastikan validitasnya

Mengapa menghabiskan banyak waktu dan usaha dengan menguji logikal *software* ? Hal ini dikarenakan sifat kerusakan alami dari *software* itu sendiri, yaitu :

5. Kesalahan logika dan kesalahan asumsi secara proposional terbalik dengan kemungkinan bahwa alur program akan dieksekusi. Kesalahan akan selalu ada ketika mendesain dan implementasi fungsi, kondisi atau kontrol yang keluar dari alur utama. Setiap harinya pemrosesan selalu berjalan dengan baik dan dimengerti sampai bertemu "kasus spesial" yang akan mengarahkannya kepada kehancuran.
6. Sering percaya bahwa alur logikal tidak akan dieksekusi ketika diketemukannya, mungkin akan dieksekusi dengan basis regular. Alur logika program biasanya berkebalikan dari intuisi, yaitu tanpa disadari asumsi mengenai alur kontrol dan data dapat mengarahkan pada kesalahan desain yang tidak dapat terlihat hanya dengan satu kali ujicoba.

7. Kesalahan *typographical* (cetakan) bersifat random. Ketika program diterjemahkan kedalam kode sumber bahasa pemrograman, maka akan terjadi kesalahan pengetikan. Banyak yang terdeteksi dengan mekanisme pemeriksaan sintaks, tetapi banyak juga yang tidak terdeteksi sampai dengan dimulainya ujicoba.

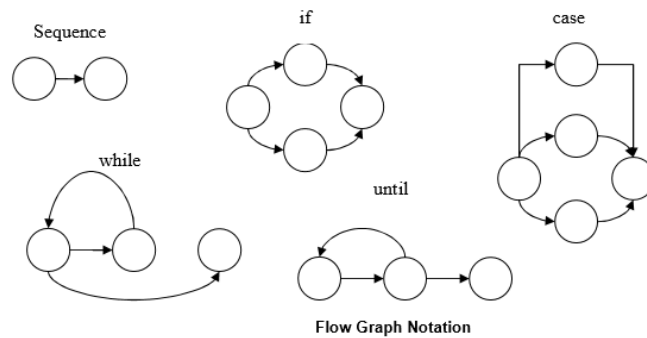
Karena alasan tersebut diatas, maka ujicoba *whitebox testing* diperlukan selain *blackbox testing*.

2. Ujicoba Berbasis Alur (*Basis Path Testing*)

Ujicoba berbasis alur merupakan teknik ujicoba *whitebox* pertama yang diusulkan oleh Tom McCabe. Metode berbasis alur memungkinkan perancang kasus uji untuk menghasilkan ukuran kompleksitas logikal dari desain prosedural dan menggunakan ukuran ini untuk mendefinisikan himpunan basis dari alur eksekusi. Kasus uji dihasilkan untuk melakukan sekumpulan basis yang dijamin untuk mengeksekusi setiap perintah dalam program, sedikitnya satu kali selama ujicoba

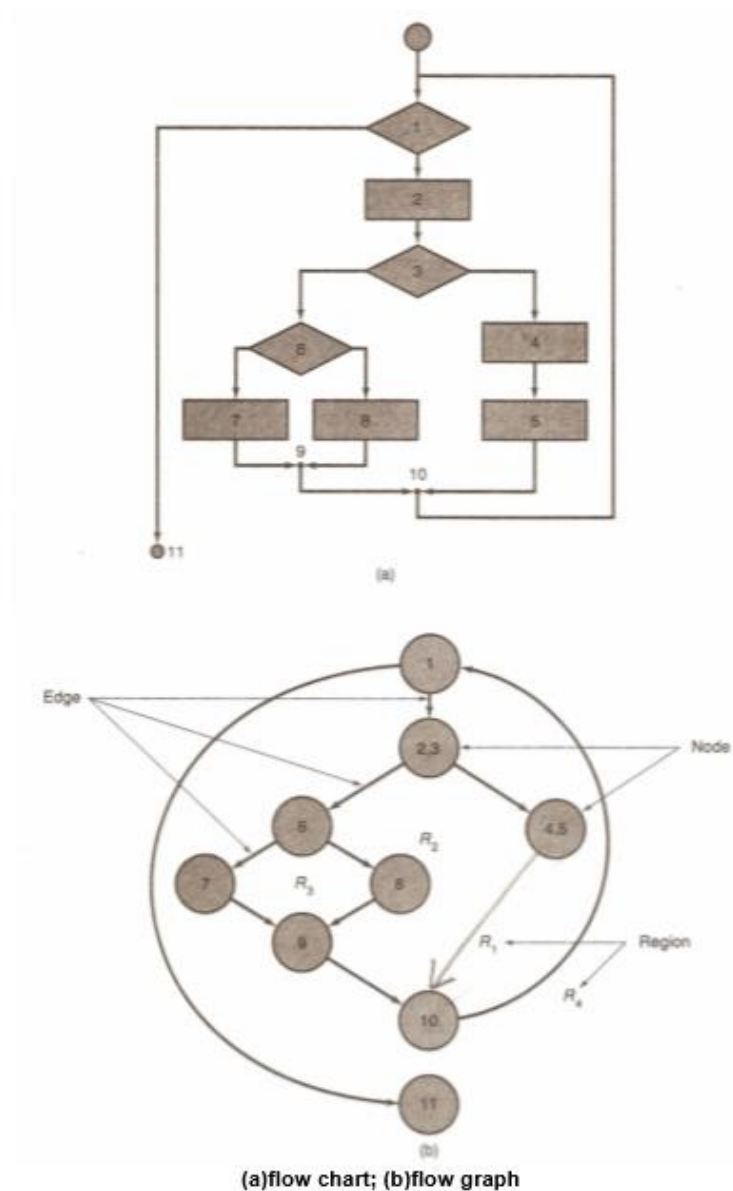
3. Notasi graf Alur (*Path Graph Notation*)

Notasi sederhana untuk merepresentasikan alur kontrol disebut graf alur (*flow graph*), seperti gambar dibawah ini :



Gambar 2.2 *flowgraph*

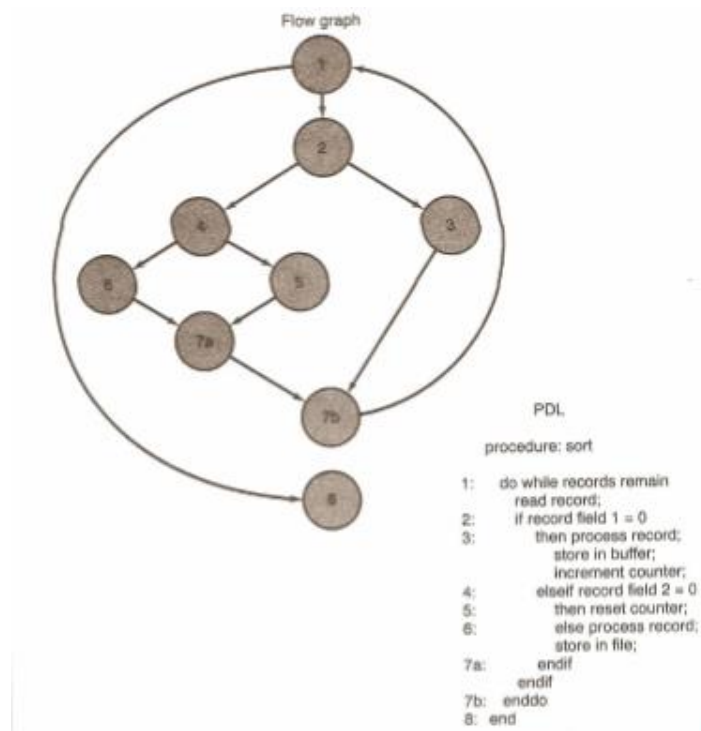
Untuk mengilustrasikan kegunaan dari diagram alir dapat dilihat pada gambar dibawah ini. Gambar bagian pertama digunakan untuk menggambarkan struktur kontrol program, sedangkan gambar bagian kedua setiap lingkaran disebut dengan *flow graph node*, merepresentasikan satu atau lebih perintah prosedural. Urutan dari simbol proses dan simbol keputusan dapat digambarkan menjadi sebuah *node*, sedangkan anak panah disebut *edges*, menggambarkan aliran dari kontrol sesuai dengan diagram alir. Sebuah *edge* harus berakhir pada sebuah *node* walaupun tidak semua *node* merepresentasikan perintah prosedural. Area yang dibatasi oleh *edge* dan *node* disebut *region*, area diluar *graph* juga dihitung sebagai *region*.



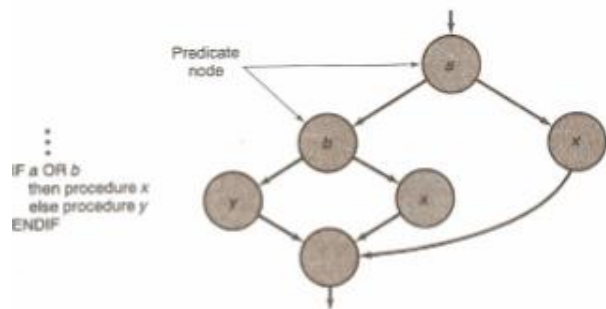
Gambar 2.3 flowchart dan flowgraph

Setiap representasi rancangan prosedural dapat diterjemahkan kedalam *flow graph*. Gambar (a) dibawah ini merupakan bagian dari PDL (*Program Design Language*) dan *flow graph*-nya (perhatikan nomor untuk setiap perintahnya) Ketika kondisi gabungan ditemukan, maka penggambaran *flow graph* akan menjadi lebih rumit. Kondisi gabungan

biasanya muncul jika satu atau lebih operator *Boolean* (*OR*, *AND*, *NAND*, *NOR*) ditemukan dalam perintah, seperti terlihat pada gambar (b) dibawah ini :



(a) Translating PDL to flow graph



(b) Translating PDL with compound conditions to flow graph

Gambar 2.4 Translating

4. *Cyclomatic Complexity*

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.

Independent path adalah alur manapun dalam program yang memperkenalkan sedikitnya satu kumpulan perintah pemrosesan atau kondisi baru. Contoh *independent path* dari gambar *flow graph* diatas :

Path 1 : 1 – 11

Path 2 : 1 – 2 – 3 – 4 – 5 – 10 – 1 – 11

Path 3 : 1 – 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path 4 : 1 – 2 – 3 – 6 – 7 – 9 – 10 – 1 – 11

Misalkan setiap *path* yang baru memunculkan *edge* yang baru, dengan *path*: 1 - 2 – 3 – 4 – 5 – 10 - 1 - 2 – 3 – 6 – 8 – 9 – 10 – 1 – 11

Path diatas tidak dianggap sebagai *independent path* karena kombinasi *path* diatas telah didefinisikan sebelumnya Ketika ditetapkan dalam *graf* alur, maka *independent path* harus bergerak sedikitnya 1 *edge* yang belum pernah dilewati sebelumnya. Kompleksitas *cyclomatic* dapat dicari dengan salah satu dari 3 cara berikut :

8. Jumlah region dari graf alur mengacu kepada kompleksitas *cyclomatic*

9. Kompleksitas *cyclomatic* untuk graf alur G didefinisikan :

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

Dimana E = jumlah *edge*, dan N = jumlah node

10. Kompleksitas *cyclomatic* untuk graf alur G didefinisikan :

$$V(G) = P + 1$$

Dimana P = jumlah *predicates nodes*

Berdasarkan *flow graph* gambar (b) diatas, maka kompleksitas *cyclomatic*-nya dapat di hitung sebagai berikut :

11. *Flow graph* diatas mempunyai 4 *region*

12. $V(G) = 11 \text{ edges} - 9 \text{ nodes} + 2 = 4$

13. $V(G) = 3 \text{ predicates nodes} + 1 = 4$

Hasil kompleksitas *cyclomatic* menggambarkan banyaknya path dan batas atas sejumlah ujicoba yang harus dirancang dan dieksekusi untuk seluruh perintah dalam program.

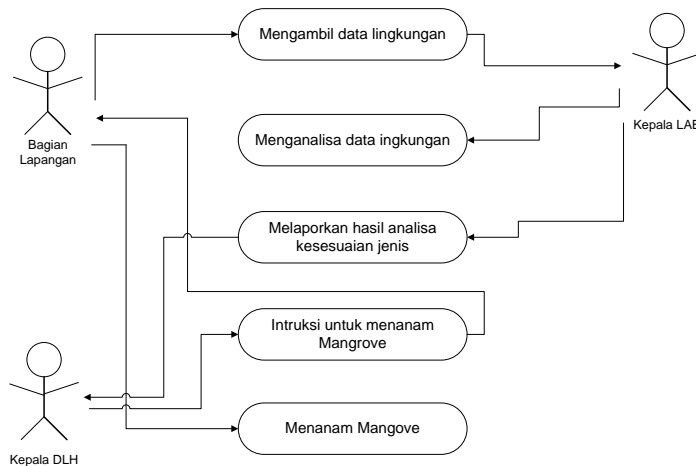
BAB III

METODE PENELITIAN

A. Penggambaran Sistem

1. Analisis sistem yang sedang berjalan

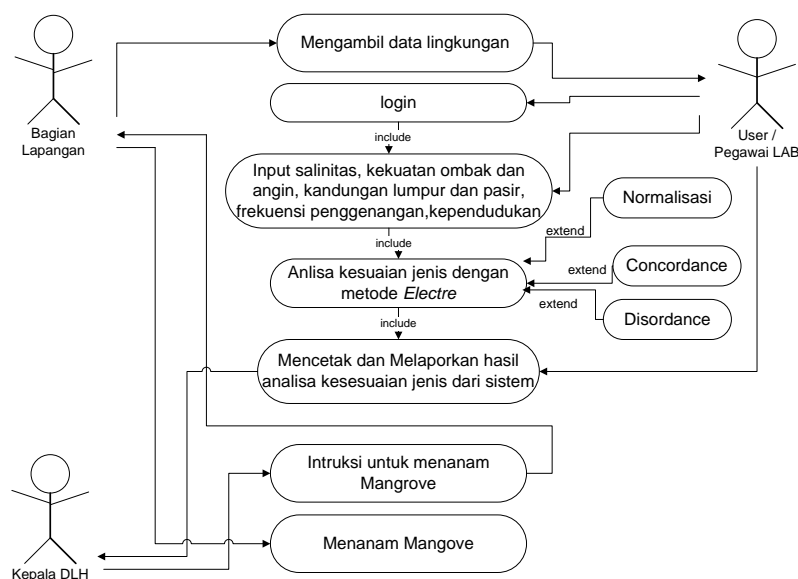
Tahapan yang di butuhkan dalam mengembangkan suatu program yaitu menganalisa sistem yang telah berjalan. Pada umumnya, bagian lapangan mengumpulkan data lingkungan yang mencakup salinitas (kadar garam), kekuatan ombak dan angin, kandungan lumpu, kandungan pasir serta frekuensi penggenangan. Kemudian diberikan kebagian laboratorium. Kemudian bagian laboratorium menganalisa data lingkungan untuk menentukan jenis *Mangrove* yang cocok untuk dilingkungan tersebut lalu melaporkan hasil analisa kepada Kepala Dinas Lingkungan Hidup Kota Makassar. Setelah itu Kepala Dinas Lingkungan Hidup Kota Makassar mengintruksikan bagian lapangan untuk menanam *Mangrove* yang sesuai degan lingkungan tempat pengambilan data.



