



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan mengenai analisis dan gambaran dari sistem yang ada dan bagaimana perancangan program aplikasi yang akan dibuat.

3.1 Bisnis Proses

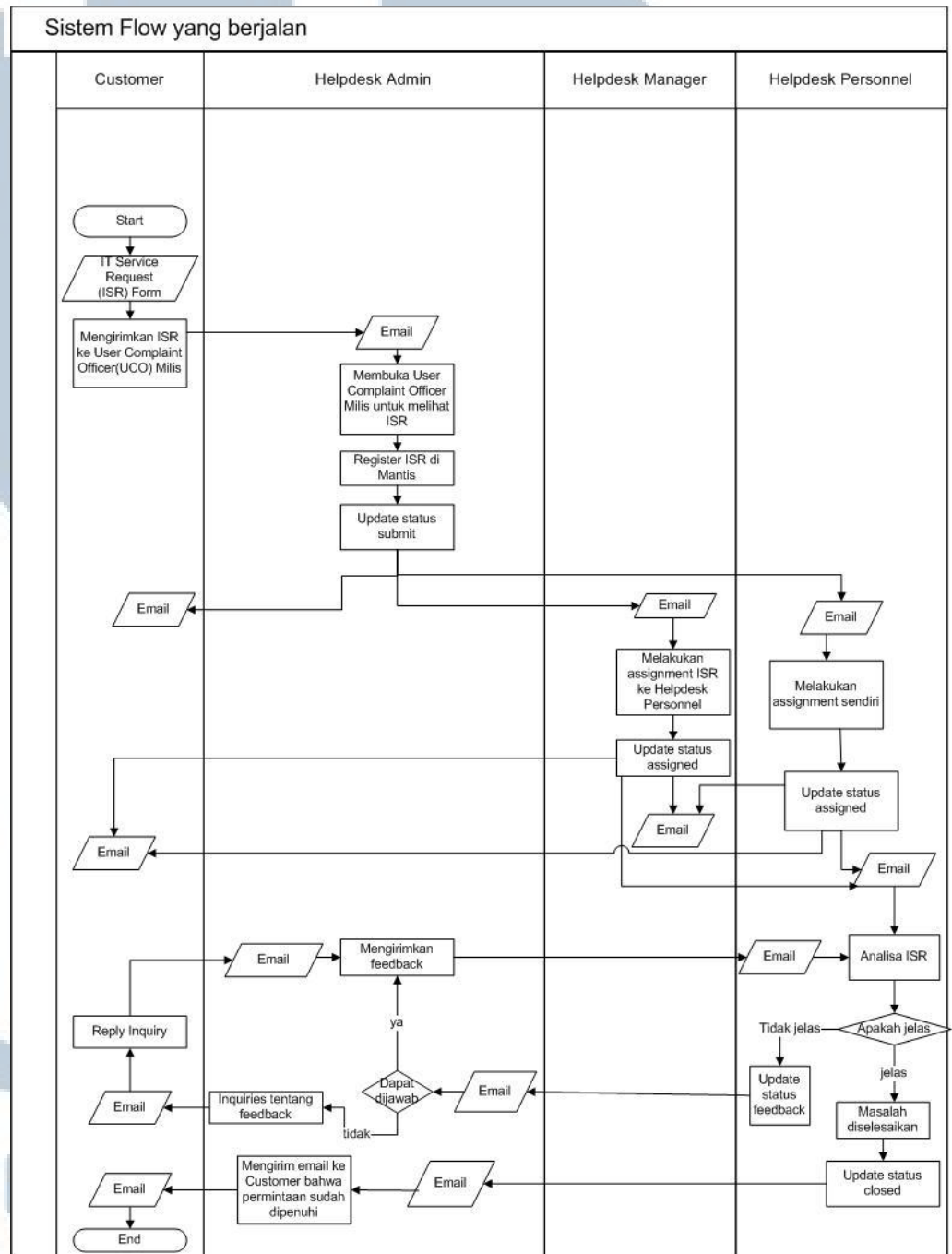
3.1.1 Bisnis Proses Sistem Berjalan

Pada sistem yang sedang berjalan, bisnis proses dapat diuraikan sebagai berikut. Pegawai PT WOM Finance yang memiliki masalah saat pengoperasian komputer mengisi form *IT Service Request (ISR)* kemudian mengirimkan *email* ke milis *User Complaint Officer (UCO)*. Setelah itu, *email* tersebut diregister oleh *helpdesk admin* ke mantis yang merupakan *software open source helpdesk* dengan status *submitted*. Lalu, mantis akan mengirim *email* notifikasi ke personel *helpdesk* dan *user* yang bersangkutan. Dilanjutkan *manager helpdesk* mengirimkan tiket kepada personel *helpdesk* dengan status *assign*. Selain dari *manager*, tiket juga dapat dipilih oleh personel *helpdesk* itu sendiri. Kemudian, mantis mengirimkan kembali *email* notifikasi ke personel *helpdesk* dan *customer (user)* yang bersangkutan). Setelah itu, personel *helpdesk* akan mengerjakan solusinya. Apabila masalah tersebut sudah diselesaikan maka status berubah menjadi *closed* dan admin akan mengirim *email* ke *user* bahwa masalah sudah diselesaikan. Namun, bila masalah tersebut kurang jelas, maka personel *helpdesk* akan mengubah tiket menjadi status *feedback* dan akan

mengirimkan *email* ke admin untuk ditanyakan. Apabila admin tidak dapat menjawab, maka *email* tersebut akan diteruskan oleh admin ke *customer* untuk mendapatkan penjelasan lebih lengkap. *Flowchart* dan bisnis proses ilustrasi di atas dapat dilihat pada Gambar 3.1 dan 3.2 di halaman berikut.



Sistem *flow* di bawah ini menunjukkan aliran sistem yang sedang berjalan



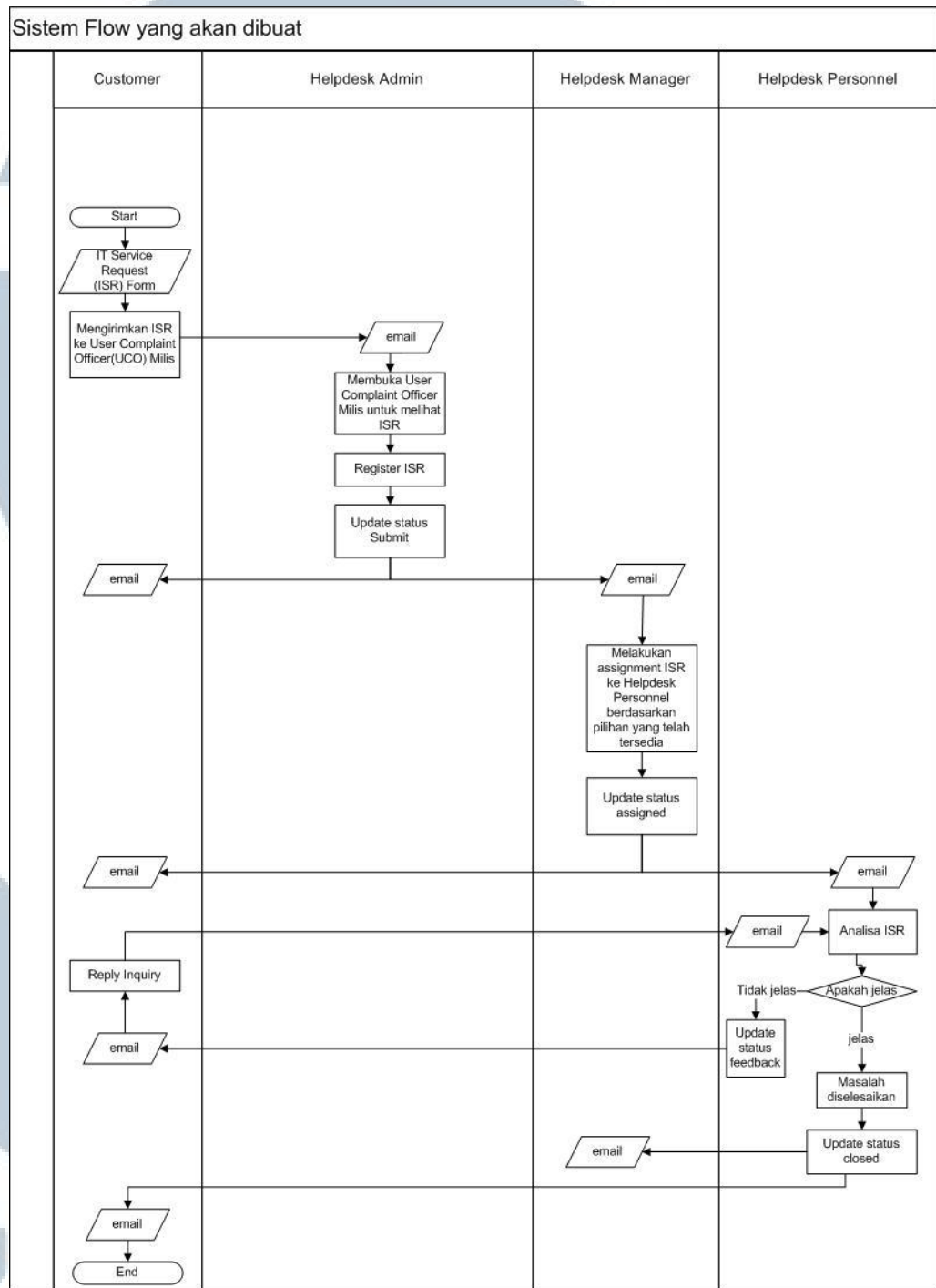
Gambar 3.1 Sistem *flow* yang sedang berjalan

3.1.2 Bisnis Proses Sistem Rancangan

Dalam penelitian ini, akan dibuatlah aplikasi *helpdesk* yang menggunakan pendekatan *Analytic Hierarchy Process* untuk membantu keputusan *manager* memilih personel *helpdesk* yang akan mengerjakan tiket. Admin, *manager*, dan personel *helpdesk* saja yang dapat melakukan *login* ke aplikasi ini. *Manager* berhak menentukan apakah *user* yang telah register ke aplikasi ini sesuai dengan jenis pekerjaannya yaitu sebagai admin atau personel. Admin bertugas untuk register ISR (*IT Service Request*) yang akan menjadi tiket untuk dikerjakan oleh personel. Setelah itu muncul notifikasi *email* ke *customer* dan *manager* dengan status *submit* bahwa ISR telah diregistrasi. *Manager* lalu membuka tiket tersebut dan memilih personel mana yang tersedia sesuai urutan prioritas dari hasil *Analytic Hierarchy Process*. Setelah memilih, maka personel tersebut serta *customer* akan mendapatkan notifikasi *email* bahwa tiket telah menjadi status *assigned*. Kemudian personel akan melakukan analisa, apakah keterangan yang dideskripsikan jelas sehingga dapat dikerjakan. Apabila kurang jelas, maka personel akan mengubah status tiket menjadi *feedback* dan mengirimkan *email* ke *customer*. Apabila jelas, personel akan mengerjakan dan setelah selesai, status diubah menjadi *closed*. Notifikasi *email* akan diterima oleh *customer* dan *manager*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

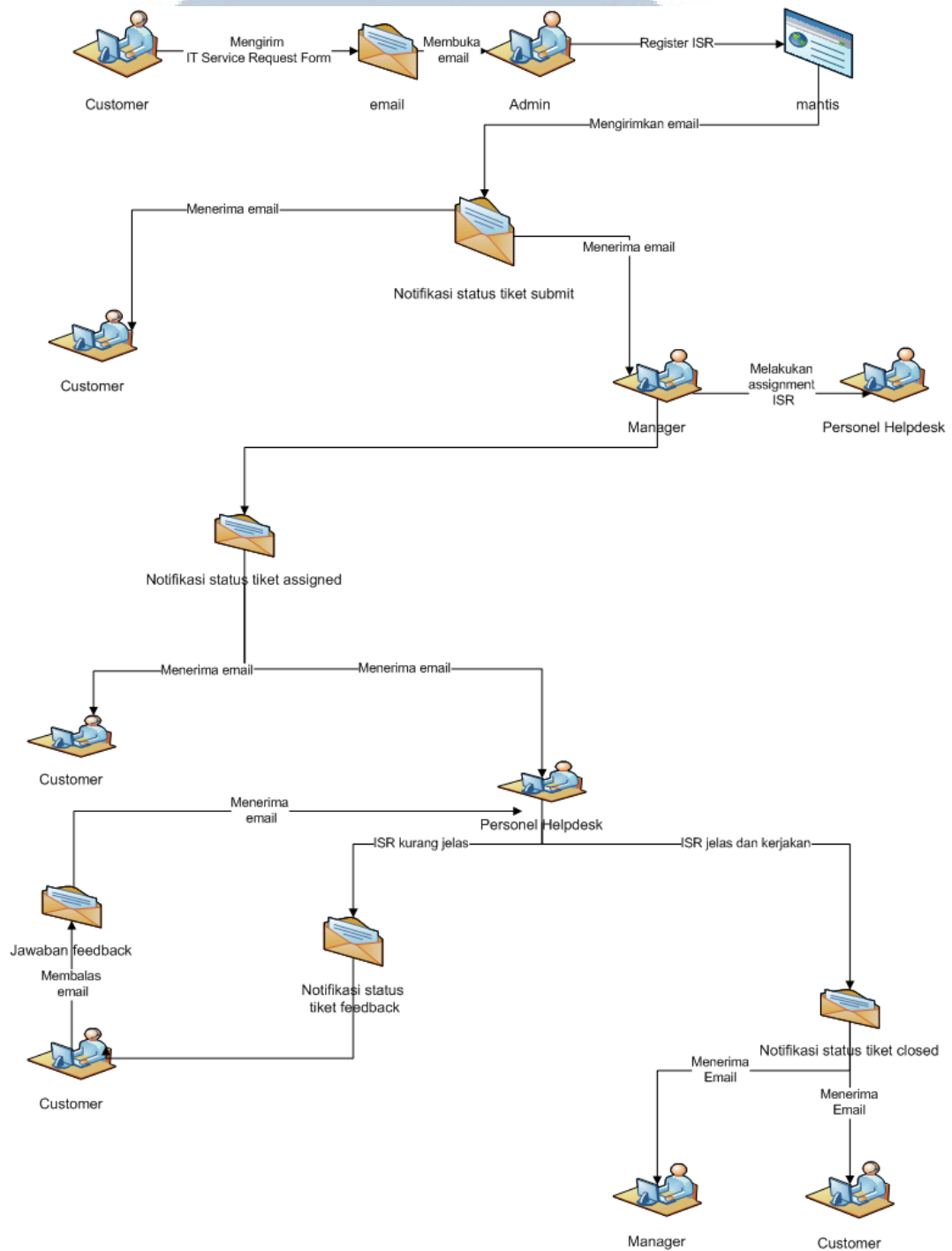
Sistem *Flow* di bawah ini menunjukkan aliran sistem yang akan dibuat



Gambar 3.3 Sistem *flow* yang akan dibuat

MULTIMEDIA
NUSANTARA

Bisnis Proses di bawah ini menunjukkan bisnis proses yang akan dibuat



Gambar 3.4 Bisnis Proses yang akan dibuat

MULTIMEDIA
NUSANTARA

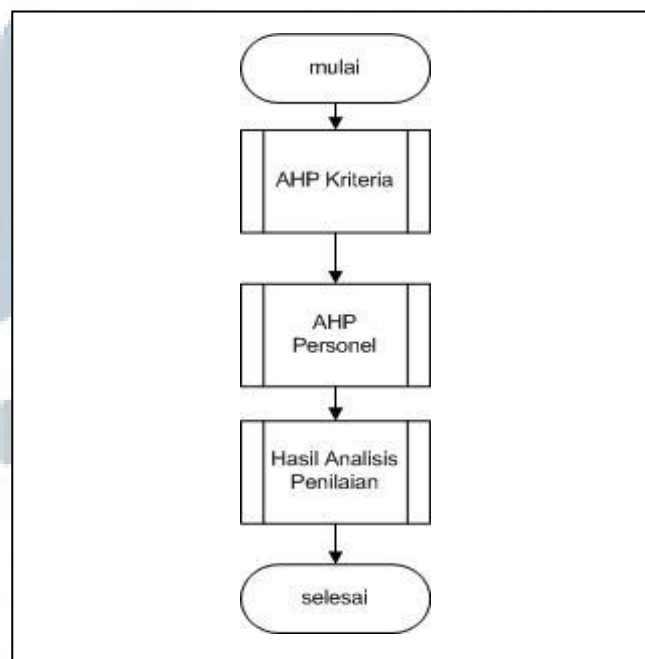
3.2 Diagram Sistem

Pada bagian ini akan dijelaskan diagram sebagai acuan pembuatan aplikasi helpdesk dimulai dari *flowchart*, *data flow diagram*, *entity relationship diagram*, struktur tabel, dan perancangan antarmuka.

Berikut ini diagram-diagram sistem yang digunakan dalam perhitungan AHP yang dimulai dari 3.2.1 *flowchart* sistem pendukung keputusan pemilihan personel sampai dengan 3.2.13 *flowchart* hasil analisis penilaian.

3.2.1 Flowchart Sistem Pendukung Keputusan Pemilihan Personel

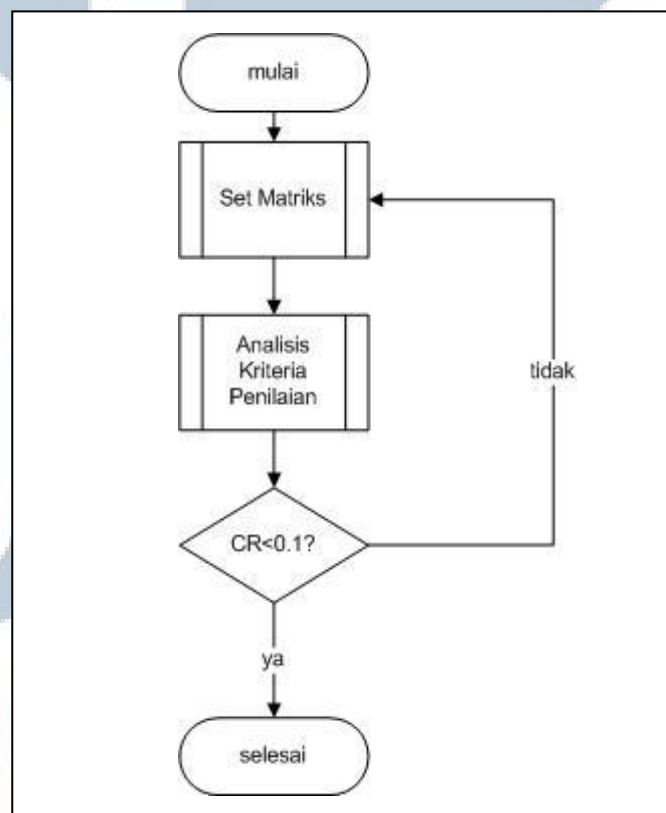
Flowchart di bawah ini menggambarkan proses AHP untuk menghitung bobot kriteria dan alternatif (personel) serta hasil analisis. Masing-masing proses ini akan dijelaskan lebih detail pada tiap sub bab. Dimulai dari AHP Kriteria yang akan dibahas lebih lanjut pada 3.2.2, AHP Personel pada 3.2.10, dan Hasil analisis penilaian pada 3.2.13.