Gambar 3.1 : Use Case Diagram Sistem yang sedang berjalan
Sumber : Hasil Olahan Penulis

2. Sistem yang diusulkan

Berdasarkan analisa sistem yang sedang berjalan, maka penulis memberikan suatu solusi pemecahan masalah dengan perancangan *Sistem Pendukung Keputusan untuk menganalisa kesesuaian jenis vegetasi Mangrove menggunakan Electre*. Bagian lapangan mengumpulkan data lingkungan. Kemudian diberikan kebagian user/pegawai LAB. Kemudian user menginput data lingkungan yang mencakup salinitas (kadar garam), kekuatan ombak dan angin, kandungan lumpu, kandungan pasir serta frekuensi penggenangan kedalam sistem. Maka sistem akan menentukan jenis *Mangrove* yang cocok untuk dilingkungan tersebut dengan algoritma *Electre* lalu dilaporkan kepada Kepala Dinas Lingkungan Hidup Kota Makassar. Setelah itu Kepala Dinas Lingkungan Hidup Kota Makassar mengintruksikan bagian lapangan untuk menanam *Mangrove* yang sesuai dengan lingkungan tersebut.



Gambar 3.2 : Use Case Diagram Sistem yang diusulkan

Sumber : Hasil Olahan Penulis

B. Analisis Kebutuhan

1. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang harus dipenuhi agar suatu sistem dapat berjalan atau dapat dikatakan kebutuhan tambahan yang memiliki *input, proses dan output*.

Kebutuhan fungsional yang harus ada dalam sistem yang akan dibangun adalah :

- a. Sistem harus dapat mempermudah bagian *laboratoruim* dalam menentukan jenis *Mangrove* yang sesuai dengan data lingkungan
- b. Sistem harus dapat menampung data jenis *Mangrove* dan data lingkungan yang ada di kota Makassar
- c. Sistem harus dapat mencetak hasil jenis vegetasi *Mangrove* yang sesuai dengan lingkungan yang dianalisa
- d. Aktor yang terlibat pada sistem ini :
 1. Bagian Lapangan, bertindak sebagai pengambil data di lingkungan yang kemudian diserahkan pada bagian *laboratorium*.
 2. Pegawai *laboratorium*, bertindak sebagai admin, bertugas menginput data yang akan dikelola oleh sistem kemudian memprosesnya.
 3. Kepala Dinas Lingkungan Hidup Kota Makassar, sebagai penerima laporan yang dicetak dari bagian *laboratorium*.

2. Kebutuhan Non Fungsional

a. *Hardware*

1. *Processor Core i3* atau lebih tinggi
2. RAM 2 GB atau yang lebih tinggi
3. Hardisk 500 GB atau di atasnya.
4. Printer.
5. Monitor SVGA atau dengan resolusi lebih tinggi.
6. *Mouse*
7. *Keyboard*

b. *Software*

1. SO *Windows 7* atau di atasnya
2. *Borland Delphi 7*
3. Database *MySQL*

c. Model yang digunakan yaitu UML

d. Pengguna sistem yaitu *User*

C. Sumber Data dan Teknik Pengumpulan Data

14. Sumber Data

a. Data Primer

Data primer merupakan data yang diperoleh peneliti secara langsung. Dalam hal ini melalui wawancara terhadap pegawai Dinas Lingkungan Hidup Kota Makassar.

b. Data Sekunder

Data sekunder adalah data yang diperoleh peneliti dari sumber yang telah ada.

15. Teknik Pengumpulan Data

c. *Observasi*

Agar sistem yang akan dibangun dapat diandalkan maka perlu untuk datang langsung ke lokasi penelitian untuk mendapatkan data struktur organisasi Dinas Lingkungan Hidup Kota Makassar sehingga bisa diketahui data-data yang diperlukan untuk membangun Sistem Pendukung Keputusan Menentukan Jenis Vegetasi *Mangrove*.

d. Wawancara

Wawancara perlu dilakukan untuk mengetahui kebutuhan apa saja yang dibutuhkan untuk menyelesaikan masalah yang telah dipaparkan pada rumusan masalah, juga untuk membuktikan bahwa sistem yang akan dibangun memang dibutuhkan di Dinas Lingkungan Hidup Kota Makassar.

e. Teknik Kepustakaan

Teknik ini digunakan dengan cara mempelajari literature berupa buku, artikel-artikel, dokumen atau arsip di buku-buku pedoman, buku-buku perpustakaan yang dianggap dapat mendukung proses pengumpulan data.

D. Waktu dan Lokasi Penelitian

Penelitian ini dilakukan mulai tanggal 27 Januari s/d 27 Februari pada Dinas Lingkungan Hidup Kota Makassar yang terletak di jalan Urip Sumoharjo No. 8 kompleks gabungan dinas Kota Makassar.

E. Gambaran Umum Lokasi Penelitian

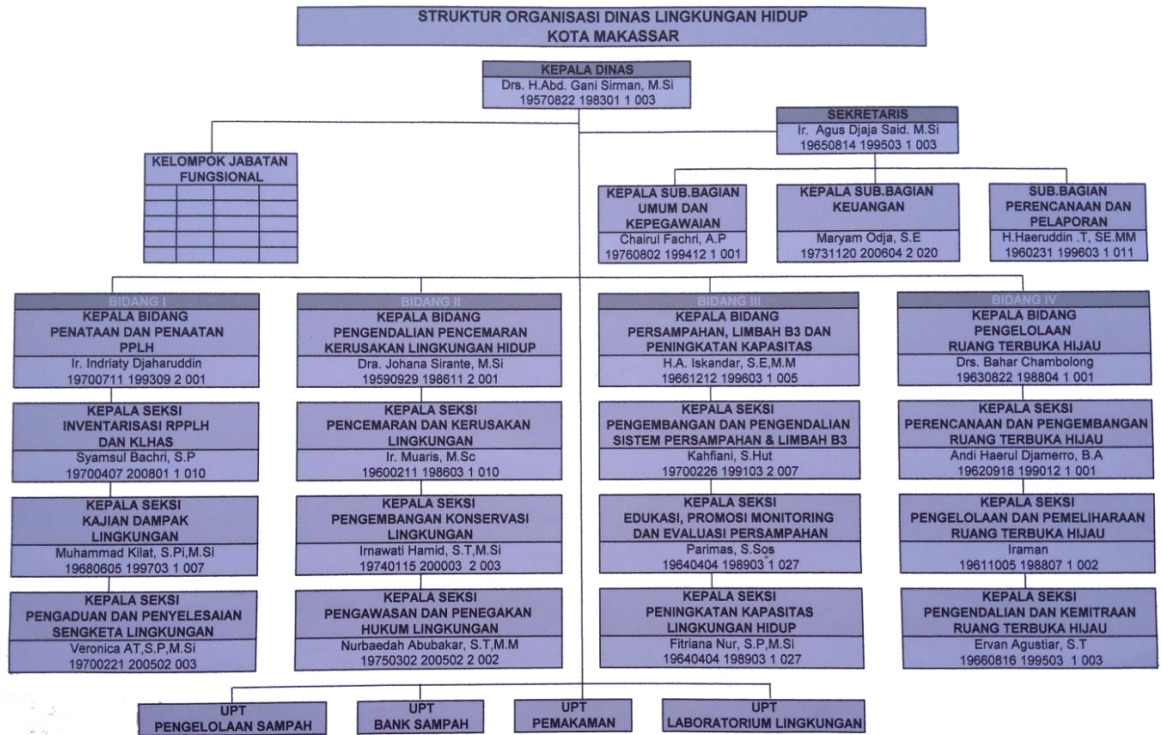
Dinas Lingkungan Hidup (DLH) kota Makassar, merupakan Organisasi yang membantu wali kota melaksanakan Urusan Pemerintahan bidang Lingkungan Hidup yang menjadi kewenangan daerah dan tugas pembentukan yang ditugaskan kepada Daerah.

Adapun uraian tugas DLH sebagai berikut :

1. Merumuskan Rencana Perlindungan dan Pengelolaan Lingkungan Hidup (RPPLH) kota Makassar.
2. Merumuskan Kajian Lingkungan Hidup Strategis (KLHS) kota Makassar.
3. Melakukan pencegahan, penanggulangan dan pemulihan pencemaran data/kerusakan lingkungan hidup dalam wilayah kota Makassar.
4. Mengelola keanekaragaman, hayati dalam wilayah kota Makassar.

5. Melakukan pengumpulan limbah bahan berbahaya dan beracun (B3) dalam wilayah Kota Makassar.
6. Melakukan penyimpanan sementara limbah bahan berbahaya dan beracun (B3).
7. Membina dan mengawasi terhadap usaha dan /atau kegiatan yang izin lingkungan dan izin perlindungan dan pengelolaan lingkungan hidup (PPLH) yang diterbitkan dalam wilayah kota Makassar.
8. Menyelenggarakan pendidikan, pelatihan dan penyuluhan lingkungan hidup untuk lembaga kemasyarakatan.
9. Memberikan penghargaan lingkungan hidup tingkat kota Makassar.
10. Menyelesaikan pengaduan masyarakat terhadap usaha dan / atau kegiatan yang izin lingkungannya diterbitkan dan berdampak dikota Makassar.
11. Mengelola persampaian, pengangkutan dan pemrosesan akhir.
12. Menerbitkan izin pendaurulangan sampah.
13. Membina dan mengawasi pengelolaan sampah yang diselenggarakan oleh pihak swasta.

Adapun Struktur Organisasi DLH sebagai berikut :



Sumber : Dinas Lingkungan Hidup Kota Makassar (2017)

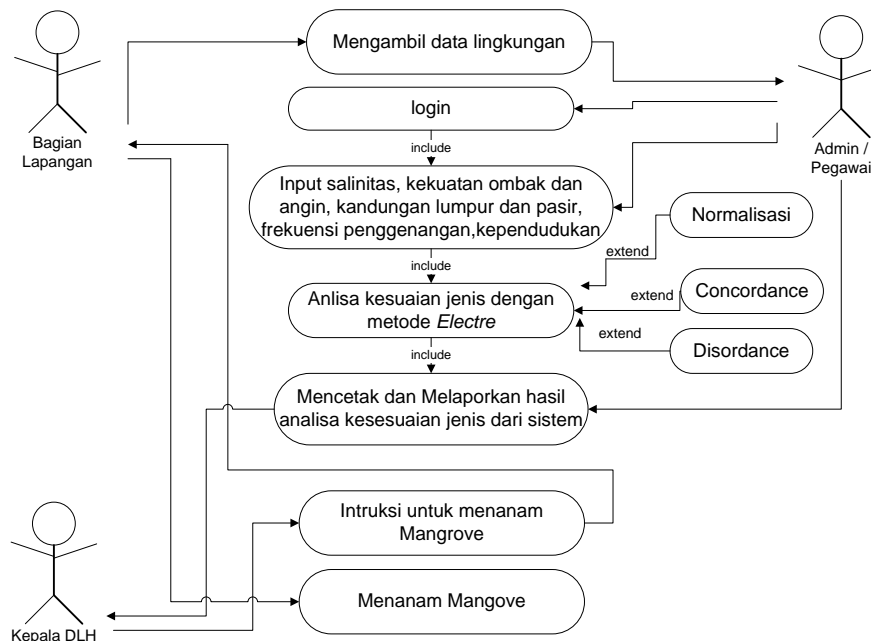
BAB IV

PERANCANGAN DAN IMPLEMENTASI SISTEM

A. Pemodelan Sistem

Pemodelan sistem adalah suatu kegiatan membuat desain teknis berdasarkan evaluasi yang telah dilakukan pada kegiatan analisis. Pada pemodelan / perancangan desain sistem, item-item dan komponen dirancang dengan tujuan untuk memudahkan dan mengefisienkan kinerja dan aktifitas kerja serta memberikan gambaran umum tentang bagaimana mekanisme yang tepat untuk mendesain suatu sistem sesuai dengan kebutuhan instansi, kelebihan dan keuntungan yang disajikan lewat suatu perancangan aplikasi.

16. Use Case Diagram



Gambar 4.1 Use Case Diagram

Pada *use case* terdapat tiga aktor yaitu admin, pegawai dan Masyarakat.

f. Bagian Lapangan

Actor : Bagian Lapangan

Brief Description : Bagian lapangan hanya mengambil data dari lingkungan kemudian diberikan kepada admin.