Gambar 3.5 *Flowchart* Sistem Pendukung Keputusan Pemilihan Personel

3.2.2 Flowchart AHP Kriteria

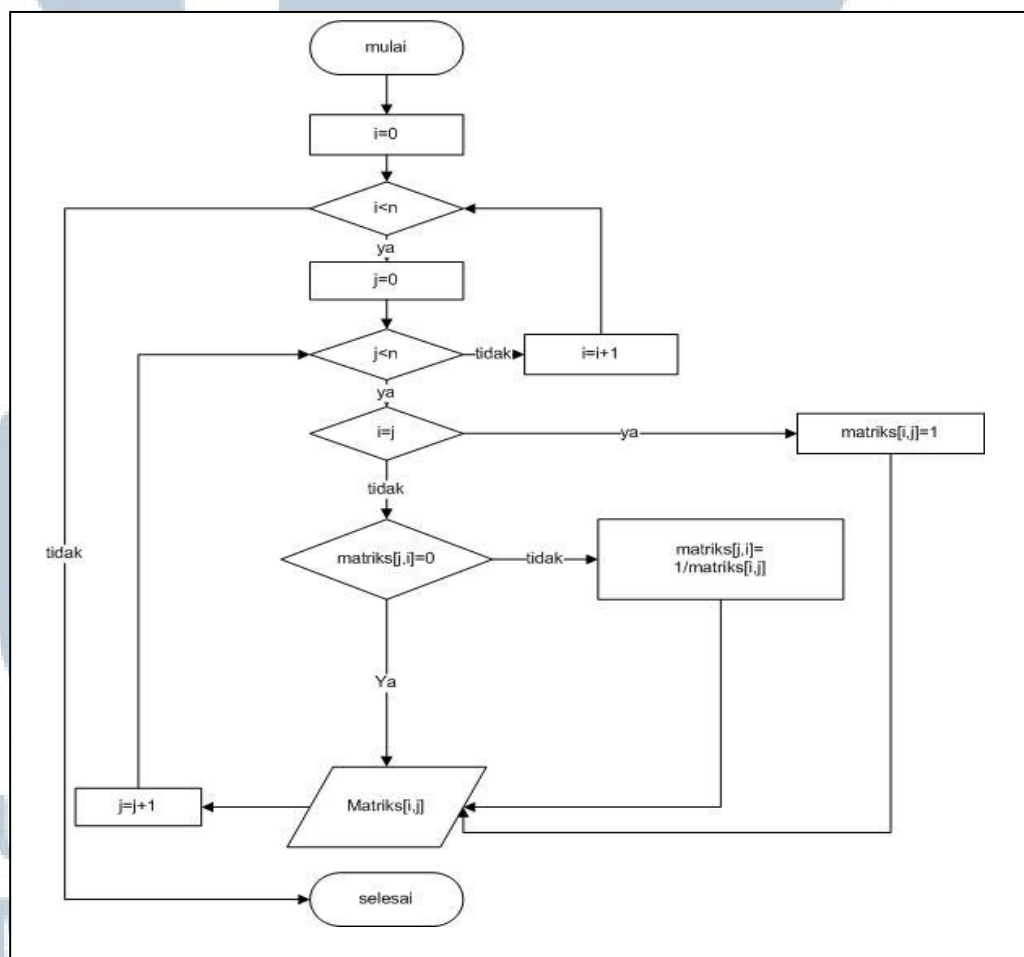
Proses yang terdapat dalam AHP kriteria ini adalah set matriks dan analisis kriteria penilaian. Proses Set Matriks akan dijelaskan lebih lanjut pada 3.2.3 dan proses Analisis Kriteria Penilaian akan dijelaskan lebih lanjut pada 3.2.4. Setelah proses analisis kriteria penilaian dilakukan maka akan didapatkan konsistensi rasio (CR). Apabila CR bernilai < 0.1 maka dapat dilanjutkan ke proses AHP personel. Apabila CR bernilai > 0.1 maka perlu melakukan proses set matriks kembali. Dalam hal ini, kriteria penilaian diambil dari *classification*.



Gambar 3.6 Flowchart AHP Kriteria

3.2.3 Flowchart Set Matriks

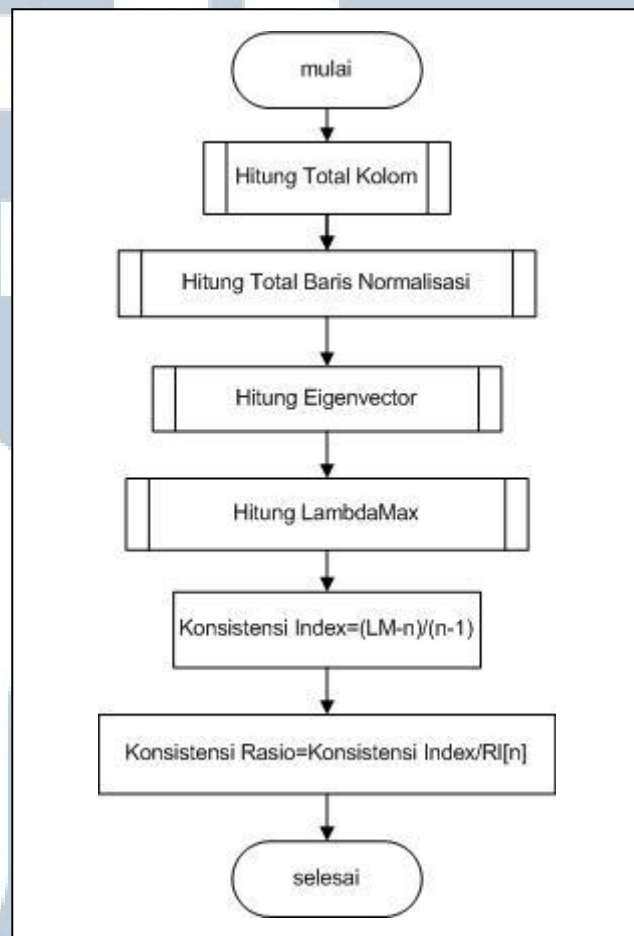
Flowchart ini menggambarkan pengisian nilai yang berasal dari masukan dari *manager helpdesk*. Pengisian dimulai dari tiap kolom pada satu baris matrix. Perbandingan matriks berpasangan akan memiliki angka 1 secara diagonal pada pojok kiri atas sampai dengan pojok kanan bawah yang berarti bahwa perbandingan terhadap dua hal yang sama adalah 1 atau *equally preferred*. Nilai yang diisi cukup pada area atas matriks yang bernilai 1 karena area bawah matriks yang bernilai 1 akan memiliki nilai kebalikannya.



Gambar 3.7 *Flowchart* Set Matriks

3.2.4 Flowchart Analisis Kriteria Penilaian

Setelah perbandingan berpasangan dilakukan, kemudian proses selanjutnya adalah hitung total kolom, hitung total baris normalisasi, hitung eigenvector, hitung lambdamax. Setelah itu, didapatkan konsistensi index dan konsistensi rasionya. Masing-masing proses ini akan dijelaskan satu per satu.

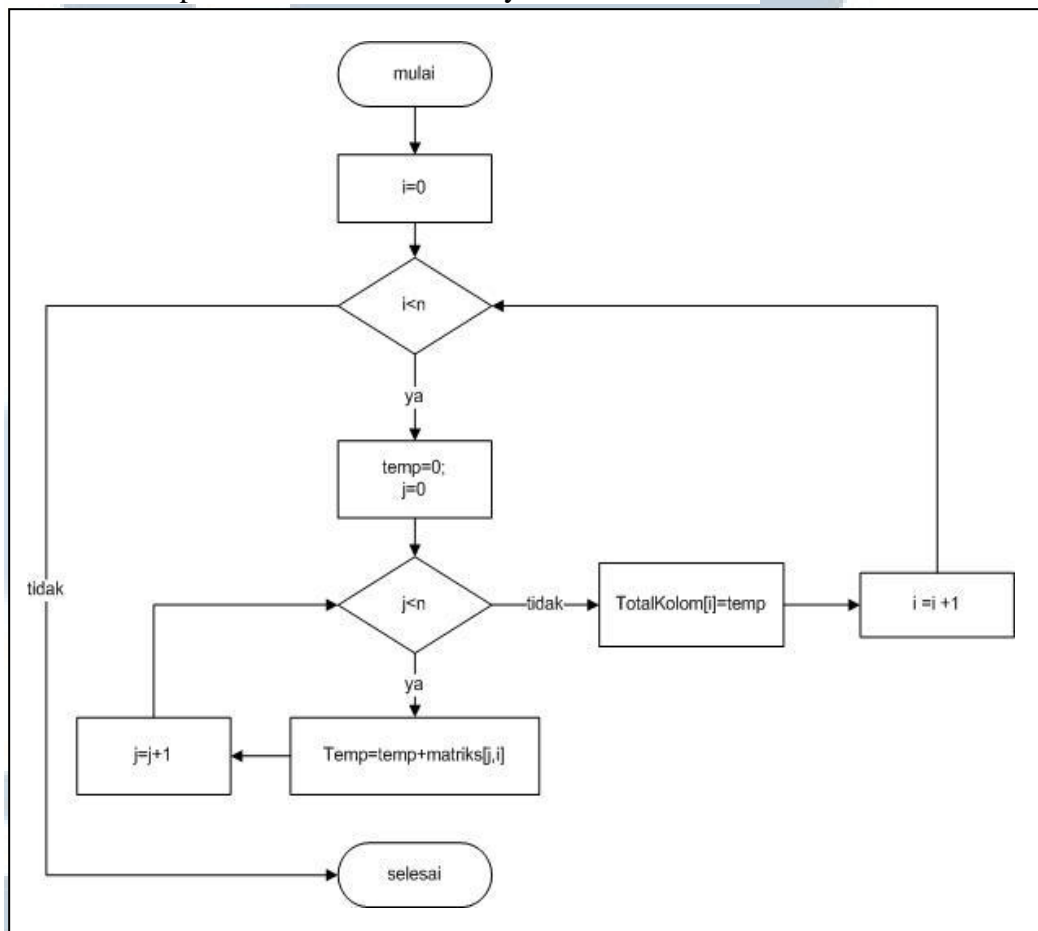


Gambar 3.8 *Flowchart* Analisis Kriteria Penilaian

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.5 Flowchart Hitung Total Kolom Matriks

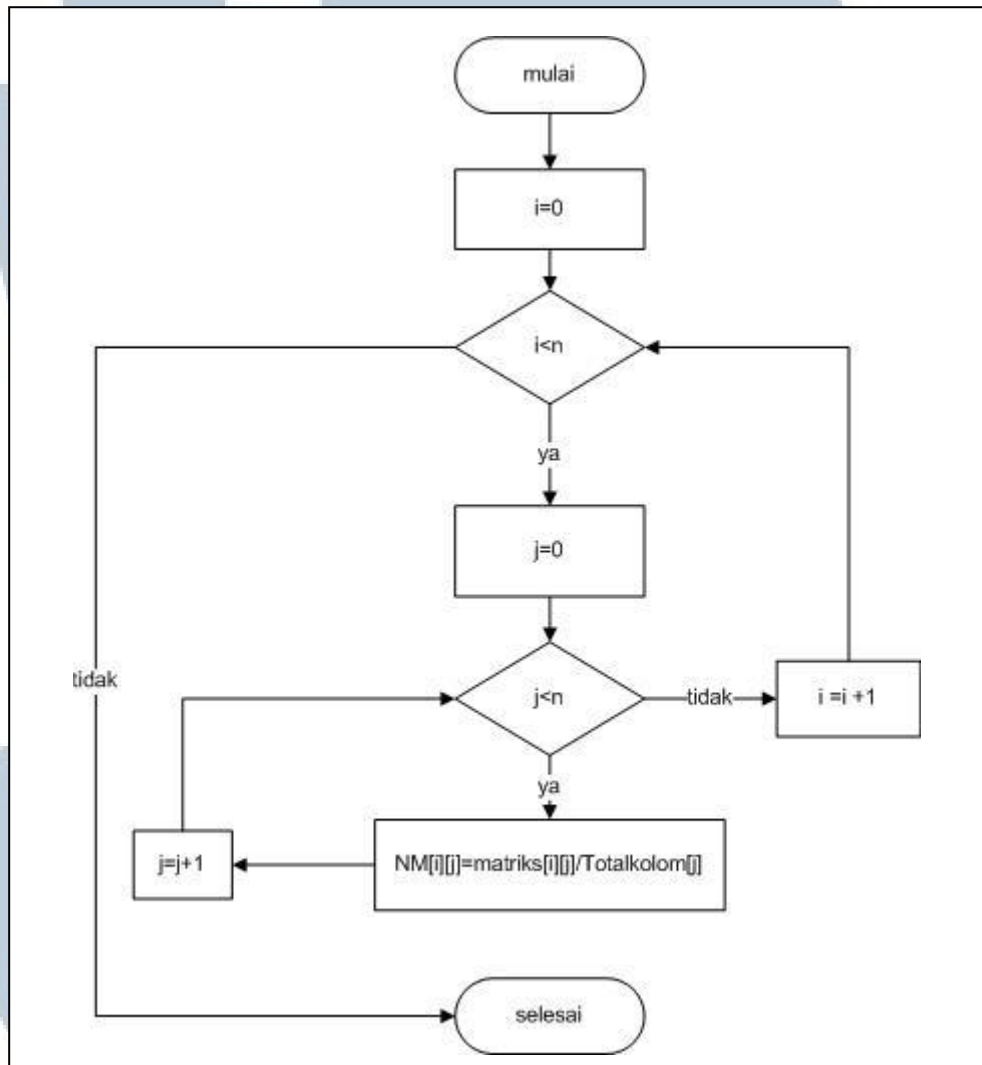
Flowchart ini menggambarkan proses penghitungan total kolom matriks yang bertujuan untuk penghitungan normalisasi matriks. Untuk melakukan normalisasi diperlukan penghitungan total kolom pada tiap matriks. Dimulai dari kolom pertama pada baris pertama sampai dengan baris akhir. Simpan variabel hasil penjumlahan tiap kolom pada tiap baris dengan nama temp. Bila penghitungan sudah sampai kolom pada baris akhir, simpan nilai variabel temp tersebut pada TotalKolom[i]. Lanjutkan sampai kolom akhir, maka akan diperoleh total kolom sebanyak ukuran matriks.



Gambar 3.9 *Flowchart* Hitung Total Kolom Matriks

3.2.6 Flowchart Normalisasi Matriks

Flowchart ini menggambarkan proses normalisasi matriks. Pada matriks[baris,kolom] dibagi dengan totalkolom[kolom] sehingga diperoleh matriks yang dinormalisasi.

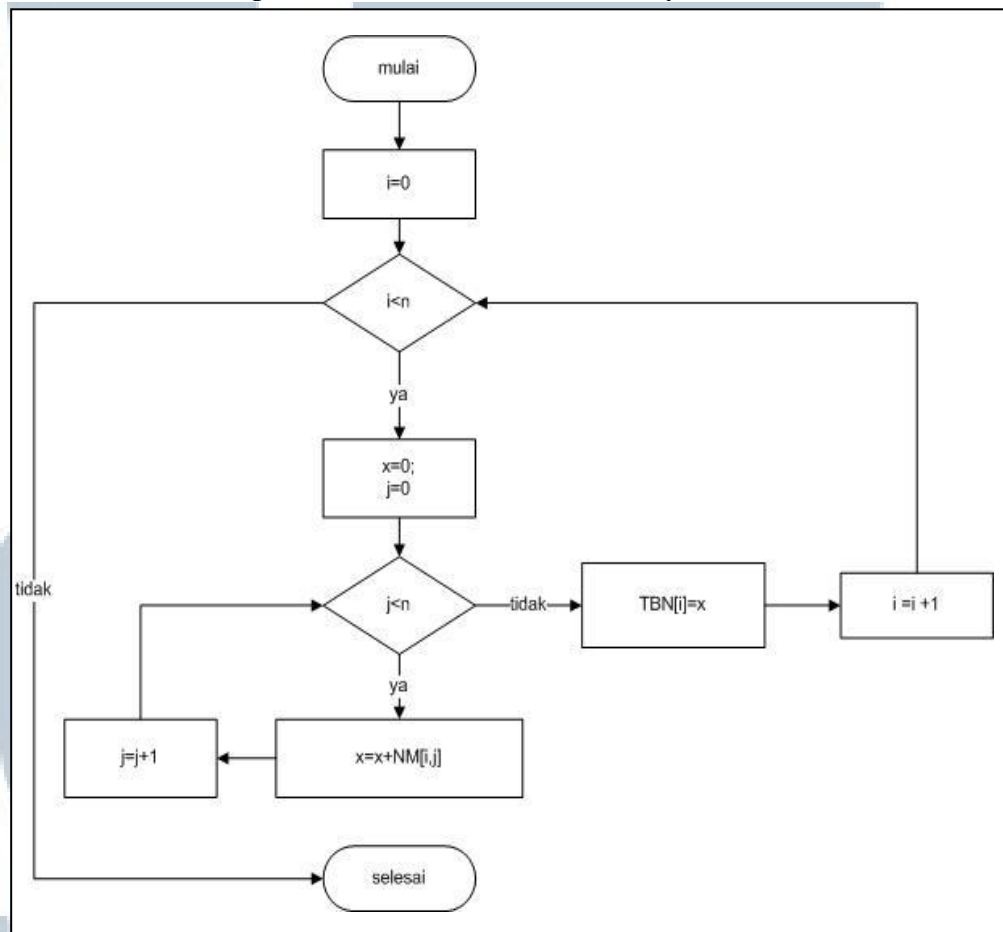


Gambar 3.10 *Flowchart* Normalisasi Matriks

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.7 Flowchart Hitung Total Baris Normalisasi

Flowchart ini menggambarkan proses total baris normalisasi matriks. Dimulai dari kolom pertama pada baris pertama sampai dengan kolom akhir. Simpan variabel hasil penjumlahan tiap kolom pada tiap baris dengan nama x . Bila penghitungan sudah sampai kolom akhir pada tiap baris, simpan nilai variabel x tersebut pada total baris normal $TBN[i]$. Lanjutkan sampai baris akhir, maka akan diperoleh total baris normal sebanyak ukuran matriks.