Main Flow : Bagian lapangan langsung memberikan data lingkungan berupa salinitas, kekuatan ombak dan angin, kandungan pasir, lumpur, frekuensi penggenangan dan kepadatan penduduk.

g. Admin

Actor : Admin

Brief Description : *Usecase login*, Mengelola Data Kecamatan, lingkungan, vegetasi, analisa kesesuaian dan melakukan *logout*.

Main Flow : Admin pertama kali melakukan melakukan *login* untuk mengelola data Kecamatan, lingkungan, vegetasi, analisa kesesuaian serta melakukan *logout*.

h. Kepala DLH

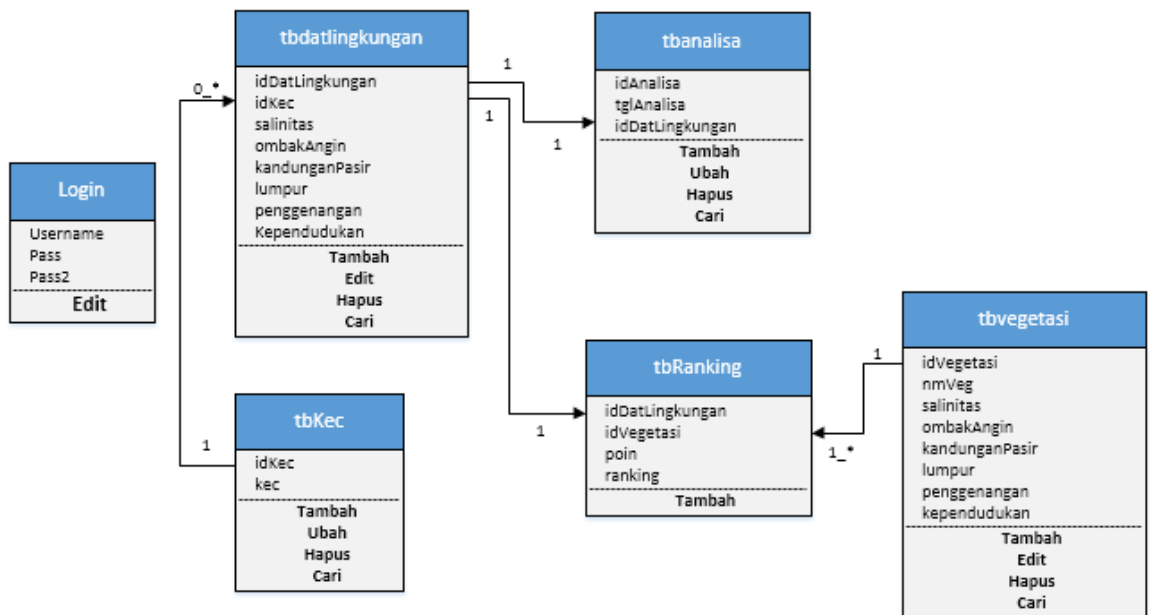
Actor : Kepala DLH

Brief Description : Menerima Laporan.

Main Flow : Kepala Dinas Lingkungan Hidup hanya menerima laporan analisa kesesuaian jenis yang dicetak oleh bagian Admin.

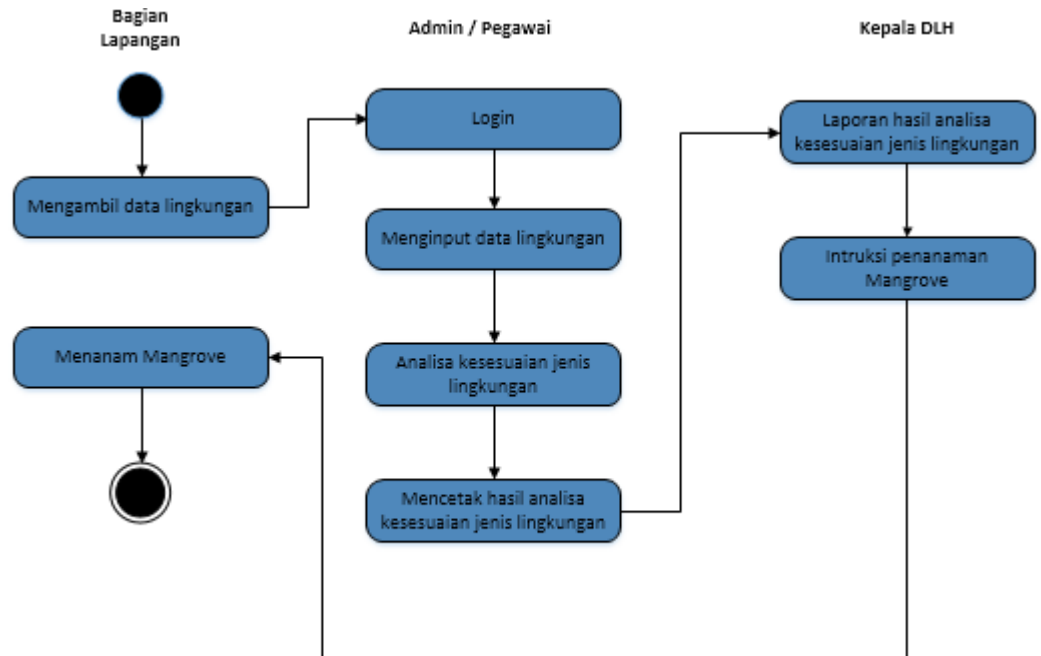
17. Class Diagram

Class diagram digunakan untuk menampilkan kelas – kelas dan paket – paket di dalam sistem. *Class diagram* memberikan gambaran sistem secara statis dan relasi antar mereka.



Gambar 4.2 Class Diagram

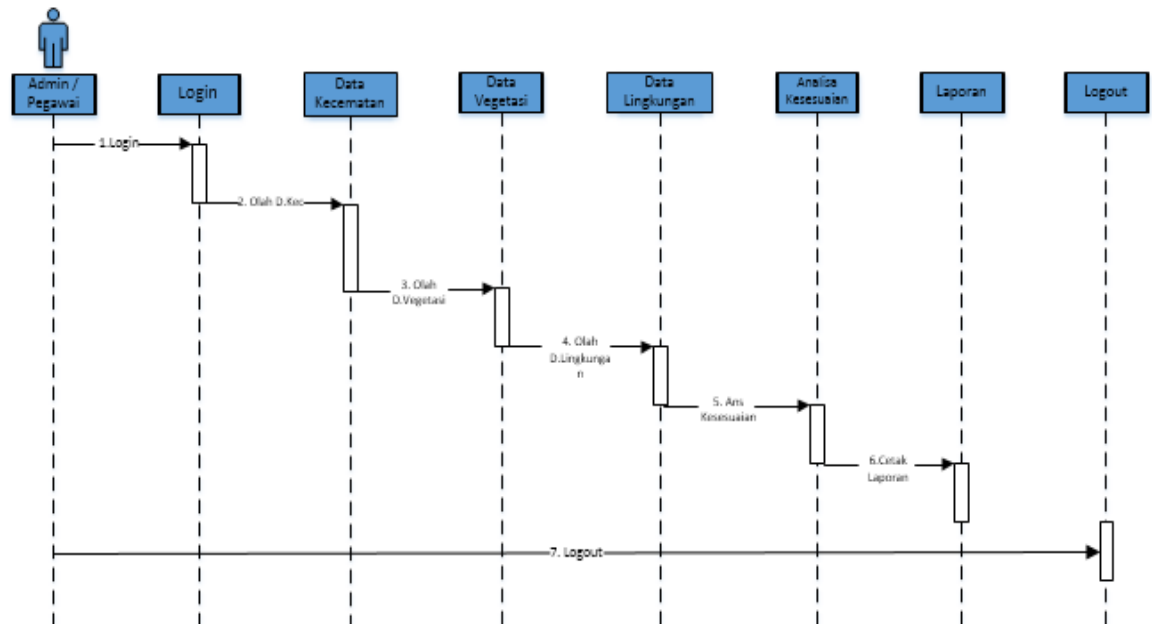
18. Activity Diagram



Gambar 4.3 Activity Diagram

Gambar 4.3 di atas adalah *activity* diagram pegawai, pertama-tama bagian lapangan mengambil data lingkungan kemudian diserahkan kepada bagian Admin, lalu Admin melakukan *login* terlebih dahulu kemudian menginput data lingkungan. Setelah itu menganalisa kesesuaian jenis lingkungan kemudian mencetak hasilnya, kemudian melaporkan kepada Kepala DLH. Kemudian kepala DLH memberi intuksi kepada bagian lapangan untuk menanam *Mangrove* yang sesuai dari analisa sistem.

19. Sequence Diagram



Gambar 4.4 *Sequence Diagram*

Gambar 4.4 diatas adalah *sequence diagram* Admin. Pertama-tama Admin membuka aplikasi dan login menggunakan *username* dan *password* yang terdaftar. Setelah berhasil login Admin memilih menu Kecamatan untuk mengolah data kecamatan, memilih menu Vegetasi untuk mengolah data Vegetasi, memilih menu Lingkungan untuk mengolah data lingkungan yang tersimpan didalam *database* dan admin mencetak laporan. Untuk menjalankan proses Analisa Kesesuaian Jenis, Admin menginput data lingkungan yang dijadikan kriteria dalam metode Electre. Kemudian Admin *logout* untuk keluar dari sistem.

B. Perancangan Database

Tabel 4.1 tbanalisa

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	idAnalisa	Int	11	<i>Auto_increment</i>
2	tglAnalisa	Date		
3	idDatLingkungan	Varchar	11	
4	ranking_satu	Varchar	30	

Tabel 4.2 tbdatingkungan

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	idDatLingkungan	Int	11	<i>Auto_increment</i>
2	idKec	varchar	11	
3	Salinitas	varchar	10	
4	ombakAngin	varchar	10	
5	kandunganPasir	varchar	10	
6	Lumpur	varchar	10	
7	Penggenangan	varchar	10	
8	Kependudukan	varchar	10	
9	salinitas2	varchar	20	
10	ombakAngin2	varchar	20	
11	kandunganPasir2	varchar	20	
12	lumpur2	varchar	20	
13	Penggenangan	varchar	20	
14	kependudukan2	varchar	20	

Tabel 4.3 tbkec

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	idKec	Int	11	<i>Auto_increment</i>
2	Kec	varchar	50	

Tabel 4.4 tbranking

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	idDatLingkungan	Int	11	
2	idVegetasi	varchar	11	
3	Poin	varchar	5	
4	Ranking	Int	5	

Tabel 4.5 tbvegetasi

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	idVegetasi	Int	11	<i>Auto_increment</i>
2	nmVeg	Varchar	50	
3	Salinitas	Int	10	
4	ombakAngin	Int	10	
5	kandunganPasir	Int	10	
6	Lumpur	Int	10	
7	Penggenangan	Int	10	
8	Kependudukan	Int	10	

Tabel 4.6 login

No	Nama <i>Field</i>	<i>Type</i>	Ukuran	Keterangan
1	Urut	Int	11	<i>Auto_increment</i>
2	Nik	Varchar	50	
3	Iduser	Varchar	50	
4	Pass	Varchar	50	
5	Password	Varchar	50	

C. Perancangan Antar Muka

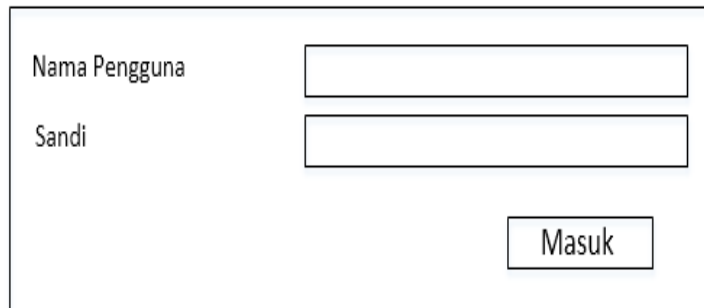
1. Rancangan *Output*

Gambar	PEMERINTAH KOTA MAKASSAR DINAS LINGKUNGAN HIDUP <small>Jalan Jendral Urip sumoharj No. 8 Tlp. (0411) 438381 Makassar 90144</small>	Gambar			
Salinitas	Toleransi Kekuatan Ombak dan Angin	Toleransi Kandungan Pasir	Toleransi Lumpur	Frekuensi Penggenangan	Kependudukan
xxx	xxx	xxx	xxx	xxx	x
Ranking Vegetasi Mangrove					
Ranking	Nama Mangrove	Poin			
1	xxx	x			
2	xxx	x			
3	xxx	x			
4	xxx	x			
5	xxx	x			
6	xxx	x			
7	xxx	x			
8	xxx	x			
9	xxx	x			
10	xxx	x			
Berdasarkan Hasil Analisa diatas, maka ditentukan Mangrove jenis xxx yang paling tepat untuk ditanam. Adapun alternatif selanjutnya yakni :					
Ranking	Nama Mangrove				
x	xxx				
x	xxx				
Makassar, xx-xx-xxxx Menyetujui, Kepala Dinas Lingkungan Hidup Kota Makassar					
xxxxxxxxxxxxxxxxxxxxxxxx NIP : xxxxxxxxxxxxxxxx					

Gambar 4.5 Rancangan *Output*

2. Rancangan *Input*

a. Rancangan *Input Login*

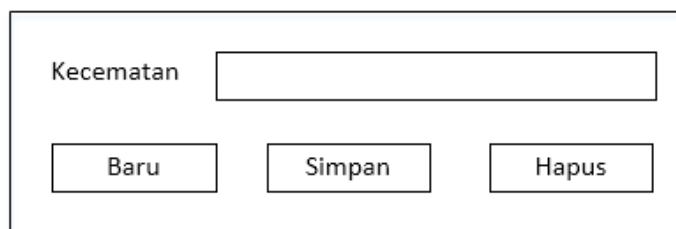


A login form with two input fields and one button. The first field is labeled 'Nama Pengguna' and the second is labeled 'Sandi'. A button labeled 'Masuk' is positioned at the bottom right.