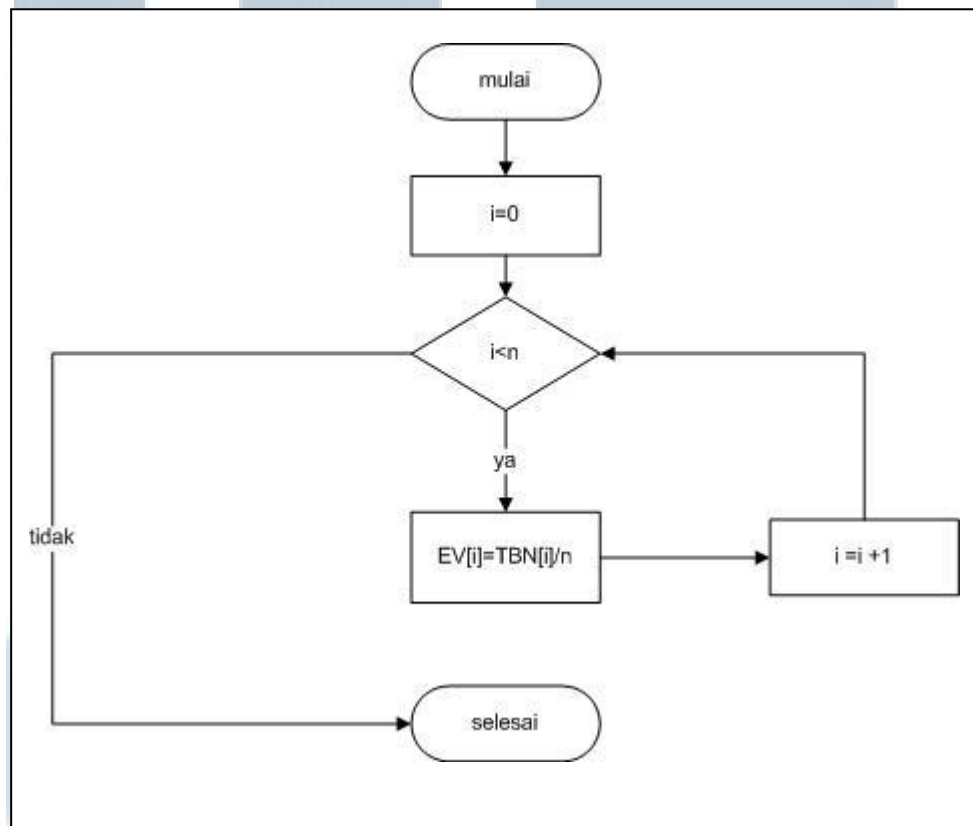


Gambar 3.11 *Flowchart* Hitung Total Baris Normalisasi

MULTIMEDIA
NUSANTARA

3.2.8 Flowchart Hitung Eigenvector

Flowchart ini menggambarkan proses hitung *eigenvector*/nilai prioritas. *Eigenvector* ini juga disebut *priority vector*. Cara memperoleh *eigenvector* tiap baris yaitu total baris normal yang didapatkan pada tiap baris dibagi dengan ukuran matriks.

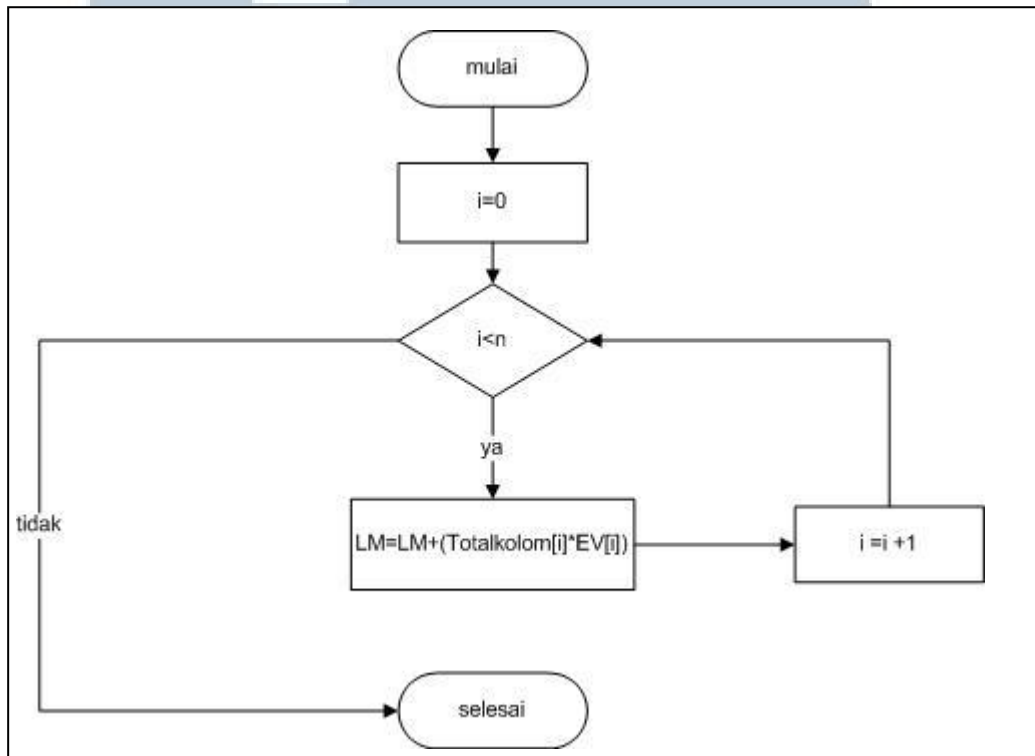


Gambar 3.12 *Flowchart* Hitung *Eigenvector*

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.9 Flowchart Hitung LambdaMax

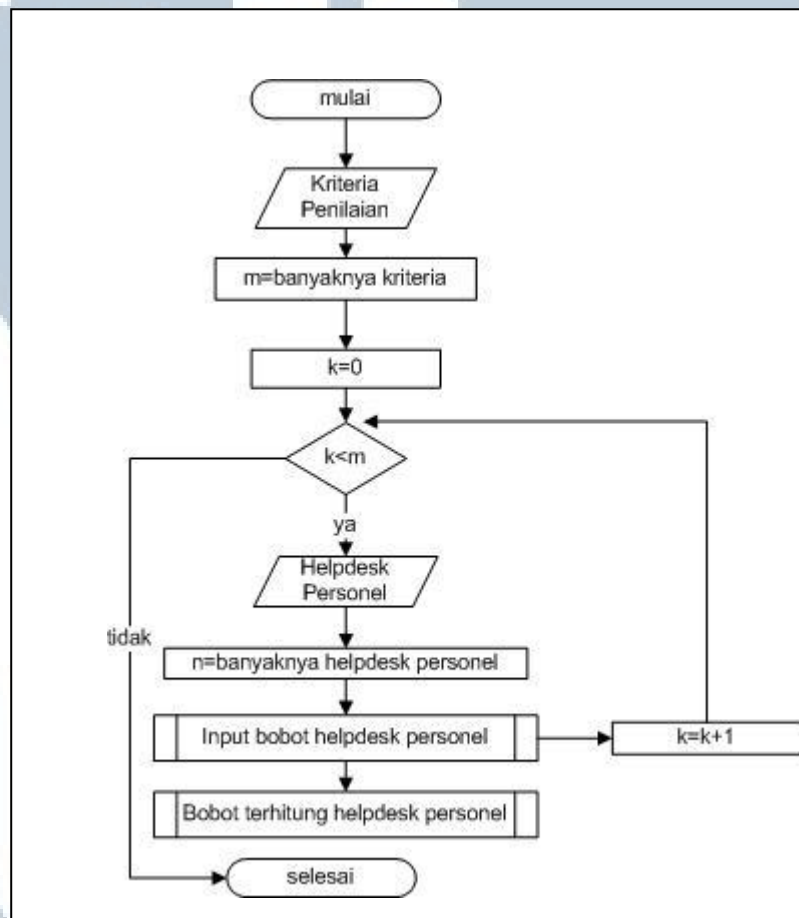
Flowchart ini menggambarkan proses hitung lambda maximum. Lambda maximum dihitung dengan cara total kolom pada tiap kolom matriks dikalikan dengan eigenvector pada tiap baris.



Gambar 3.13 *Flowchart* Hitung LambdaMax

3.2.10 Flowchart AHP Personel

Setelah mengetahui nilai prioritas tiap kriteria yang didapatkan dari proses AHP Kriteria, maka lanjutkan ke proses AHP Personel. Proses yang terdapat dalam AHP Personel ini yaitu input bobot personel tiap kriteria dan hitung nilai prioritas personel tiap kriteria.

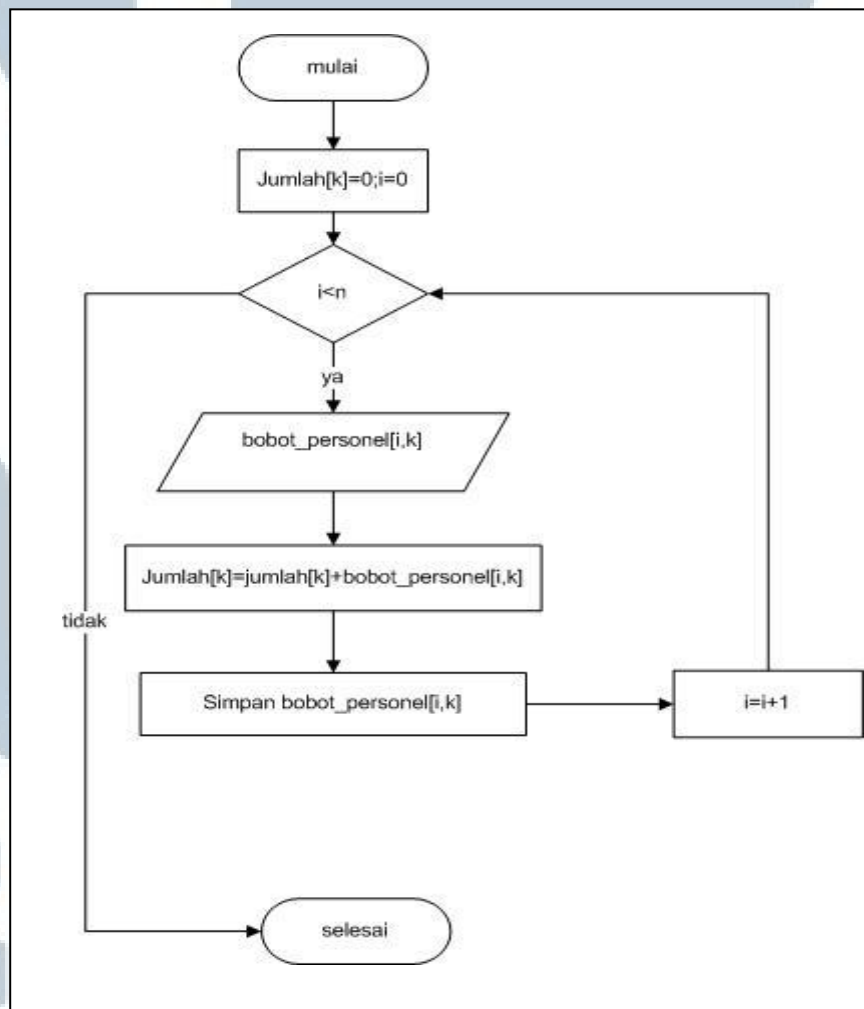


Gambar 3.14 *Flowchart* AHP Personel

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.11 Flowchart Input Bobot Helpdesk Personel

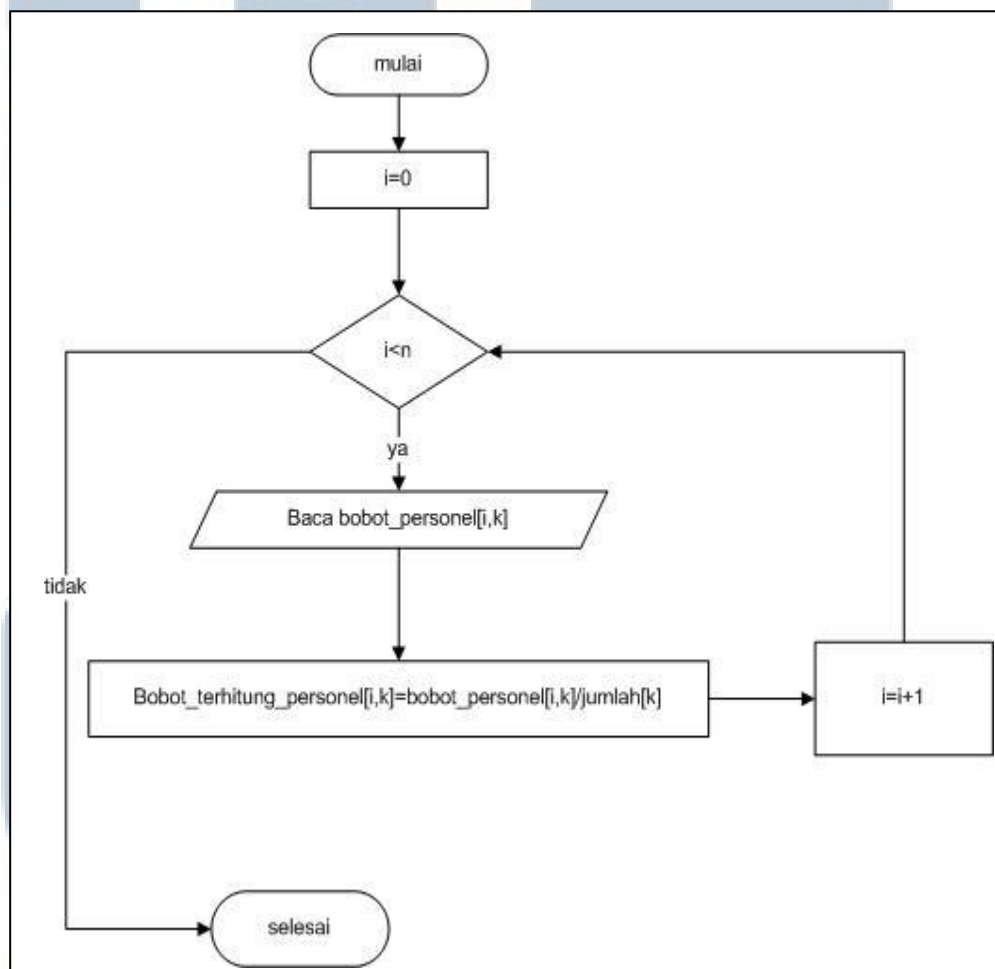
Flowchart di bawah ini menggambarkan proses memasukkan nilai personel tiap kriteria. Dimulai dari input bobot personel tiap kriteria. Jumlah[k] merupakan penghitungan untuk menjumlahkan bobot personel pada tiap kriteria. Setelah itu, bobot personel pada tiap kriteria disimpan. Apabila i sudah sampai jumlah personel, k bertambah 1 yang dapat dilihat pada Gambar 3.14 dan ulangi untuk input bobot *helpdesk* personel sampai dengan kriteria terakhir.



Gambar 3.15 *Flowchart* Input Bobot Helpdesk Personel

3.2.12 Flowchart Bobot Terhitung Helpdesk Personel

Setelah proses pemasukan nilai bobot personel tiap kriteria disimpan, kemudian dilakukan proses penghitungan nilai intensitas akhir. Rumus penghitungan nilai intensitas karyawan per kriteria ini adalah dengan melakukan pembagian antara bobot karyawan per kriteria dengan jumlah bobot karyawan per kriteria yang telah dimasukkan tersebut.

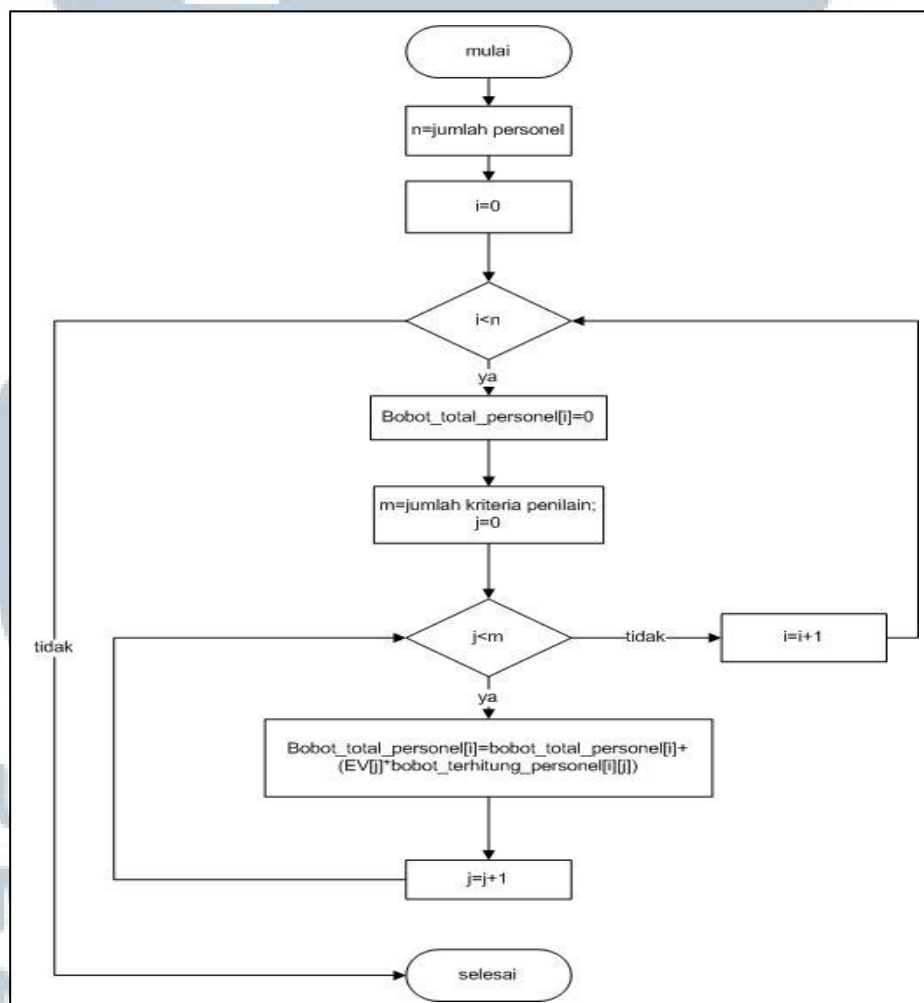


Gambar 3.16 Flowchart Bobot Terhitung Helpdesk Personel

UNIVERSITAS
MULTIMEDIA
NUSANTARA

3.2.13 Flowchart Hasil Analisis Penilaian

Flowchart di bawah ini menggambarkan proses hasil akhir nilai prioritas personel. Bobot total personel pada masing-masing personel yaitu bobot total personel sebelumnya ditambah dengan eigenvector tiap kriteria dikalikan bobot terhitung personel tiap kriteria yang telah diperoleh sebelumnya pada Gambar 3.16. Ulangi sampai dengan i kurang dari jumlah personel, sehingga hasil akhir bobot total personel akan diperoleh dan dapat diketahui personel mana yang lebih baik sesuai dengan penghitungan AHP yang berasal dari input *manager* tersebut.