Nama Pengguna	<input type="text"/>
Sandi	<input type="text"/>
<input type="button" value="Masuk"/>	

Gambar 4.6 Rancangan *Input Login*

b. Rancangan *Input Data Kecamatan*

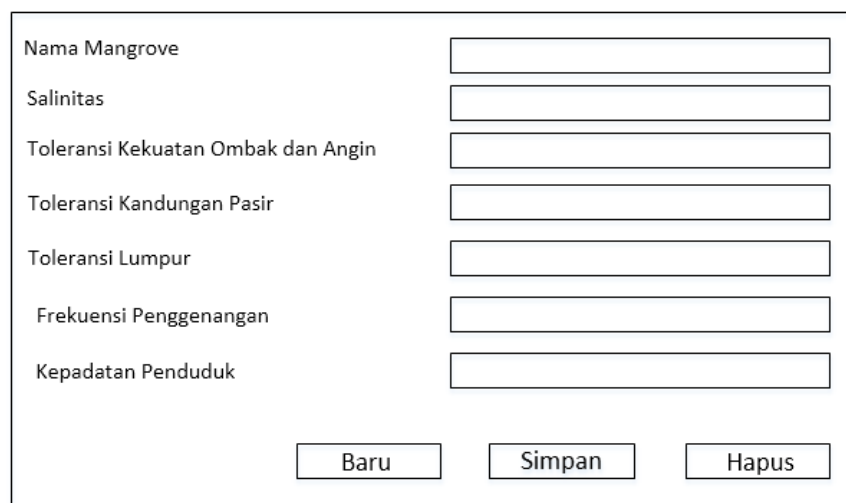


A form for entering 'Kecamatan' data. It features a single input field labeled 'Kecamatan' and three buttons: 'Baru', 'Simpan', and 'Hapus'.

Kecamatan	<input type="text"/>
<input type="button" value="Baru"/> <input type="button" value="Simpan"/> <input type="button" value="Hapus"/>	

Gambar 4.7 Rancangan *Input Data Kecamatan*

c. Rancangan *Input Data Vegetasi Mangrove*



A form for entering mangrove vegetation data. It contains seven input fields with labels: 'Nama Mangrove', 'Salinitas', 'Toleransi Kekuatan Ombak dan Angin', 'Toleransi Kandungan Pasir', 'Toleransi Lumpur', 'Frekuensi Penggenangan', and 'Kepadatan Penduduk'. At the bottom, there are three buttons: 'Baru', 'Simpan', and 'Hapus'.

Nama Mangrove	<input type="text"/>
Salinitas	<input type="text"/>
Toleransi Kekuatan Ombak dan Angin	<input type="text"/>
Toleransi Kandungan Pasir	<input type="text"/>
Toleransi Lumpur	<input type="text"/>
Frekuensi Penggenangan	<input type="text"/>
Kepadatan Penduduk	<input type="text"/>
<input type="button" value="Baru"/> <input type="button" value="Simpan"/> <input type="button" value="Hapus"/>	

Gambar 4.8 Rancangan *Input Data Vegetasi Mangrove*

d. Rancangan *Input Data Analisa Lingkungan*

Kecamatan	<input type="text"/>
Salinitas	<input type="text"/>
Kekuatan Ombak dan Angin	<input type="text"/>
Kandungan Pasir	<input type="text"/>
Kandungan Lumpur	<input type="text"/>
Frekuensi Penggenangan	<input type="text"/>
Padat Penduduk	<input type="text"/>
<input type="button" value="Baru"/> <input type="button" value="Simpan"/> <input type="button" value="Hapus"/>	

Gambar 4.9 Rancangan *Input Data Analisa Lingkungan*

e. Rancangan *Input Analisa Kesesuaian Jenis Lingkungan Mangrove*

Kecamatan	<input type="text"/>
Salinitas	<input type="text"/>
Toleransi Kekuatan Ombak dan Angin	<input type="text"/>
Toleransi Kandungan Pasir	<input type="text"/>
Toleransi Lumpur	<input type="text"/>
Frekuensi Penggenangan	<input type="text"/>
Kepadatan Penduduk	<input type="text"/>
<input type="button" value="Baru"/> <input type="button" value="Simpan"/> <input type="button" value="Hapus"/>	

Ga

Gambar 4.10 Rancangan *Input Analisa Kesesuaian*

f. Rancangan *Input Pengguna*

Kecamatan	<input type="text"/>
Salinitas	<input type="text"/>
Toleransi Kekuatan Ombak dan Angin	<input type="text"/>
<input type="button" value="Baru"/> <input type="button" value="Simpan"/> <input type="button" value="Hapus"/>	

Gambar 4.11 Rancangan *input pengguna*

D. Implementasi

1. Implementasi Algoritma

Algoritma adalah prosedur langkah-langkah untuk penghitungan, pemrosesan data, dan penalaran otomatis.

Electre merupakan salah satu metode pengambilan keputusan multikriteria berdasarkan pada konsep *outranking* dengan menggunakan perbandingan berpasangan dari alternatif-alternatif berdasarkan setiap kriteria yang sesuai. Electre digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi dan alternatif yang sesuai dapat dihasilkan. Dengan kata lain, Electre digunakan untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria dilibatkan.

Implementasi dengan contoh kasus jika data lingkungan sebagai berikut :

Table 4.7 Contoh data lingkungan

Salinitas	Kekuatan Ombak & angin	Kandungan Pasir	Kandungan Lumpur	Frekuensi Penggenangan	Kependudukan
0-10	Lemah	Sedang	Tinggi	0-9 Hari/Bulan	Ya

Sumber : Hasil olahan penulis

Konversi data lingkungan ke Nilai :

Table 4.8 Konversi data lingkungan

Salinitas	Kekuatan Ombak & angin	Kandungan Pasir	Kandungan Lumpur	Frekuensi Penggenangan	Kependudukan
1	2	3	4	2	2

Sumber : Hasil olahan penulis

Tahap 1. Normalisasi matriks keputusan

Tahap pertama dalam Algoritma Electre adalah tahap normalisasi, Dalam prosedur ini, setiap atribut diubah menjadi nilai yang *comparable* yang menghasilkan matrix R .

Listing

```
for h:=1 to totKolom do
begin for i:=1 to total do
begin
esqrt[i]:=0;
for j:=1 to total do
begin
esqrt[i]:=esqrt[i]+(StrToFloat
(kriteria.Cells[h,j])*
StrToFloat(kriteria.Cells[h,j]));
end;
Normalisasi.Cells[0,i]:=kriteria.Cells[0,i];
Normalisasi.Cells[h,i]:=FormatFloat('0.####',
StrToFloat(kriteria.Cells[h,i])/sqrt(esqrt[i]));
end;
end;
```

Menghasilkan

Normalisasi Matrix

nmVeg	salinitas	ombakAngin	kandunganPasir	lum
Api-api	0.3288	0.3841	0.2063	0.32
Nipah	0.1644	0.128	0.4126	0.41
Bayur Laut	0.3288	0.128	0.3094	0.24
Nyirih	0.3288	0.2561	0.3094	0.24
Tanjang Merah	0.3288	0.2561	0.2063	0.24

Tahap 2. Pembobotan pada matriks yang telah dinormalisasi

Setelah dinormalisasi, tahap kedua yang dilakukan yaitu setiap kolom dari matriks R dikalikan dengan bobot-bobot yang ditentukan oleh pembuat keputusan.

Listing

```
for h:=1 to totKolom do
begin
for i:=1 to total do
begin
Pembobotan.Cells[0,i]:=kriteria.Cells[0,i];
Pembobotan.Cells[h,i]:=FormatFloat('0.####',
StrToFloat(Normalisasi.Cells[h,i])*
StrToFloat(bobot.Cells[h,1]));
end;
end;
end;
```

Menghasilkan

Pembobotan pada Matrix yang telah dinormalisasi

nmVeg	salinitas	ombakAngin	kandunganPasir	lum
Api-api	0.3288	0.7682	0.6189	1.31
Nipah	0.1644	0.256	1.2378	1.64
Bayur Laut	0.3288	0.256	0.9282	0.98
Nyirih	0.3288	0.5122	0.9282	0.98
Tanjang Merah	0.3288	0.5122	0.6189	0.98

Tahap 3. Menentukan himpunan *concordance* dan *discordance*

Setelah tahap pembobotan, dilakukan tahap ketiga yakni kriteria dibagi menjadi dua himpunan bagian, yaitu *concordance* dan *discordance*.

Listing

```

for h := 1 to total do
begin
for i := 1 to total do
begin
for j := 1 to totKolom do
begin
if StrToFloat(Pembobotan.Cells[j,h]) >=
StrToFloat(Pembobotan.Cells[j,i]) then
begin
if Concordance.Cells[h,i] <> '-' then
begin
Concordance.Cells[i,h]
:=Concordance.Cells[i,h] +
FloatToStr(j) + ' ';
dance[i,h]:=StrToFloat(bobot.Cells[j,1])+
dance[i,h];
end;
end;
if StrToFloat(Pembobotan.Cells[j,h]) < StrToFloat
(Pembobotan.Cells[j,i]) then
begin
if Disordance.Cells[h,i] <> '-' then
begin
Disordance.Cells[i,h] :=
Disordance.Cells[i,h] +
FloatToStr(j) + ' ';
MatDis[i,h]:=StrToFloat(bobot.Cells[j,1]
)+MatDis[i,h];
end;
end;
end; end; end; end;

```

Menghasilkan

Concordance

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	1 2 6	1 2 4 6	1 2 4 6	1 2 3 4 6
Nipah	3 4 5 6	-	2 3 4 5 6	3 4 5 6	3 4 5 6
Bayur Laut	1 3 5 6	1 2 6	-	1 3 4 5 6	1 3 4 6
Nyirih	1 3 5 6	1 2 6	1 2 3 4 5 6	-	1 2 3 4 6
Tanjang Merah	1 3 5 6	1 2 6	1 2 4 5 6	1 2 4 5 6	-

Disordance

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	3 4 5	3 5	3 5	5
Nipah	1 2	-	1	1 2	1 2
Bayur Laut	2 4	3 4 5	-	2	2 5
Nyirih	2 4	3 4 5		-	5
Tanjang Merah	2 4	3 4 5	3	3	-

Tahap 4. Menghitung matriks *concordance* dan *discordance*

Kemudian dilakukan tahap keempat yakni, untuk menentukan nilai dari elemen-elemen matriks *concordance* adalah dengan menjumlahkan bobot-bobot yang termasuk pada himpunan *concordance*. Untuk menentukan nilai dari elemen-elemen pada matriks *discordance* dengan membagi maksimum selisih kriteria yang termasuk ke dalam himpunan bagian *disordance* dengan maksimum selisih nilai kriteria yang ada.

Listing

```
for h := 1 to total do
  begin
    for i := 1 to total do
      begin
        for j := 1 to totKolom do
          begin
            if StrToFloat(Pembobotan.Cells[j,h]) >=
              StrToFloat(Pembobotan.Cells[j,i]) then
              begin
                if MatConcordance.Cells[h,i] <> '-' then
                  MatConcordance.Cells[i,h]:=FloatToStr(Sum
                    (dance[i,h]));
                if MatDisordance.Cells[h,i] <> '-' then
                  MatDisordance.Cells[i,h]:=FloatToStr(Sum
                    (MatDis[i,h]));
              end;
            end;
          end;
        end;
      end;
    end;
  end;
```

Menghasilkan

Matriks Concordance

Tanjang Merah	8	5	11	11	-
Tancang	12	5	14	14	14
Bius	12	5	14	14	14
Tinjang	14	7	14	14	14
Tongke Besar	14	10	14	14	14
Bakau	14	7	14	14	14

Matriks Discordance

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	9	5	5	2
Nipah	3	-	1	3	3
Bayur Laut	6	9	-	2	4
Nyirih	6	9	0	-	2
Tanjang Merah	6	9	3	3	-

Tahap 5. Menghitung matriks dominan *concordance* dan *discordance*

Pada tahap 5 dibentuk matriks F sebagai matriks dominan *concordance* dan *discordance* yang dapat dibangun dengan membandingkan setiap nilai elemen matriks *concordance* dan *discordance* dengan nilai *threshold*. Setelah matriks F dibentuk, kemudian dibentuk matriks G sebagai matriks dominan *discordance* dapat dibangun dengan bantuan nilai *threshold*.

Listing

```
for h := 1 to total do
begin
  for i := 1 to total do
  begin
    if (DomConcordance.Cells[h,i] <> '-') then
    begin
      if (MatConcordance.Cells[i,h] <> '-') or
      (MatConcordance.Cells[i,h] <> null) then
        if StrToFloat(MatConcordance.Cells[i,h])
          >= StrToFloat(edThresConcor.Text) then
          DomConcordance.Cells[i,h]:='1'
        else
          DomConcordance.Cells[i,h]:='0';
        end;
      if (DomDisordanc.Cells[h,i] <> '-') then
      begin
        if (MatDisordance.Cells[i,h] <> '-') or
        (MatDisordance.Cells[i,h] <> null) then
          if StrToFloat(MatDisordance.Cells[i,h])
            >= StrToFloat(edThresDisor.Text) then
            DomDisordanc.Cells[i,h]:='1'
          else
            DomDisordanc.Cells[i,h]:='0';
          end;
        end;
      end;
    end;
  end;
end;
```

Menghasilkan

Matriks Dominan Concordance (F)

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	0	0	0	1
Nipah	1	-	1	1	1
Bayur Laut	0	0	-	1	0
Nyirih	0	0	1	-	1
Tanjang Merah	0	0	1	1	-

Matriks Dominan Disordance (G)

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	1	1	1	0
Nipah	1	-	0	1	1
Bayur Laut	1	1	-	0	1
Nyirih	1	1	0	-	0
Tanjang Merah	1	1	1	1	-

Nilai Threshold Matrik Dominan Concordance

10.21111111111111

Nilai Threshold Matrik Dominan Disordance

2.41

Tahap 6. Menentukan *aggregate dominance matrix*

Pada tahap ini dibentuk Matriks E sebagai *aggregate dominance* matriks yang setiap elemennya merupakan perkalian antara elemen matriks F dengan elemen matriks G yang bersesuaian.

Listing

```
for h := 1 to total do
begin
for i := 1 to total do
begin
if (Agregate.Cells[h,i] <> '-') then
begin
if (DomConcordance.Cells[i,h] <> '-') or
(DomDisordanc.Cells[i,h] <> '-') then
Agregate.Cells[i,h]:=FloatToStr(StrToFloat
(DomConcordance.Cells[i,h])*StrToFloat
(DomDisordanc.Cells[i,h]));
end;
end;
end;
end;
```

Menghasilkan

Agregate Dominance Matrix E

	Api-api	Nipah	Bayur Laut	Nyirih	Tanjang M
Api-api	-	0	0	0	0
Nipah	1	-	0	1	1
Bayur Laut	0	0	-	0	0
Nyirih	0	0	0	-	0
Tanjang Merah	0	0	1	1	-

Tahap 7. Eliminasi alternatif yang *less favourable*

Matriks E memberikan urutan pilihan dari setiap alternatif, yaitu bila $e_{kl} = 1$ maka alternative A_k merupakan alternatif yang lebih baik daripada A_l . Sehingga, baris dalam matriks E yang memiliki jumlah $e_{kl}=1$

paling sedikit dapat dieliminasi. Dengan demikian, alternative terbaik adalah alternatif yang mendominasi alternatif lainnya.

Listing

```
for h := 1 to total do
begin
  for i := 1 to total do
  begin
    if (Agregate.Cells[h,i] <> '-') then
    begin
      Agregate.cells[total+1,h]:=FloatToStr(StrToFloat
      (Agregate.Cells[i,h])+ StrToFloat
      (Agregate.cells[total+1,h]));
    end;
  end;
end;
```

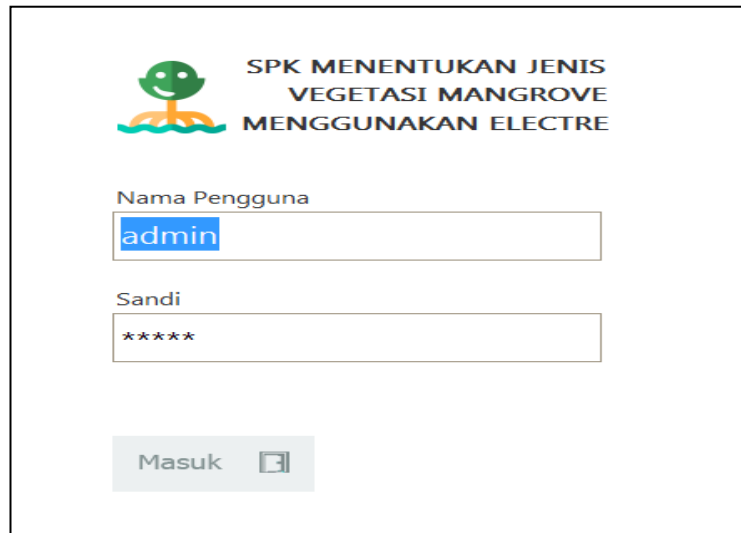
Menghasilkan

Ranking Vegetasi

Ranking	Nama Vegetasi	Poin
1	Nipah	8
2	Tanjang Merah	2
3	Bakau	1
4	Tinjang	1
5	Tongke Besar	0
6	Bius	0

2. Pembahasan Program Utama

a. Tampilan *Login*



SPK MENENTUKAN JENIS VEGETASI MANGROVE MENGGUNAKAN ELECTRE

Nama Pengguna
admin

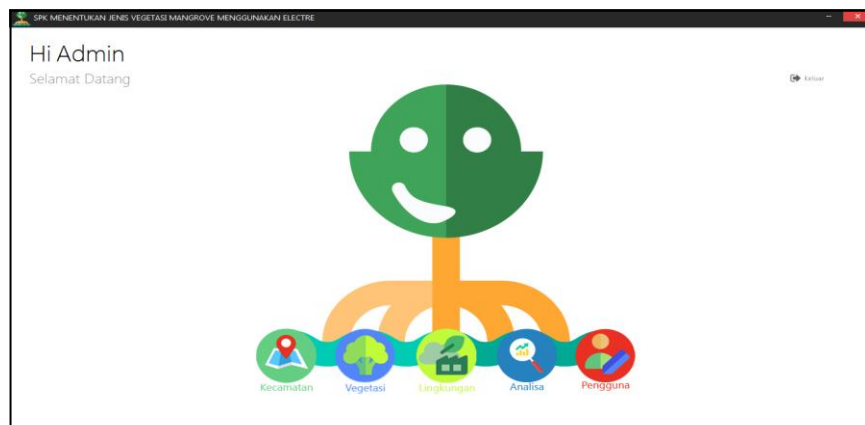
Sandi

Masuk

Gambar 4.12 Tampilan *Login*

Tampilan form login digunakan untuk masuk kedalam sistem dengan cara masukkan *Nama pengguna* dan *sandi* yang benar lalu klik *Masuk*

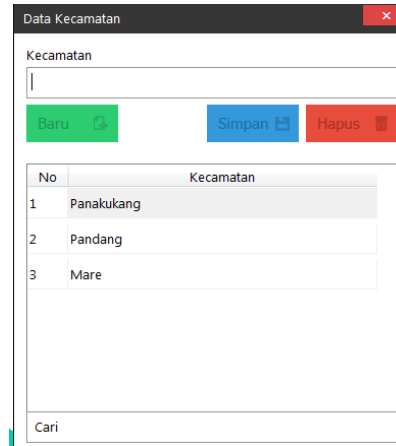
b. Tampilan Menu Utama



Gambar 4.13 Tampilan Menu Utama

Tampilan utama berisi Kecamatan, Vegetasi, Lingkungan, Analisa dan Pengguna.

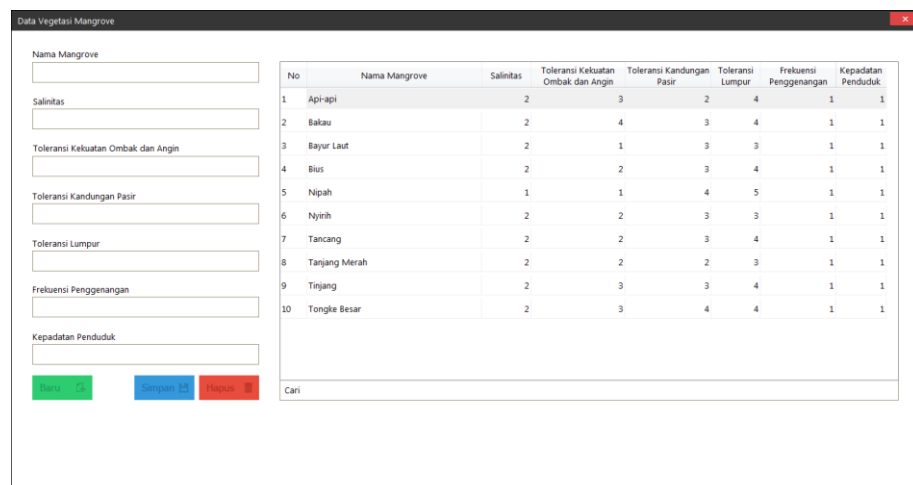
c. Tampilan Menu Data Kecamatan



Gambar 4.14 Menu Data Kecamatan

Menu Data Kecamatan yang merupakan salah satu menu di tampilan utama. Menu ini berisi data kecamatan. Untuk menambah data, klik tombol *Baru* lalu masukkan *Kecamatan*, kemudian klik tombol *Simpan*. Untuk memperbarui, klik salah satu *Kecamatan* lalu perbarui data *Kecamatan* kemudian klik *Simpan*. Sedangkan untuk menghapus, klik data *Kecamatan* lalu klik tombol *Hapus*.

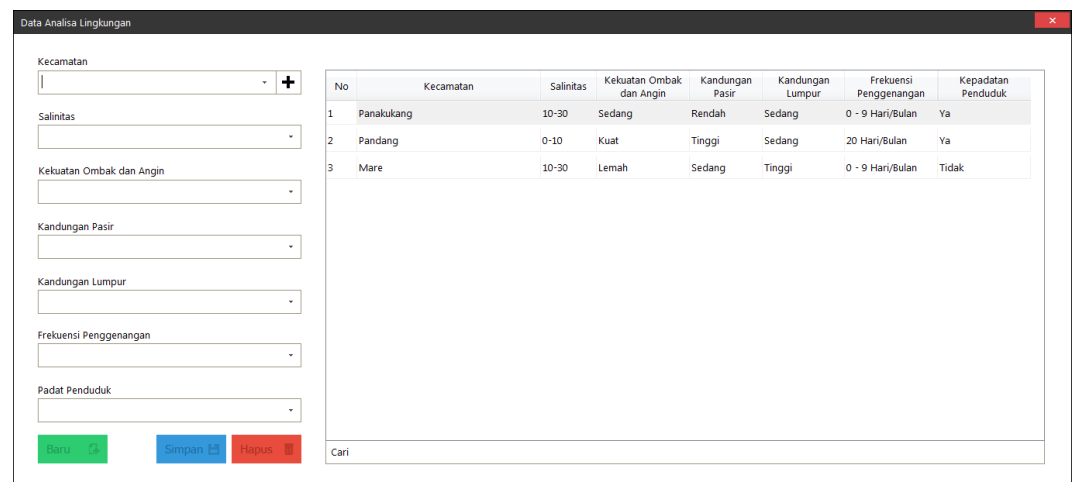
d. Tampilan Menu Data Vegetasi *Mangrove*



Gambar 4.15 Menu Data Vegetasi *Mangrove*

Tampilan Menu Data Vegetasi *Mangrove* yang merupakan salah satu menu di tampilan utama. Menu ini berisi *alternative Vegetasi Mangrove*. Untuk menambah data, klik tombol *Baru* lalu masukkan data vegetasi, kemudian klik tombol *Simpan*. Untuk memperbarui, klik salah satu data vegetasi lalu perbarui data yang salah kemudian klik *Simpan*. Sedangkan untuk menghapus, klik data yang ingin dihapus lalu klik tombol *Hapus*.

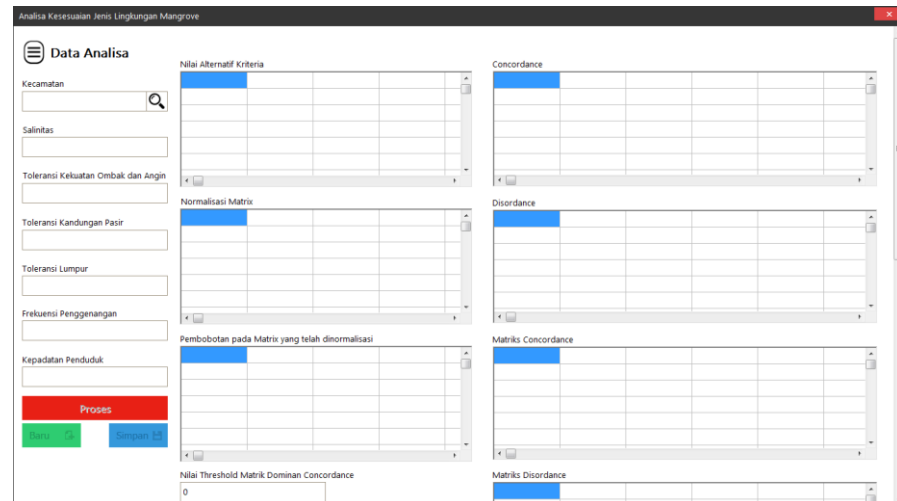
e. Tampilan Menu Data Analisa Lingkungan



Gambar 4.16 Menu Data Analisa Lingkungan

Tampilan Menu Data Analisa Lingkungan yang merupakan salah satu menu di tampilan utama. Menu ini berisi data lingkungan yang diterima dari bagian lapangan. Untuk menambah data, klik tombol *Baru* lalu masukkan data *Analisa*, kemudian klik tombol *Simpan*. Untuk memperbarui, klik salah satu data *Analisa* lalu perbarui data yang salah kemudian klik *Simpan*. Sedangkan untuk menghapus, klik data yang ingin dihapus lalu klik tombol *Hapus*.

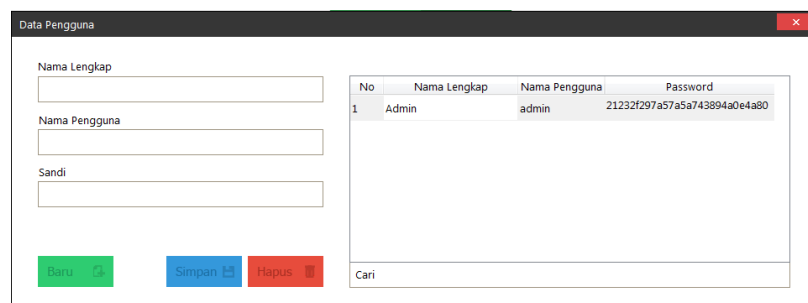
f. Tampilan Menu Data Analisa Kesesuaian Jenis Lingkungan *Mangrove*



Gambar 4.17 Menu Data Analisa Kesesuaian Jenis Lingkungan *Mangrove*

Menu ini digunakan untuk menganalisa kesesuaian jenis vegetasi *Mangrove*. Untuk memlulai Analisa, klik tombol *Cari* pada bagian *Input Kecamatan* lalu pilih kecamatan kemudian klik tombol *Proses*. Setelah data terproses lalu klik tombol *Simpan*. Maka *Laporan Anlisa* otomatis akan muncul.

g. Tampilan Menu Data Pengguna




Gambar 4.18 Menu Data Pengguna


Menu ini berisi data pengguna. Untuk menambah data, klik tombol *Baru* lalu masukkan data pengguna, kemudian klik tombol

Simpan. Untuk memperbarui, klik salah satu data pengguna lalu perbarui data yang salah kemudian klik *Simpan*. Sedangkan untuk menghapus, klik data yang ingin dihapus lalu klik tombol *Hapus*.

h. Tampilan Laporan



PEMERINTAH KOTA MAKASSAR
DINAS LINGKUNGAN HIDUP
 Jalan Jenderal Urip Sumoharjo No. 8 Tlp. (0411) 438381 Makassar 90144



Hasil Ranking dari Analisa Lingkungan
 Tanggal : 11 August 2017

Data Lingkungan

Kecamatan	Salinitas	Kekuatan ombak & angin	Kandungan Pasir	Kandungan Lumpur	Frekuensi Penggenangan	Kependudukan
Mariso	0-10	Lemah	Sedang	Tinggi	0 - 9 Hari/Bulan	Ya

Ranking Vegetasi Mangrove

Ranking	Nama Mangrove	Poin
1	Nipah	8
2	Tanjang Merah	2
3	Bakau	1
4	Tinjang	1
5	Tongke Besar	0
6	Blus	0
7	Tancang	0
8	Nyirih	0
9	Bayur Laut	0
10	Api-api	0

Berdasarkan Hasil Analisa diatas, maka ditentukan Mangrove jenis **Nipah** yang paling tepat untuk ditanam. Adapun alternatif selanjutnya yakni :

Ranking	Nama Mangrove
2	Tanjang Merah
3	Bakau

Makassar, 05-09-2017

Menyetujui,
 Kepala Dinas Lingkungan Hidup
 Kota Makassar

Drs. H. Abd. Gani Sirman, M.Si
 NIP : 19570822 198301 1 003

Gambar 4.19 Laporan Hasil Analisa

Laporan hasil analisa berupa ranking nama *Mangrove*, dimana yang berada pada ranking pertama merupakan *Mangrove* paling tepat untuk ditanam, sistem juga menampilkan alternatif yang berada pada ranking kedua dan ketiga.

E. Pengujian Sistem

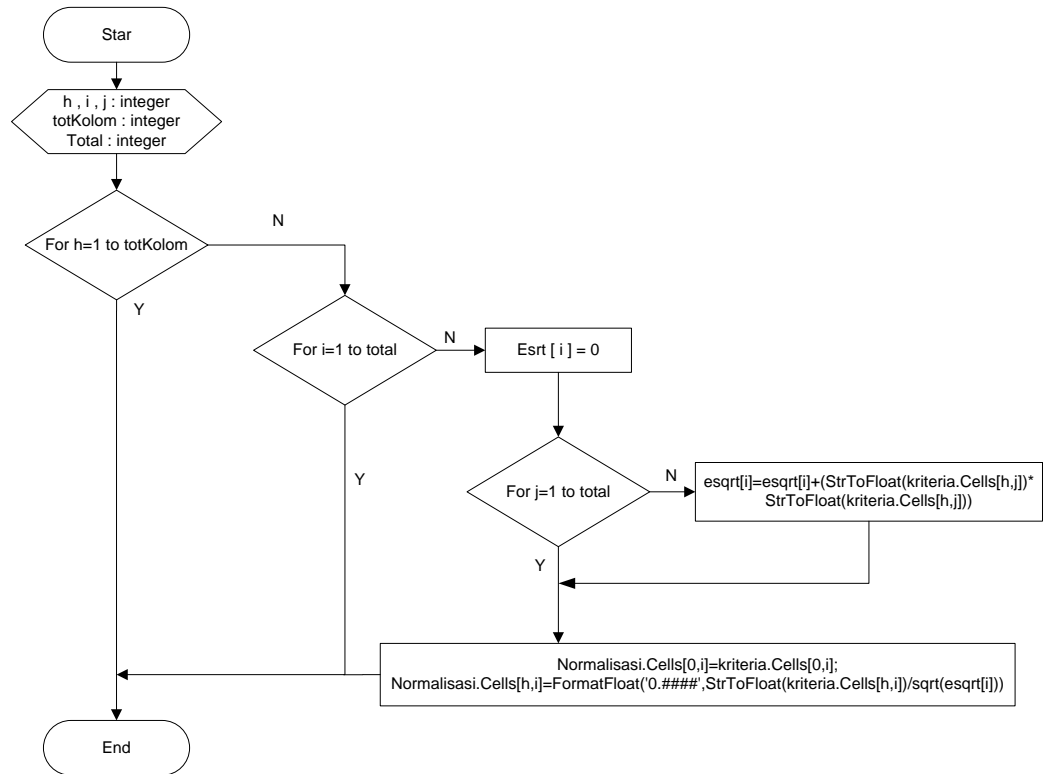
Pengujian sistem metode Electre :

1. Tahap 1. Normalisasi matriks keputusan

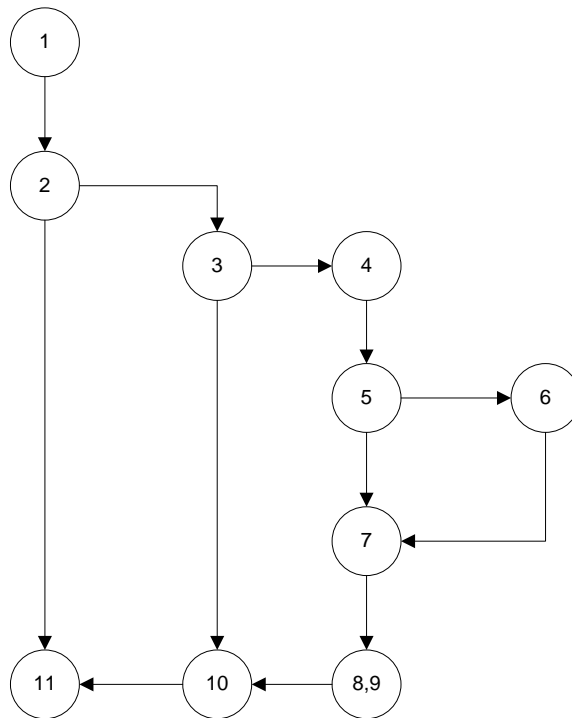
a. *Source Code*

```
for h:=1 to totKolom do
  begin for i:=1 to total do
    begin
      esqrt[i]:=0;
      for j:=1 to total do
        begin
          esqrt[i]:=esqrt[i]+(StrToFloat(kriteria.Cells[h,j])
          *StrToFloat(kriteria.Cells[h,j]));
        end;
      Normalisasi.Cells[0,i]:=kriteria.Cells[0,i];
      Normalisasi.Cells[h,i]:=FormatFloat('0.###',
      StrToFloat(kriteria.Cells[h,i])/sqrt(esqrt[i]));
    end;
  end;
end;
```

b. Flow Chart



c. Flowgraph



d. Kompleksitas Siklomatis

$$\begin{aligned}
 V(G) &= \text{Jumlah Region} \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= (12 - 10) + 2 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 3 + 1 \\
 &= 4
 \end{aligned}$$

e. Independent Path

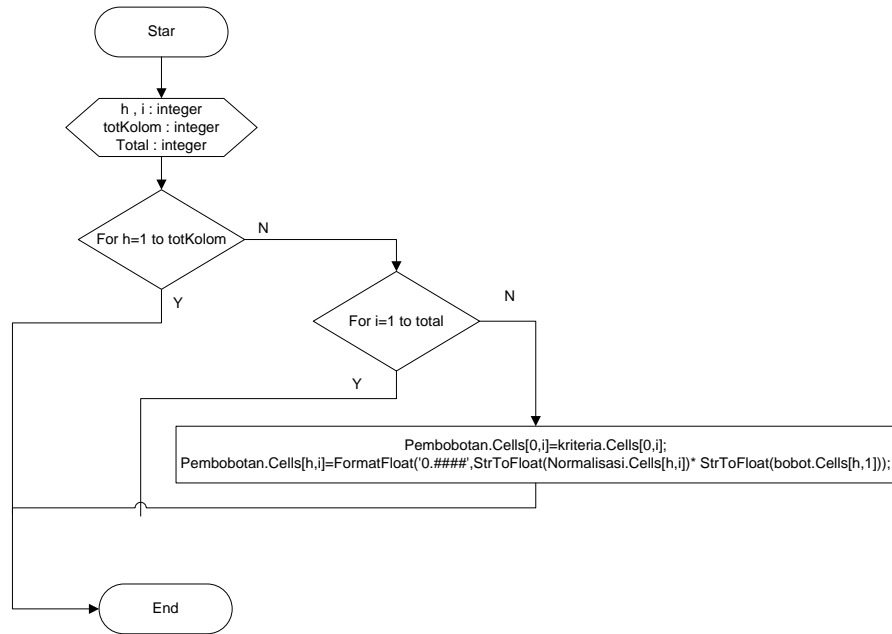
1,2,11

1,2,3,10,11

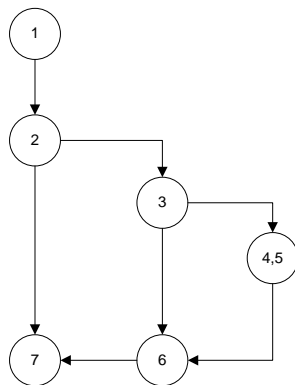
1,2,3,4,5,6,7,8,9,10,11

1,2,3,4,5,7,8,9,10,11

b. *Flow Chart*



c. *Flowgraph*



d. *Kompleksitas Siklomatis*

$$V(G) = \text{Jumlah Region} \\ = 3$$

$$V(G) = E - N + 2 \\ = (7 - 6) + 2 \\ = 3$$

$$V(G) = P + 1 \\ = 2 + 1 \\ = 3$$

e. *Independent Path*

1,2,6,7

1,2,3,6,7

1,2,3,4,5,6,7

f. *Graph Matrik*

	1	2	3	4	5	6	7
1		1					
2			1				1
3				1		1	
4					1		
5						1	
6							1

1-1=0

2-1=1

2-1=1

1-1=0

1-1=0

1-1=0

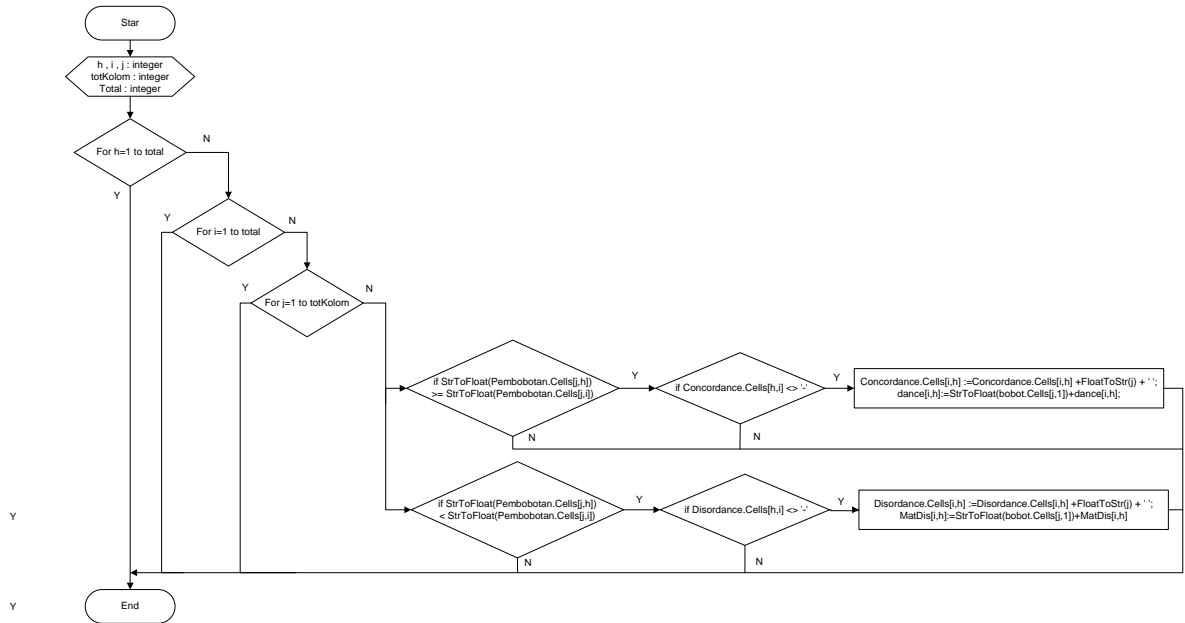
=2+1=3

3. Tahap 3. Menentukan himpunan *concordance* dan *discordance*

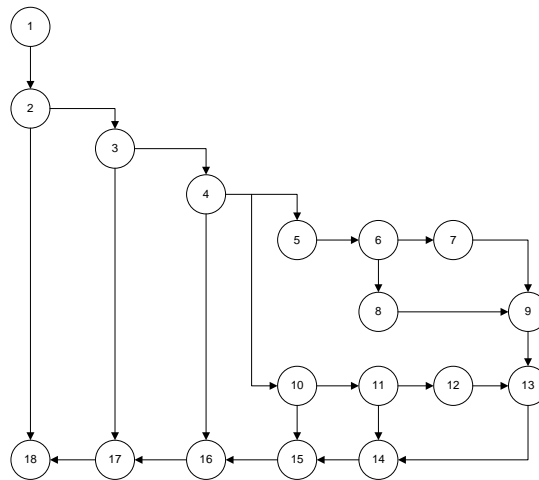
a. Source Code

```
for h := 1 to total do
begin
for i := 1 to total do
begin
for j := 1 to totKolom do
begin
if StrToFloat(Pembobotan.Cells[j,h]) >=
StrToFloat(Pembobotan.Cells[j,i]) then
begin
if Concordance.Cells[h,i] <> '-' then
begin
Concordance.Cells[i,h]
:=Concordance.Cells[i,h] +
FloatToStr(j) + ' ';
dance[i,h]:=StrToFloat(bobot.Cells[j,1])+dance[i,h];
end;
end;
if StrToFloat(Pembobotan.Cells[j,h]) < StrToFloat
(Pembobotan.Cells[j,i]) then
begin
if Disordance.Cells[h,i] <> '-' then
begin
Disordance.Cells[i,h]
:=Disordance.Cells[i,h] +
FloatToStr(j) + ' ';
MatDis[i,h]:=StrToFloat(bobot.Cells[j,1))+MatDis[i,h];
end;
end;
end;
end;
end;
end;
```

b. *Flow Chart*



c. *Flowgraph*



d. *Kompleksitas Siklomatis*

$$\begin{aligned}
 V(G) &= \text{Jumlah Region} \\
 &= 8 \\
 V(G) &= E - N + 2 \\
 &= (24 - 18) + 2 \\
 &= 8 \\
 V(G) &= P + 1 \\
 &= 7 + 1 = 8
 \end{aligned}$$

e. *Independent Path*

1,2,18

1,2,3,17,18

1,2,3,4,16,17,18

1,2,3,4,10,15,16,17,18

1,2,3,4,10,11,14,15,16,17,18

1,2,3,4,10,11,12,13,14,15,16,17,18

1,2,3,4,5,6,8,9,13,14,15,16,17,18

1,2,3,4,5,6,7,9,13,14,15,16,17,18

f. *Graph Matrik*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		1																
2			1															1
3				1													1	
4					1					1						1		
5						1												
6							1	1										
7									1									
8									1									
9													1					
10											1				1			
11												1		1				
12													1					
13														1				
14															1			
15																1		
16																	1	
17																		1

1-1=0

2-1=1

2-1=1

3-1=2

1-1=0

2-1=1

1-1=0

1-1=0

1-1=0

2-1=1

2-1=1

1-1=0

1-1=0

1-1=0

1-1=0

1-1=0

1-1=0

1-1=0

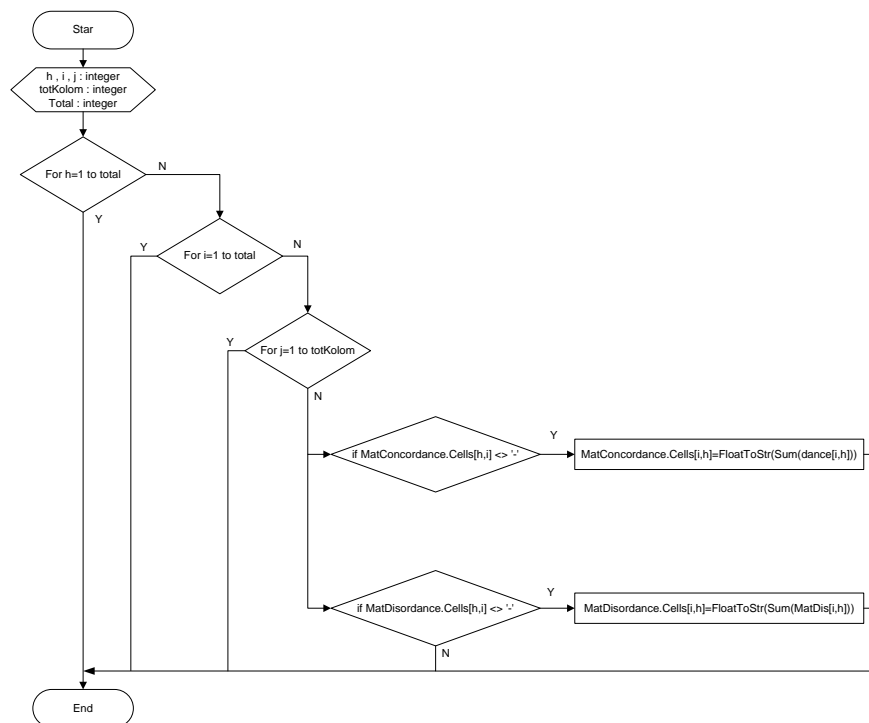
=7+1=8

4. Tahap 4. Menghitung matriks *concordance* dan *discordance*

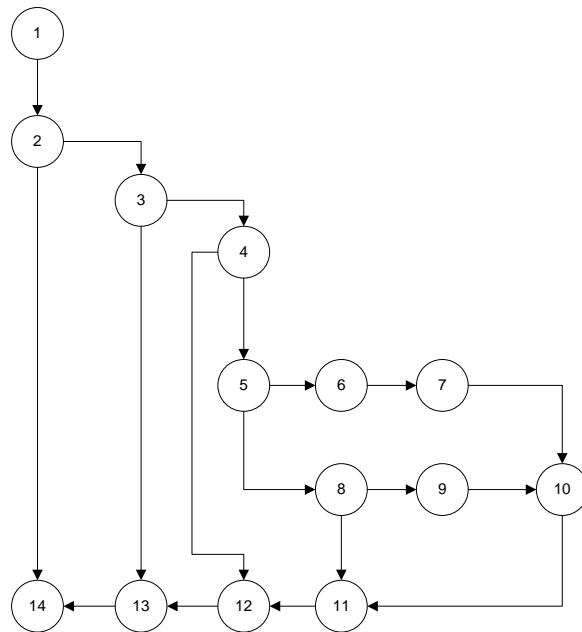
a. Source Code

```
for h := 1 to total do
begin
for i := 1 to total do
begin
for j := 1 to totKolom do
begin
if StrToFloat(Pembobotan.Cells[j,h]) >=
StrToFloat(Pembobotan.Cells[j,i]) then
begin
if MatConcordance.Cells[h,i] <> '-' then
MatConcordance.Cells[h,i]:=FloatToStr(Sum
(dance[i,h]));
if MatDisordance.Cells[h,i] <> '-' then
MatDisordance.Cells[h,i]:=FloatToStr(Sum
(MatDis[i,h]));
end;
end;
end;
end;
end;
```

b. Flow Chart



c. *Flowgraph*



d. Kompleksitas Siklomatis

$$V(G) = \text{Jumlah Region} \\ = 6$$

$$V(G) = E - N + 2 \\ = (18 - 14) + 2 \\ = 6$$

$$V(G) = P + 1 \\ = 5 + 1 \\ = 6$$

e. *Independent Path*

1,2,14

1,2,3,13,14

1,2,3,4,12,13,14

1,2,3,4,5,6,7,10,11,12,13,14

1,2,3,4,5,8,11,12,13,14

1,2,3,4,5,8,9,10,11,12,13,14

f. *Graph Matrik*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1		1												
2			1											1
3				1									1	
4					1							1		
5						1		1						
6							1							
7										1				
8									1		1			
9										1				
10											1			
11												1		
12													1	
13														1

1-1=0
 2-1=1
 2-1=1
 2-1=1
 2-1=1
 1-1=0
 1-1=0
 2-1=1
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 =5+1=6

5. Tahap 5. Menghitung matriks dominan *concordance* dan *discordance*

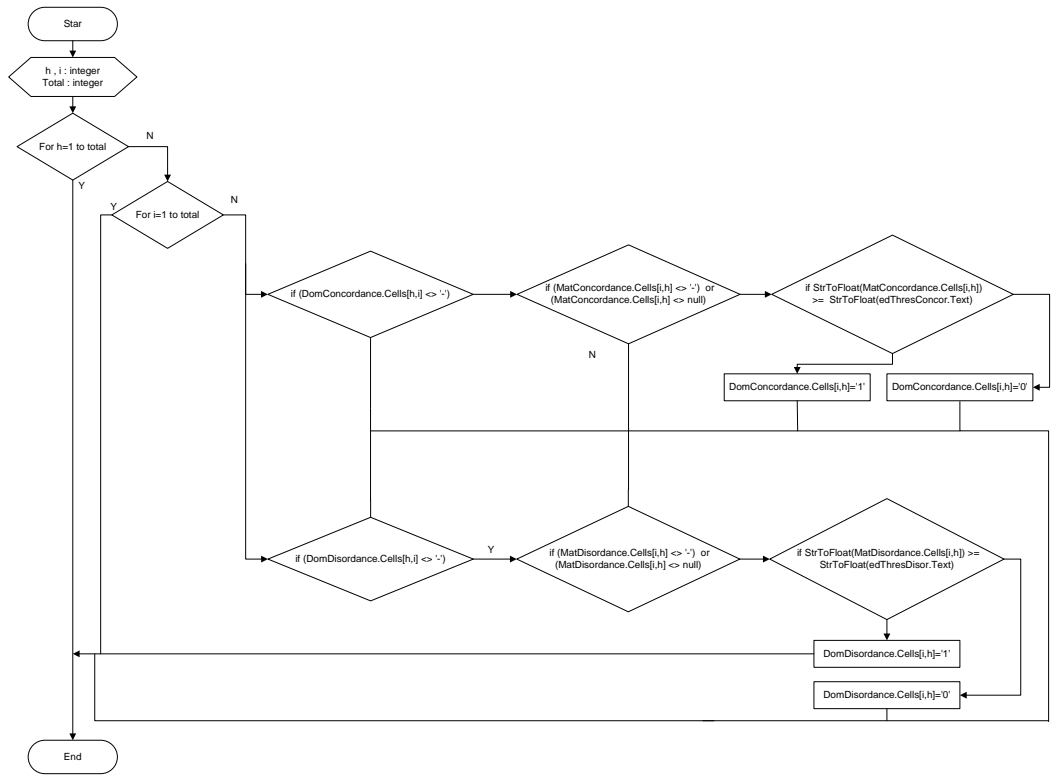
a. *Source Code*

```

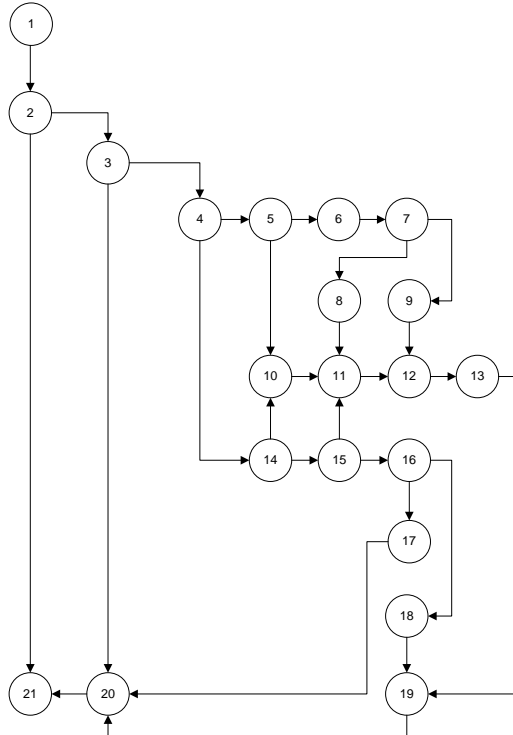
for h := 1 to total do
  begin
    for i := 1 to total do
      begin
        if (DomConcordance.Cells[h,i] <> '-') then
          begin
            if (MatConcordance.Cells[i,h] <> '-') or
              (MatConcordance.Cells[i,h] <> null) then
              if StrToFloat(MatConcordance.Cells[i,h]) >=
                StrToFloat(edThresConcor.Text) then
                DomConcordance.Cells[i,h]:='1'
            else
              DomConcordance.Cells[i,h]:='0';
            end;
          if (DomDisordanc.Cells[h,i] <> '-') then
            begin
              if (MatDisordance.Cells[i,h] <> '-') or
                (MatDisordance.Cells[i,h] <> null) then
                if StrToFloat(MatDisordance.Cells[i,h]) >=
                  StrToFloat(edThresDisor.Text) then
                  DomDisordanc.Cells[i,h]:='1'
            else
              DomDisordanc.Cells[i,h]:='0';
            end;
          end;
        end;
      end;
    end;
  end;

```

b. Flow Chart



c. Flowgraph



d. Kompleksitas Siklomatis

$$\begin{aligned}V(G) &= \text{Jumlah } Region \\ &= 6\end{aligned}$$

$$\begin{aligned}V(G) &= E - N + 2 \\ &= (28 - 21) + 2 \\ &= 9\end{aligned}$$

$$\begin{aligned}V(G) &= P + 1 \\ &= 8 + 1 \\ &= 9\end{aligned}$$

e. *Independent Path*

1,2,21

1,2,3,20,21

1,2,3,4,5,10,11,12,13,19,20,21

1,2,3,4,5,6,7,8,11,12,13,19,20,21

1,2,3,4,5,6,7,9,12,13,19,20,21

1,2,3,4,14,15,16,17,20,21

1,2,3,4,14,10,11,12,13,19,20,21

1,2,3,4,14,15,11,12,13,19,20,21

1,2,3,4,14,15,16,18,19,20,21

f. *Graph Matrik*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
1		1																				
2			1																		1	
3				1																1		
4					1									1								
5						1				1												
6							1															
7								1	1													
8											1											
9												1										
10											1											
11												1										
12													1									
13																			1			
14										1					1							
15											1					1						
16																	1	1				
17																				1		
18																			1			
19																					1	
20																						1

1-1=0
 2-1=1
 2-1=1
 2-1=1
 2-1=1
 1-1=0
 2-1=1
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 2-1=1
 2-1=1
 2-1=1
 1-1=0
 1-1=0
 1-1=0
 1-1=0
 =8+1=9

6. Tahap 6. Menentukan *agregate dominance matrix*

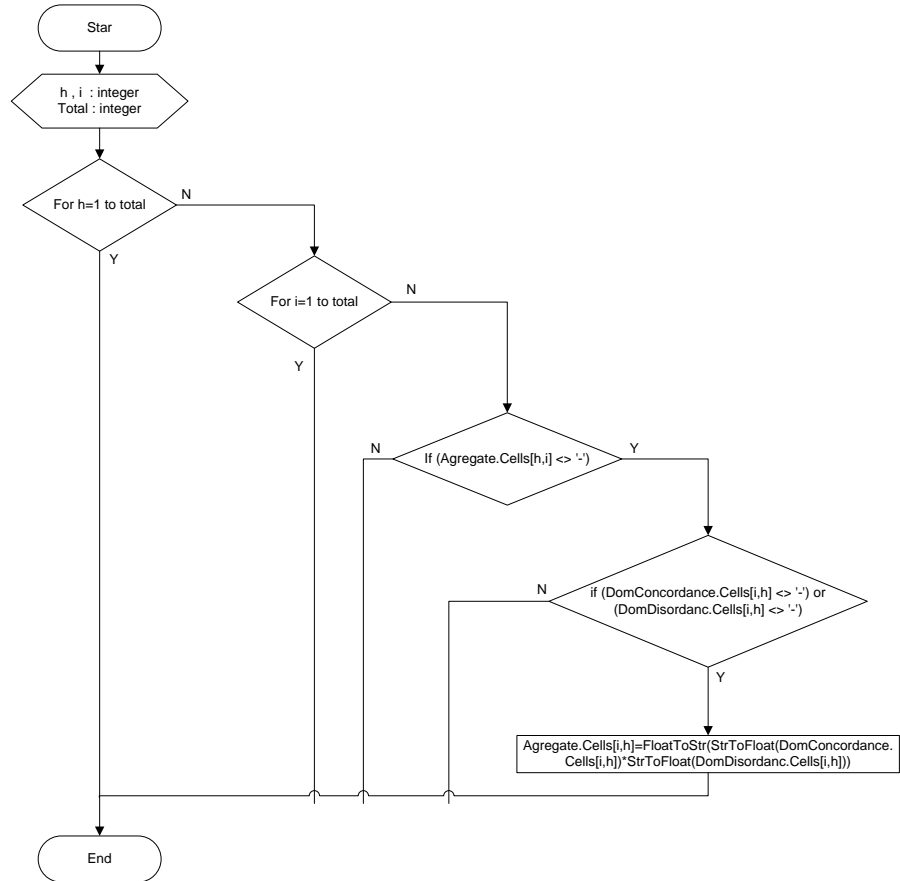
a. *Source Code*

```

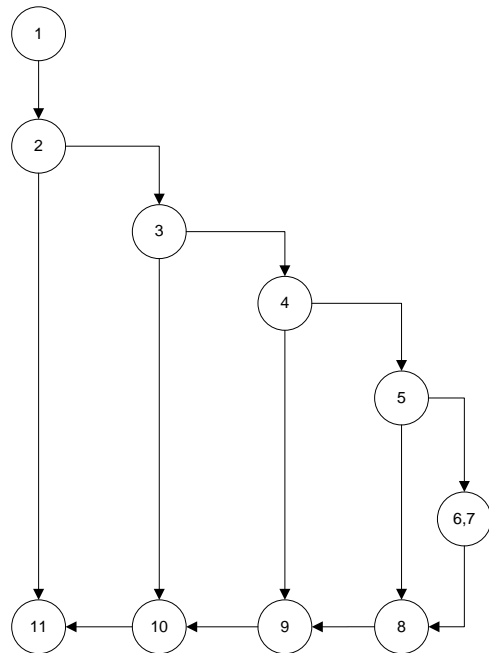
for h := 1 to total do
begin
    for i := 1 to total do
    begin
        if (Agregate.Cells[h,i] <> '-') then
        begin
            if (DomConcordance.Cells[i,h] <> '-') or
            (DomDisordanc.Cells[i,h] <> '-') then
            Agregate.Cells[i,h]:=FloatToStr(StrToFloat
            (DomConcordance.Cells[i,h])*StrToFloat
            (DomDisordanc.Cells[i,h]));
            end;
        end;
    end;
end;

```

b. *Flow Chart*



c. *Flowgraph*



d. Kompleksitas Siklomatis

$$\begin{aligned} V(G) &= \text{Jumlah Region} \\ &= 5 \end{aligned}$$

$$\begin{aligned} V(G) &= E - N + 2 \\ &= (13 - 10) + 2 \\ &= 5 \end{aligned}$$

$$\begin{aligned} V(G) &= P + 1 \\ &= 4 + 1 \\ &= 5 \end{aligned}$$

e. *Independent Path*

1,2,11

1,2,3,10,11

1,2,3,4,9,10,11

1,2,3,4,5,8,9,10,11

1,2,3,4,5,6,7,8,9,10,11

f. *Graph Matrik*

	1	2	3	4	5	6	7	8	9	10	11
1		1									
2			1								1
3				1						1	
4					1				1		
5						1		1			
6							1				
7								1			
8									1		
9										1	
10											1

1-1=0

2-1=1

2-1=1

2-1=1

2-1=1

1-1=0

1-1=0

1-1=0

1-1=0

1-1=0

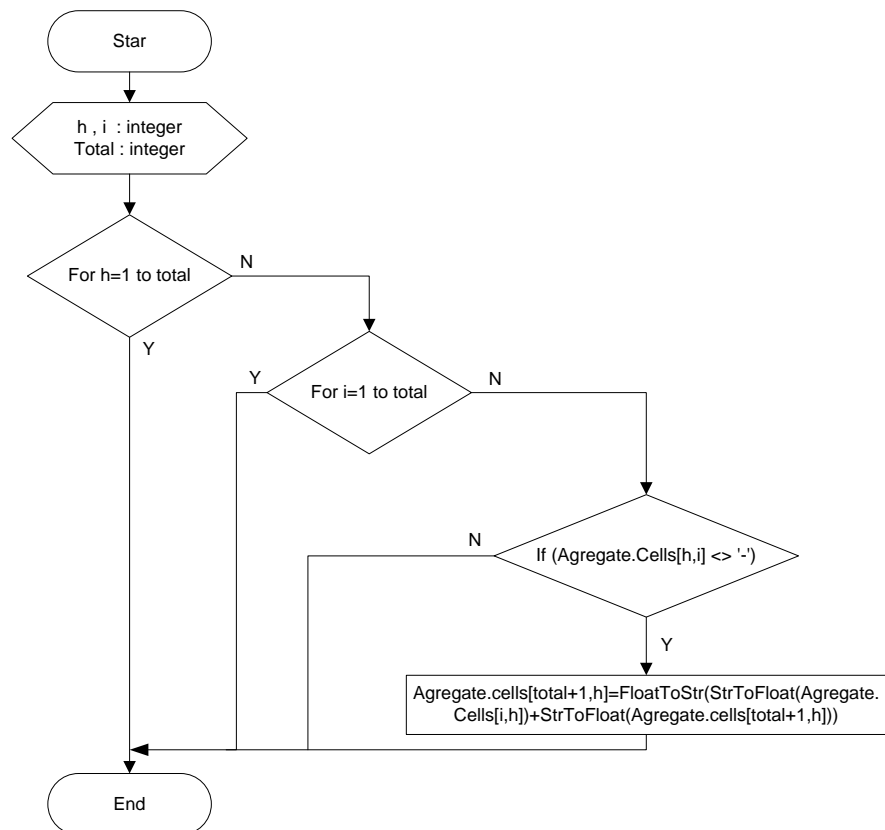
= 4+1=5

7. Tahap 7. Eliminasi alternatif yang *less favourable*

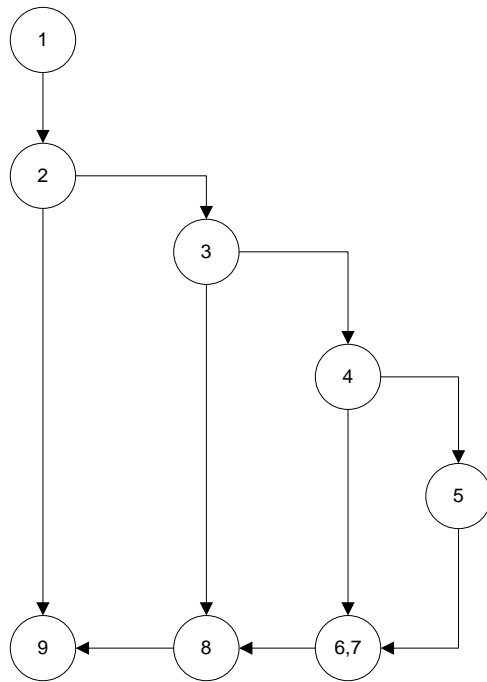
a. Source Code

```
for h := 1 to total do
begin
  for i := 1 to total do
  begin
    if (Agregate.Cells[h,i] <> '-') then
    begin
      Agregate.cells[total+1,h]:=FloatToStr
      (StrToFloat (Agregate.Cells[i,h])+
      StrToFloat (Agregate.cells[total+1,h]));
    end;
  end;
end;
```

b. Flow Chart



c. *Flowgraph*



d. Kompleksitas Siklomatis

$$\begin{aligned}
 V(G) &= \text{Jumlah Region} \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= E - N + 2 \\
 &= (10 - 8) + 2 \\
 &= 4
 \end{aligned}$$

$$\begin{aligned}
 V(G) &= P + 1 \\
 &= 3 + 1 \\
 &= 4
 \end{aligned}$$

e. *Independent Path*

1,2,9

1,2,3,8,9

1,2,3,4,6,7,8,9

1,2,3,4,5,6,7,8,9

f. *Graph Matrik*

	1	2	3	4	5	6	7	8	9
1		1							
2			1						1
3				1				1	
4					1	1			
5						1			
6							1		
7								1	
8									1

$$1-1=0$$

$$2-1=1$$

$$2-1=1$$

$$2-1=1$$

$$1-1=0$$

$$2-1=1$$

$$1-1=0$$

$$1-1=0$$

$$=3+1=4$$

BAB V

KESIMPULAN

A. Kesimpulan

1. Sistem ini memberikan hasil berupa ranking *Mangrove* berdasarkan analisa lingkungan dan penentuan Jenis *Mangrove* yang cocok untuk ditanam.
2. Sistem ini dapat digunakan untuk membantu Dinas Lingkungan Hidup untuk menentukan jenis vegetasi *Mangrove* yang cocok dalam suatu lingkungan.
3. Electre digunakan pada kondisi dimana alternatif yang kurang sesuai dengan kriteria dieliminasi dan alternatif yang sesuai dapat dihasilkan. Sehingga cocok untuk kasus-kasus dengan banyak alternatif namun hanya sedikit kriteria yang dilibatkan.

B. Saran

Setelah melalui proses pengembangan pada Sistem Pendukung Keputusan Menentukan Jenis Vegetasi *Mangrove* menggunakan Algoritma *Elimination And Choice Expressing Reality* (Electre), maka disarankan agar sistem yang telah dibuat dikembangkan menjadi aplikasi *mobile* yang lebih *modern* lagi.

DAFTAR PUSTAKA

- AF. Stoner (Hasan, 2002). Jilid I. *Teori Pengambilan Keputusan*. Jakarta: Ghalia Indonesia
- Atmosudirdjo, Prof. Dr. Prajudi, S.H. 1979. *Pengambilan Keputusan (Decision Making)*. Jakarta.
- Black. 2007. *Behavioral* atau *black box* tests.
- Dadan Umar Daihani. 2001. *Sistem Pendukung Keputusan*, Penerbit Elex Media Komputindo, Jakarta.
- Efraim Turban. 2005. *Decision Support Systems and Intelligent System*.
- George R. Terry dan Leslie W. Rue. 2008. *Dasar-Dasar Manajemen*, Bumi Aksara, Jakarta.
- Herdino Ambargo, 2013. *Definisi dan Dasar Pengambilan Keputusan*. [Online]. dhino-ambargo.blogspot.com. diakses pada tanggal 07 Januari 2017.
- Ilmu Teknik Sipil. 2017. *Pemeriksaan Kandungan Lumpur*. [online]. <https://ilmutekniksipil.com>. diakses tanggal 06 Agustus 2017.
- Janko, Wolfgang & Bernoider Edward, 2005, *Multi-Criteria Decision Making An Application Study of ELECTRE & TOPSIS*. Dalam *Fuzzy Multi-Attribute Decision Making (MADM)*. Yogyakarta : Graha Ilmu.
- Kadir, Abdul. 2004. *Pemrograman Database Dengan Delphi 7*. Andi offset: Yogyakarta.
- Komputer, Wahana. 2010. *Panduan Belajar MySQL Database Server*. Media Kita : Jakarta.
- Kusmana, C. 2002. *Ekologi Mangrove*. Fakultas Kehutanan IPB. Bogor.

- Nugroho Bunafit. 2010. Database Relasional dengan MySQL. Yogyakarta. Penerbit Andi.
- M. Khazali.1999. Panduan Teknis Penanaman *Mangrove* bersama Masyarakat, Bogor.
- Peraturan Menteri Kehutanan. Nomor : P.03/MENHUT-V/2004 Tanggal 22 Juli 2004
- Raymond Etal. 2010. *Hutan Mangrove Di Daerah Pesisir Makassar*. [Online]. goldenheart.id. diakses pada tanggal 26 April 2017.
- Retno Kusumaningrum. 2010. SPK untuk menganalisa kesesuaian Jenis vegetasi *Mangrove*. Diponegoro.
- Sehatista. 2017. *Anemometer*. [online]. [https:// Sehatista.com](https://Sehatista.com). diakses tanggal 06 Agustus 2017.
- Semuaikan. 2017. *Pengertian salinitas*. [online]. <https:// semuaikan.com>. diakses tanggal 06 Agustus 2017.
- Sumiati. 2007. *Metode Pembelajaran Pendekatan Individual*. Bandung.
- Sholiq. 2006. *Pemodelan Sistem Informasi Berorientasi Obyek dengan UML*. Yogyakarta : Graha Ilmu.
- Tian. 2005. *Black box Testing* atau disebut juga *Functional Testing*.
- Wikipedia. 2017. *MySQL*. [online]. <https://id.wikipedia.org>. diakses tanggal 07 Januari 2017.
- Wikipedia. 2017. *Pengujian perangkat lunak*. [online]. <https://id.wikipedia.org>. diakses tanggal 07 Januari 2017.
- Yakub. 2012. *Pengantar Sistem Informasi*. Graha Ilmu : Yogyakarta.