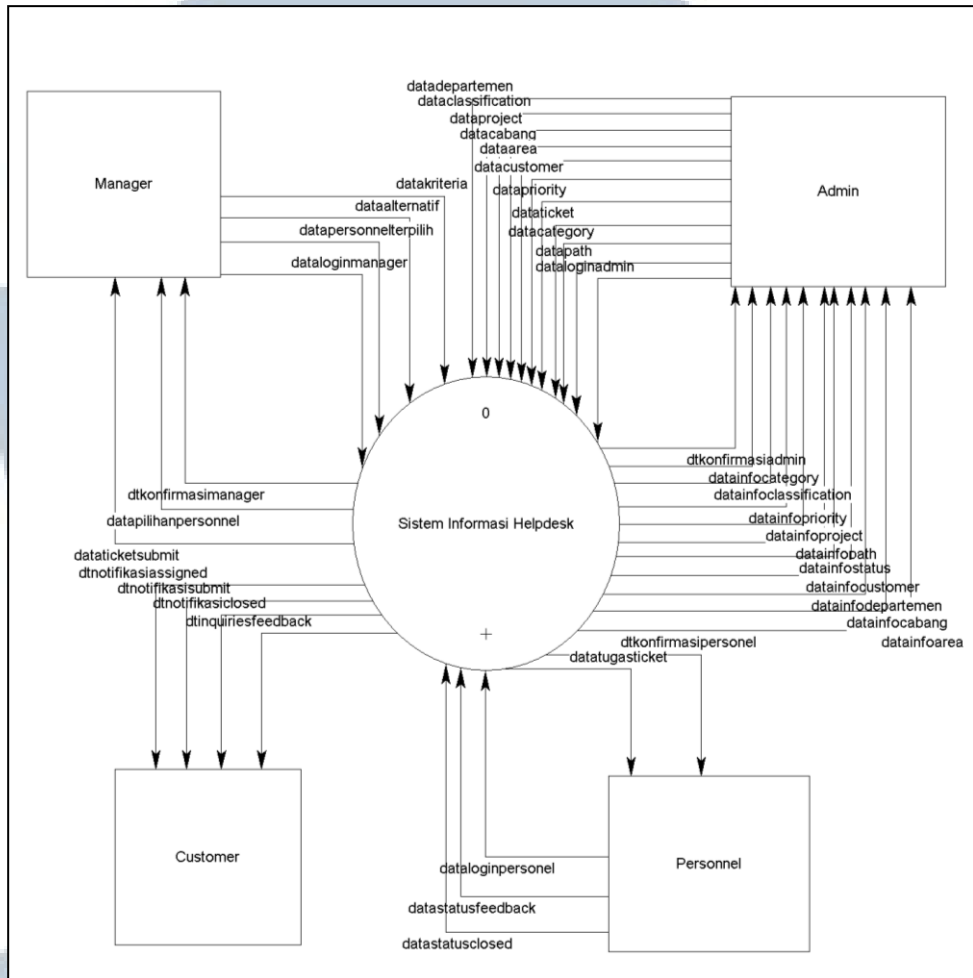


Gambar 3.17 *Flowchart* Hasil Analisis Penilaian

3.2.14 Data Flow Diagram

Context diagram pada Gambar 3.18 menggambarkan data yang masuk dan keluar dari keseluruhan aplikasi *helpdesk* ini. Rancangan DFD pada level 0 ini menjelaskan bahwa secara garis besar terdapat empat entitas yaitu *manager*, *admin*, *customer*, dan *personnel helpdesk*. *Manager*, *admin*, dan *personnel helpdesk* dapat mengakses sistem informasi *helpdesk* sedangkan *customer* tidak dapat mengakses sistem informasi *helpdesk*. Entitas *manager* dapat melakukan pengalokasian personel *helpdesk* untuk mengerjakan tiket. Entitas *admin* dapat melakukan pengisian data tiket. Entitas *personnel* dapat melakukan perubahan status untuk menjadi *closed/feedback*. Entitas *customer* hanya menerima notifikasi dari setiap perubahan status tiket yang ada melalui *email*.





Gambar 3.18 Context Diagram

UNIVERSITAS
MULTIMEDIA
NUSANTARA

Pada DFD level 1, penulis menggambarkan terdapat tujuh proses, yaitu validasi *user* yang dilakukan pada saat melakukan *login*. Admin, *manager*, dan personel *helpdesk* mengisi *username* dan *password*. Lalu, aplikasi akan mengecek apakah sesuai atau tidak. Jika sesuai, maka dapat masuk ke dalam aplikasi. Admin dapat melakukan *maintain data*, yaitu melakukan perubahan data. Kemudian, admin juga dapat membuat tiket, dimulai dengan menginput ISR, lalu tiket di-*submit*. Saat tiket di-*submit*, dilakukan penghitungan AHP untuk mengetahui hasil alternatif dari bobot kriteria dan alternatif yang telah diinput oleh *manager*. Setelah itu, dijalankan notifikasi status *submit* yaitu berupa *email* yang akan dikirimkan ke *manager* dan *customer*. Manager pun melakukan pemilihan personel *helpdesk* yang sudah tersedia dari hasil hitung AHP. Lalu, dijalankan notifikasi status *assign*, yaitu personel *helpdesk* yang telah di-*assign* dan *customer* akan mendapat notifikasi *email* bahwa tiket telah di-*assign*. Setelah itu, personel *helpdesk* akan melakukan analisis, apabila ia merasa kurang jelas terhadap apa yang dikeluhkan, maka ia akan mengubah status tiket menjadi *feedback* dan mengirimkan *email* ke *customer* tersebut. Namun, apabila sudah merasa jelas dan ISR tersebut diselesaikan, maka personel *helpdesk* akan mengubah status tiket tersebut menjadi *closed*. Lalu, notifikasi *email* akan dikirimkan ke *customer* dan *manager*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3.2.15 Entity Relationship Diagram

Dalam rancangan ERD ini terdapat entitas area, cabang, kriteria, departemen, *classification*, *project*, *priority*, *customer*, *ticket*, eigenvaluekriteria, alternatif, eigenvaluealternatif, file, job, helpdesk_staff, *login*, *category*, hasilalternatif, historicalheader, historicaldetail, dan status.

Pada entitas area ini dapat dimiliki oleh banyak entitas cabang. Banyak *customer* yang bekerja pada departemen dan cabang yang sama. Banyak *ticket* yang memiliki *project*, *priority*, *category*, *classification*, dan *customer* yang sama. Banyak *ticket* dapat dikerjakan oleh entitas helpdesk_staff. Kriteria diambil berdasarkan *classification*. Eigenvaluekriteria berhubungan erat dengan *classification*. Banyak alternatif yang dihasilkan *classification*. Alternatif itu sendiri menghasilkan banyak eigenvaluealternatif untuk penghitungan AHP tersebut. Alternatif berasal dari helpdesk_staff. Tiap *ticket* menyimpan *historicalheader* yang menghasilkan banyak *historicaldetail*. Status dimiliki oleh tiap *historicaldetail*. Helpdesk_staff dapat mengakses aplikasi helpdesk yang berhubungan dengan *login*. Hasilalternatif itu merupakan helpdesk_staff yang telah diurutkan sesuai penghitungan AHP. Tiap *ticket* menghasilkan banyak hasilalternatif.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

3.2.16 Struktur Tabel

Aplikasi helpdesk ini memiliki dua puluh dua tabel yaitu

Nama Tabel : area

Deskripsi : nama area di mana saja perusahaan berlokasi

Primary Key : area_id

Foreign Key : -

Tabel 3.1 Struktur Tabel Area

No	Nama Kolom	Tipe	Panjang	Keterangan
1	area_id	Varchar	10	Id Area
2	area_desc	Varchar	50	Deskripsi area

Nama Tabel : cabang

Deskripsi : nama cabang yang dimiliki oleh perusahaan serta lokasinya

Primary Key : cab_id

Foreign Key : area_id

Tabel 3.2 Struktur Tabel Cabang

No	Nama Kolom	Tipe	Panjang	Keterangan
1	cab_id	Varchar	10	Id cabang
2	cab_name	Varchar	50	Nama cabang
3	area_id	Varchar	10	Id Area

Nama Tabel : classification

Deskripsi : nama klasifikasi masalah yang dihadapi *customer*

Primary Key : classification_id

Foreign Key : -

Tabel 3.3 Struktur Tabel Classification

No	Nama Kolom	Tipe	Panjang	Keterangan
1	classification_id	Varchar	5	Id classification
2	classification_desc	Varchar	50	Deskripsi classification

Nama Tabel : customer

Deskripsi : nama *customer* yang mengalami masalah

Primary Key : cust_id

Foreign Key : dept_id, cab_id

Tabel 3.4 Struktur Tabel Customer

No	Nama Kolom	Tipe	Panjang	Keterangan
1	cust_id	Varchar	100	Id <i>customer</i>
2	cust_name	Varchar	50	Nama <i>customer</i>
3	cust_email	Varchar	50	<i>Email customer</i>
4	dept_id	Varchar	10	Id departemen
5	cab_id	Varchar	10	Id cabang

Nama Tabel : departemen

Deskripsi : nama departemen dari *customer* yang mengalami masalah

Primary Key : dept_id

Foreign Key : -

Tabel 3.5 Struktur Tabel Departemen

No	Nama Kolom	Tipe	Panjang	Keterangan
1	dept_id	Varchar	10	Id departemen
2	dept_desc	Varchar	50	Deskripsi departemen

Nama Tabel : job

Deskripsi: nama pekerjaan yang dapat mengakses aplikasi *helpdesk* yaitu admin, *manager*, *personnel* atau *undefined*

Primary Key: job_id

Foreign Key: -

Tabel 3.6 Struktur Tabel Job

No	Nama Kolom	Tipe	Panjang	Keterangan
1	job_id	Varchar	5	Id job
2	job_desc	Varchar	50	Deskripsi job

Nama Tabel : kriteria

Deskripsi: kriteria untuk penghitungan *analytical hierarchy process*

Primary Key: -

Foreign Key: classification1_id, classification2_id

Tabel 3.7 Struktur Tabel Kriteria

No	Nama Kolom	Tipe	Panjang	Keterangan
1	classification1_id	Varchar	5	Id kriteria 1
2	classification2_id	Varchar	5	Id kriteria 2
3	kriteria_bobot	Float		Bobot kriteria
4	valid_datetime	datetime		kapan waktu bobot tersebut berlaku

Nama Tabel : priority

Deskripsi: prioritas masalah yang dihadapi oleh *personnel helpdesk*

Primary Key: priority_id

Foreign Key: -

Tabel 3.8 Struktur Tabel *Priority*

No	Nama Kolom	Tipe	Panjang	Keterangan
1	priority_id	Varchar	10	Id priority
2	priority_desc	Varchar	50	Deskripsi priority

Nama Tabel : project

Deskripsi: tipe masalah *project*

Primary Key: project_id

Foreign Key: -

Tabel 3.9 Struktur Tabel Project

No	Nama Kolom	Tipe	Panjang	Keterangan
1	project_id	Varchar	10	Id project
2	project_name	Varchar	50	Deskripsi project

Nama Tabel : category

Deskripsi: kategori jenis *IT Service Request* yang dihadapi

Primary Key: category_id

Foreign Key: -

Tabel 3.10 Struktur Tabel Category

No	Nama Kolom	Tipe	Panjang	Keterangan
1	category_id	Varchar	5	Id category
2	category_name	Varchar	50	Nama category

Nama Tabel : status

Deskripsi: status dari tiket apakah sudah *feedback* atau *closed*

Primary Key: status_id

Foreign Key: -

Tabel 3.11 Struktur Tabel Status

No	Nama Kolom	Tipe	Panjang	Keterangan
1	status_id	Varchar	10	Id status
2	status_desc	Varchar	50	Deskripsi status

Nama Tabel : ticket

Deskripsi: *IT Service Request* yang telah diinput oleh admin ke aplikasi menjadi ticket untuk diproses selanjutnya

Primary Key: ticket_id

Foreign Key: *project_id*, *cust_id*, *user_id*, *priority_id*, *category_id*, *classification_id*, *ticket_file_id*

Tabel 3.12 Struktur Tabel Ticket

No	Nama Kolom	Tipe	Panjang	Keterangan
1	ticket_id	Varchar	255	Id ticket
2	project_id	Varchar	10	Id <i>project</i>
3	cust_id	Varchar	100	Id <i>customer</i>
4	ticket_issued_date	Datetime		Tanggal ticket diregister
5	user_id	Varchar	5	Id <i>user</i>
6	priority_id	Varchar	10	Id <i>priority</i>
7	category_id	Varchar	5	Id <i>category</i>
8	ticket_file_id	Integer	11	File yang diupload sebagai info tambahan
9	classification_id	Varchar	10	Id <i>classification</i>
10	ticket_summary	Varchar	50	Garis besar masalah yang dihadapi
11	ticket_description	Varchar	500	Keterangan lebih lanjut atas masalah yang dihadapi

Nama Tabel : helpdesk_staff

Deskripsi: siapa saja yang dapat mengakses aplikasi *helpdesk* ini

Primary Key: *user_id*

Foreign Key: *job_id*

Tabel 3.13 Struktur Tabel Helpdesk_staff

No	Nama Kolom	Tipe	Panjang	Keterangan
1	user_id	Varchar	5	Id <i>user</i>
2	User_completename	Varchar	50	Nama <i>user</i>
3	createdDate	datetime		<i>Email user</i>
4	user_email	Varchar	64	<i>email user</i>
5	job_id	varchar	5	Id <i>job</i>
6	Activation	Varchar	40	Aktivasi <i>email</i>

Nama Tabel : alternatif

Deskripsi: alternatif untuk penghitungan *analytical hierarchy process*

Primary Key: -

Foreign Key: *user_id*, *classification_id*

Tabel 3.14 Struktur Tabel Alternatif

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>user_id</i>	Varchar	5	Id User
2	alternatif_bobot	Float		Bobot alternatif
3	classification_id	Varchar	5	Id kriteria
4	Valid_datetime	Datetime		Tanggal berlakunya bobot

Nama Tabel : historicalHeader

Deskripsi: penyimpanan *personnel helpdesk* mana yang akan menangani *ticket*

Primary Key: historical_id

Foreign Key: ticket_id

Tabel 3.15 Struktur Tabel HistoricalHeader

No	Nama Kolom	Tipe	Panjang	Keterangan
1	historical_id	Varchar	255	Id historical
2	<i>user_id</i>	Varchar	5	Id User
3	ticket_id	Varchar	255	Id ticket
4	startdate_assigned	Datetime		Tanggal mulai ticket diassigned ke <i>personnel helpdesk</i>

Nama Tabel : historicalDetail

Deskripsi: penyimpanan perubahan status tiket yang terjadi

Primary Key: -

Foreign Key: historical_id, status_id

Tabel 3.16 Struktur Tabel HistoricalDetail

No	Nama Kolom	Tipe	Panjang	Keterangan
1	historical_id	Varchar	255	Id historical
2	Date	Datetime		Tanggal terjadi perubahan status
3	status_id	Varchar	5	Id status
4	historical_desc	Varchar	100	Keterangan terjadi perubahan status

Nama Tabel : eigenvaluealternatif

Deskripsi : prioritas alternatif yang telah dihitung menggunakan AHP

Primary Key : -

Foreign Key : *classification_id, user_id*

Tabel 3.17 Struktur Tabel Eigenvaluealternatif

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>eigenvalue</i>	Float		Prioritas bobot
2	<i>classification_id</i>	Varchar	5	Id <i>Classification</i>
3	<i>user_id</i>	Varchar	5	Id <i>User</i>
4	<i>valid_datetime</i>	Datetime		Tanggal berlakunya bobot alternatif

Nama Tabel : *eigenvaluekriteria*

Deskripsi : tabel untuk menyimpan prioritas kriteria

Primary Key : -

Foreign Key : *classification_id*

Tabel 3.18 Struktur Tabel Eigenvaluekriteria

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>classification_id</i>	Varchar	5	Id <i>Classification</i>
2	<i>eigen</i>	Float		Bobot prioritas kriteria

Nama Tabel : *file*

Deskripsi : file penyimpanan dari ISR yang diupload

Primary Key : *ticket_file_id*

Foreign Key : -

Tabel 3.19 Struktur Tabel File

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>Ticket_file_id</i>	Varchar	255	Id file ticket
2	<i>Name</i>	Varchar	30	Nama file
3	<i>type</i>	Varchar	30	Type file
4	<i>size</i>	Integer	11	Besarnya ukuran file

Nama Tabel : *hasilalternatif*

Deskripsi : hasil alternatif urutan personel

Primary Key : -

Foreign Key : *user_id, ticket_id*

Tabel 3.20 Struktur Tabel hasilalternatif

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>bobot</i>	Float		Bobot hasil akhir perhitungan
2	<i>ticket_id</i>	Varchar	255	Id Ticket
3	<i>user_id</i>	Integer	5	Id User

Nama Tabel : *login*

Deskripsi : data *login*

Primary Key : -

Foreign Key : *user_id*

Tabel 3.21 Struktur Tabel Login

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>User_id</i>	Varchar	5	Id user
2	<i>Username</i>	Varchar	20	<i>username</i>
3	<i>user_password</i>	Varchar	32	<i>password User</i>

Nama Tabel : *path*

Deskripsi : lokasi folder penyimpanan file

Primary Key : -

Foreign Key : *path*

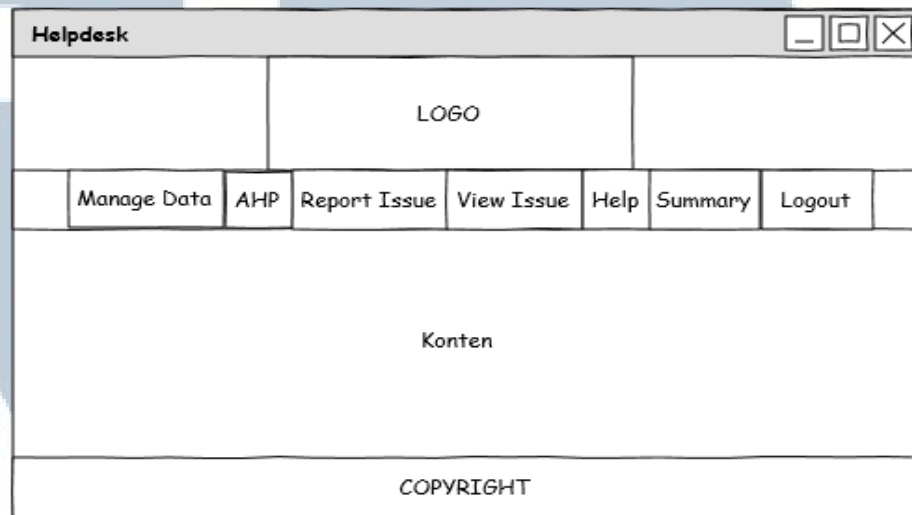
Tabel 3.22 Struktur Tabel *path*

No	Nama Kolom	Tipe	Panjang	Keterangan
1	<i>Path</i>	Varchar	100	Lokasi penyimpanan file

3.2.17 Perancangan Antarmuka

Proses perancangan antarmuka pada penelitian ini bertujuan untuk mendefinisikan antarmuka yang nantinya akan ditampilkan pada sistem. Secara umum, tiap halaman memiliki empat komponen utama, yaitu *header* (logo), menu (*manage data*, *AHP*, *report issue*, *view issue*, *help*, *summary*, *logout*), konten, dan *footer* (*copyright*). Bagian *header*, menu, dan *footer* akan selalu

ditampilkan sama, sedangkan konten akan berubah sesuai dengan halaman konten yang dipilih oleh pengguna. Berikut ini adalah gambar halaman rancangan antarmuka.



Gambar 3.21 Halaman Rancangan Antarmuka

Menu yang tersedia yaitu *Manage Data*, *AHP*, *Report Issue*, *View Issue*, *Help*, *Summary*, dan *Logout*. Terdapat 13 submenu di *Manage Data*, dimulai dari *Change Password*, *User*, *Path*, *Area*, *Branch*, *Classification*, *Department*, *Job*, *Status*, *Category*, *Priority*, *Project*, dan *Customer*. Terdapat 3 submenu di *AHP* yaitu *Help AHP*, *Criteria*, dan *Alternative*.

3.3 Kebutuhan Perangkat Lunak dan Keras Implementasi

Berikut ini adalah spesifikasi perangkat keras yang digunakan dalam penelitian ini

1. Processor Intel Atom N450
2. RAM 1 GB
3. Hard disk 250 GB

Spesifikasi perangkat lunak yang digunakan adalah sebagai berikut

1. Sistem operasi Windows 7
2. WampServer 2.2
3. Browser Google Chrome
4. Text editor (Netbeans, Notepad++)

A large, light blue circular logo with a stylized white graphic inside, resembling a grid or a series of connected nodes.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA