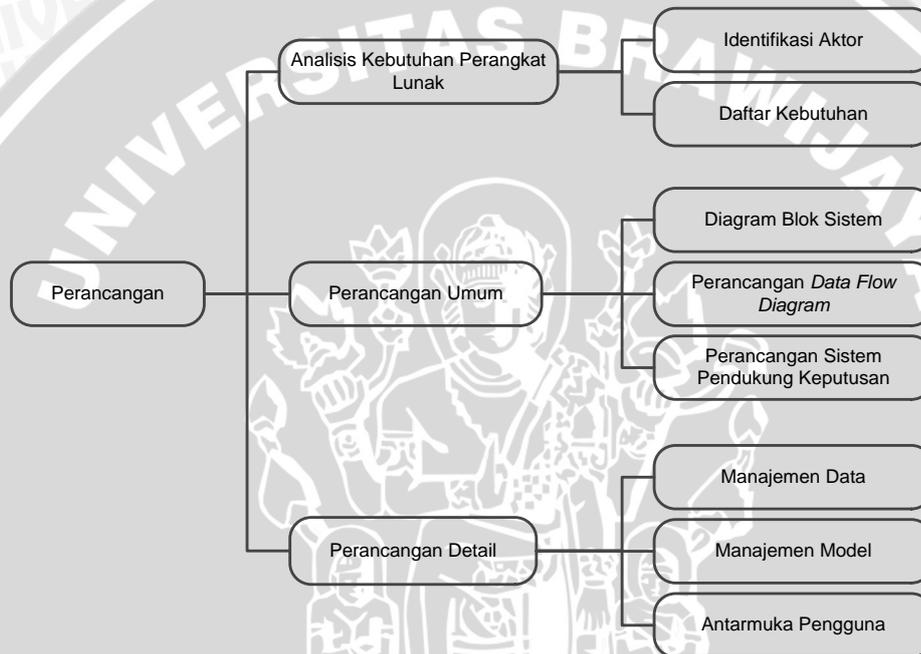


BAB IV PERANCANGAN

Bab ini menjelaskan perancangan “Sistem Pendukung Keputusan Pemilihan Alat Transportasi Umum Kota Malang Berdasar Jalur Terpendek dengan Menggunakan Algoritma *Floyd-Warshall*” sebagai produk akhir skripsi. Perancangan sistem pendukung keputusan ini terdiri dari tiga tahap yang dapat dilihat pada gambar pohon perancangan berikut ini:



Gambar 4.1 Pohon Perancangan
Sumber: Perancangan

4.1. Analisis Kebutuhan Perangkat Lunak

Pada analisis kebutuhan ini diawali dengan identifikasi perubahan aliran data serta fungsi-fungsi yang mengubahnya, penjabaran kebutuhan dan kemudian memodelkannya ke dalam suatu *data flow diagram*. Analisis kebutuhan ini ditujukan untuk menggambarkan kebutuhan-kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1. Identifikasi Aktor

Tahap ini mempunyai tujuan untuk melakukan identifikasi terhadap aktor yang dibutuhkan untuk proses yang terjadi dalam sistem. **Tabel 4.1**

memperlihatkan aktor beserta penjelasannya yang merupakan hasil dari proses identifikasi aktor.

Tabel 4.1 Identifikasi Aktor

Aktor	Deskripsi Aktor
<i>User</i>	<i>User</i> merupakan aktor pengguna dari sistem pendukung keputusan pemilihan transportasi umum Kota Malang. Aktor ini terdiri dari masyarakat luas, khususnya para turis yang berkunjung ke malang maupun masyarakat pendatang, yang dapat menggunakan sistem ini secara bebas tanpa login.
<i>Admin</i>	<i>Admin</i> merupakan aktor yang bertugas untuk melakukan <i>maintenance</i> sistem dan update data melalui sistem maupun secara langsung di dalam database.

Sumber: Perancangan

4.1.2 Daftar Kebutuhan

Daftar kebutuhan ini terdiri dari sebuah kolom yang menguraikan kebutuhan sistem maupun *interface*, dan pada kolom yang lain akan menunjukkan nama proses yang akan menunjukkan fungsionalitas masing-masing kebutuhan tersebut. Daftar kebutuhan fungsional dan non fungsional keseluruhan sistem ditunjukkan pada **Tabel 4.2**.

Tabel 4.2 Daftar Kebutuhan Fungsional

ID	Requirements	Terminator/Entitas	Nama Aliran Data
SRS_001_01	Sistem mampu melakukan perhitungan jarak terpendek dengan metode <i>Floyd-Warshall</i> dari data yang diinputkan <i>user</i> .	Sistem Pendukung Keputusan.	
SRS_001_02	Sistem mampu memproses data angkutan kota yang sesuai dengan jalur yang telah diproses.	Sistem Pendukung Keputusan.	
SRS_001_03	Sistem mampu melakukan perhitungan	Sistem Pendukung Keputusan.	

	estimasi waktu sesuai dengan jenis transportasi dan jarak dari jalur yang telah diproses.		
SRS_001_04	Sistem mampu melakukan perhitungan tarif sesuai dengan jenis transportasi dan jarak dari jalur yang telah diproses.	Sistem Pendukung Keputusan.	
IRS_100_01	Sistem mampu menerima input data nama jalan atau bangunan oleh <i>user</i> .	<i>User</i> , Sistem Pendukung Keputusan.	Input Lokasi Awal, Input Lokasi Tujuan.
IRS_100_02	Sistem mampu menampilkan jarak terpendek yang telah diproses.	Sistem Pendukung Keputusan, <i>User</i> .	Hasil Jarak Terpendek.
IRS_100_03	Sistem mampu menampilkan nama jalur yang telah diproses.	Sistem Pendukung Keputusan, <i>User</i> .	Hasil Rute yang Dilewati.
IRS_100_04	Sistem mampu menampilkan data transportasi yang sesuai dengan jalur yang telah diproses.	Sistem Pendukung Keputusan, <i>User</i> .	Hasil Transportasi yang Sesuai.
IRS_100_05	Sistem mampu menampilkan estimasi waktu dari tiap jenis	Sistem Pendukung Keputusan, <i>User</i> .	Hasil Estimasi Waktu.

	transportasi.		
IRS_100_06	Sistem mampu menampilkan perkiraan tarif dari tiap jenis transportasi.	Sistem Pendukung Keputusan, <i>User</i> .	Hasil Perkiraan Tarif.
IRS_100_07	Sistem mampu menerima perubahan data dari <i>admin</i> .	<i>Admin</i> , Sistem Pendukung Keputusan.	Update Data.

Sumber: Perancangan

Daftar kebutuhan non-fungsional aplikasi sistem pendukung keputusan diperlihatkan pada **Tabel 4.3**.

Tabel 4.3 Daftar Kebutuhan Non-Fungsional

Parameter	Deskripsi Kebutuhan
<i>Usability</i>	Terkait dengan kemudahan penggunaan sistem atau perangkat lunak oleh <i>user</i> .
<i>Memory</i>	Aplikasi ini harus ringan dan tidak membutuhkan <i>memory</i> yang besar.

Sumber: Perancangan

4.2 Perancangan Umum

Perancangan perancangan umum yang akan dibahas meliputi diagram blok sistem, perancangan *data flow diagram*, dan perancangan SPK, serta penjelasan di setiap poinnya.

4.2.1 Diagram Blok Sistem

Secara umum diagram blok sistem secara keseluruhan digambarkan pada **Gambar 4.2** berikut:



Gambar 4.2 Diagram Blok Sistem
Sumber: Perancangan

- *User*
User atau pengguna dapat berinteraksi dengan pendukung keputusan dan mendapatkan informasi tentang pemilihan transportasi umum Kota Malang.

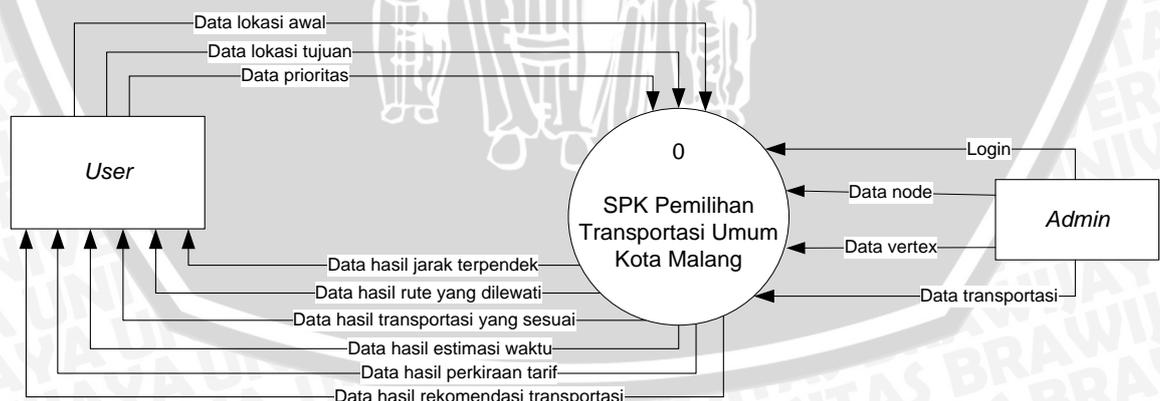
- SPK Pemilihan Transportasi Umum Kota Malang Berdasar Jalur Terpendek Berperan sebagai sistem yang yang membantu *user* dalam menentukan keputusan. Metode yang digunakan dalam sistem ini adalah perbandingan yang sebelumnya melalui proses pencarian terpendek dengan algoritma *Floyd-Warshall*.
- *Admin*
Admin bertugas untuk melakukan *maintenance* sistem dan update data melalui sistem maupun secara langsung di dalam *database*.

4.2.2 Perancangan Data Flow Diagram

Kebutuhan data maupun fungsional yang diperlukan oleh sistem digambarkan dalam diagram alir data. Secara keseluruhan sistem ini memiliki beberapa level dalam pemodelan diagram, diantaranya: *Data Flow Diagram* (DFD) *level 0* atau *Data Context Diagram* (DCD) dan DFD *level 1*.

4.2.2.1 Data Context Diagram

Context Diagram sebagai *top* diagram yang dapat memberikan gambaran umum dari sebuah sistem, diantaranya: pemodelan aliran-aliran masuk dan keluar baik yang berasal dari sistem maupun yang berasal dari entitas-entitas eksternal. Sedangkan masing-masing *level* pada DFD menggambarkan proses yang lebih rinci terhadap perubahan data dan fungsionalitas dalam sistem.



Gambar 4.3 DFD *Level 0* SPK

Sumber: Perancangan

Context Diagram merupakan gambaran masukan, proses dan keluaran pada sistem yang akan dibangun, baik dari sisi *admin* maupun *user*. *Admin*

memiliki *username* dan *password* sehingga memiliki hak akses yang berbeda dengan *user*. Dengan melakukan *login*, *admin* dapat melakukan *insert*, *update*, dan *delete* data melalui sistem. *Admin* juga dapat melakukan perubahan data secara langsung melalui database.

User tidak memiliki *username* maupun *password*, sehingga dapat secara bebas dan langsung menggunakan sistem. *User* diberikan hak akses untuk melakukan proses penginputan data lokasi awal dan lokasi yang akan dituju. Dari data yang dimasukkan oleh *user* maka sistem akan memproses jarak terpendek dan rute yang akan dilewati dengan menggunakan *Floyd-Warshall*. Kemudian data rute tersebut akan diproses dengan data trayek angkot dan didapatkan trayek angkot yang sesuai. Setelah didapat data trayek yang sesuai, akan dilakukan perhitungan estimasi waktu dan perkiraan tarif dari angkot, ojek, dan taksi, sesuai dengan rumus perhitungan masing-masing transportasi.

4.2.2.2 Data Flow Diagram Level 1

DFD *level 1* menggambarkan semua proses yang terjadi dalam sistem pendukung keputusan pemilihan transportasi umum Kota Malang, diantaranya adalah:

1. Proses Input Lokasi

Merupakan proses yang dilakukan oleh *user* yaitu memasukkan lokasi awal dan lokasi yang akan dituju oleh *user*. Lokasi dapat berupa nama jalan, maupun lokasi, tempat, atau bangunan. Lokasi tersebut nantinya akan disesuaikan dengan nama lokasi yang ada di dalam database.

2. Proses Penentuan Rute Terpendek

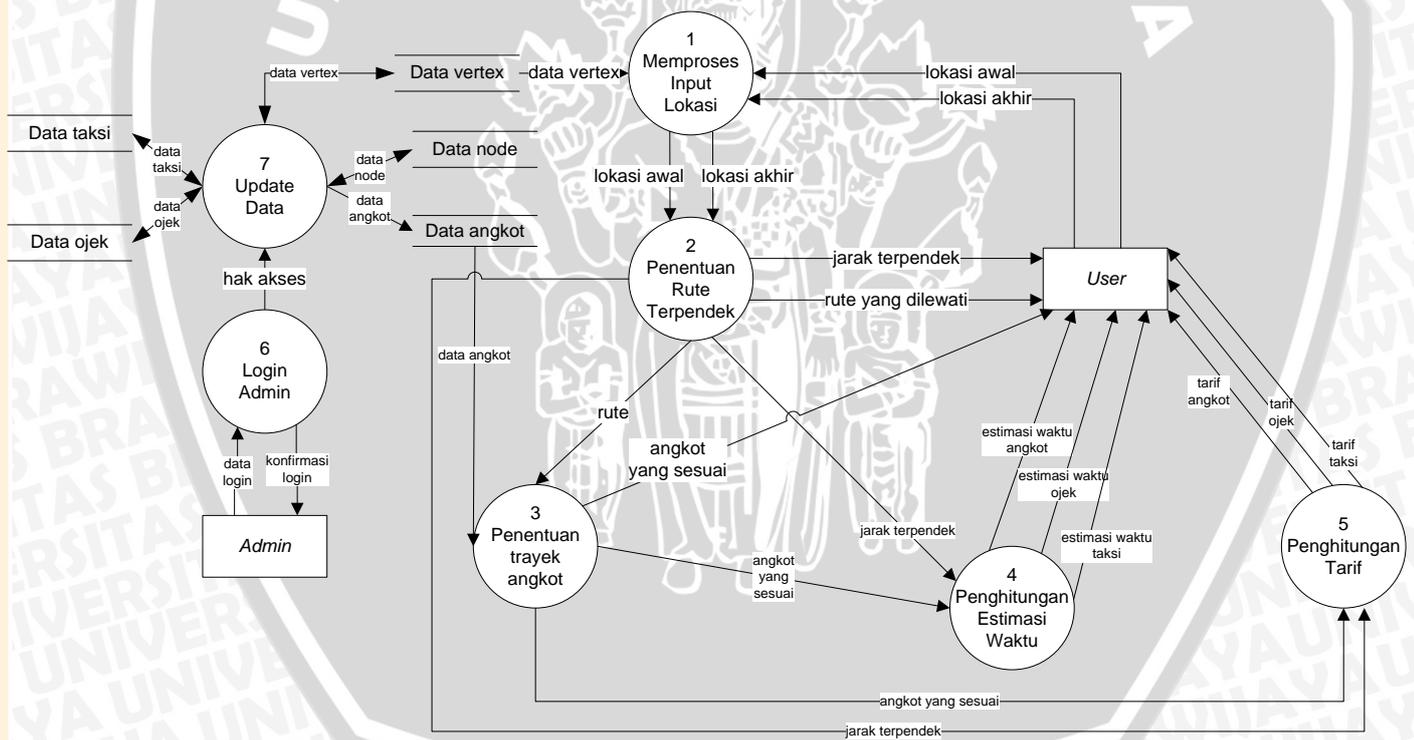
Merupakan proses penghitungan jarak dan diproses untuk menentukan jarak terpendek dengan menggunakan algoritma *Floyd-Warshall*. Setelah diperoleh jarak terpendek, maka sistem akan menampilkan rute-rute yang dilewati dari lokasi awal menuju lokasi tujuan.

3. Proses Penentuan Trayek Angkot

Setelah diperoleh jalur terpendek yang dilewati dari proses sebelumnya, maka akan dilakukan penyesuaian antara jalur tersebut dengan data trayek angkot. Setiap jalan yang dilewati oleh jalur terpendek dibandingkan satu per satu

dengan trayek angkot. Apabila ada angkot yang sama dengan jalan sebelum dan sesudahnya, maka angkot tersebut yang digunakan. Apabila ada angkot yang hanya sama dengan jalan sesudahnya, maka angkot tersebut yang digunakan. Apabila ada angkot yang sama sekali tidak sama dengan angkot sebelum dan sesudahnya, maka angkot tersebut yang diambil. Perbedaan jenis angkot yang diambil biasanya disebut dengan kegiatan oper. Setiap kali oper, tarif yang dibayarkan akan bertambah dan estimasi waktu yang diperlukan bertambah pula.

Sistem ini hanya melakukan pencarian trayek angkot yang sesuai dengan rute jalan karena hanya angkot yang memiliki rute yang sudah ditentukan oleh pemerintah. Untuk transportasi umum lain, yaitu taksi dan ojek, tidak memiliki trayek tertentu.



Gambar 4.3 DFD Level 1 SPK

Sumber: Perancangan

4. Proses Perhitungan Estimasi Waktu

Dari proses penentuan rute terpendek, maka didapatkan juga total panjang jalan yang akan dilewati. Jarak tersebut akan diproses dengan rumus yang ditentukan sehingga diperoleh estimasi waktu masing-masing jenis transportasi.

Kecepatan rata-rata tiap jenis transportasi berbeda sehingga diperoleh estimasi waktu yang berbeda pula. Untuk jenis transportasi angkot, jika ada kegiatan oper, maka estimasi waktu akan bertambah pula disamping perhitungan menurut kecepatan.

5. Proses Perhitungan Tarif

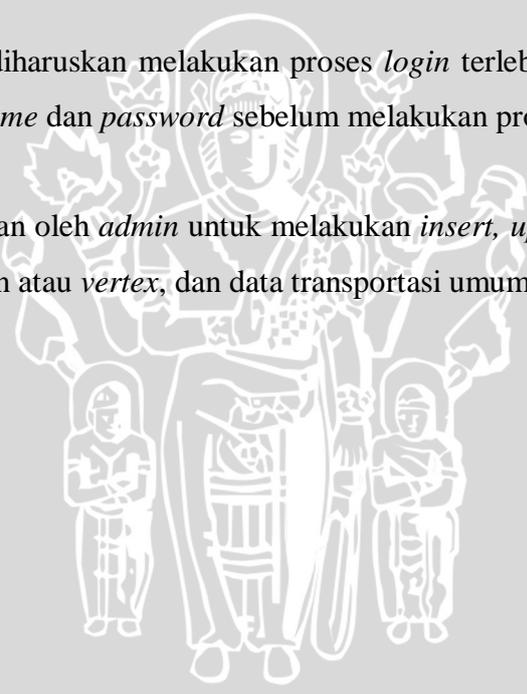
Dari proses penentuan rute terpendek, maka didapatkan juga total panjang jalan yang akan dilewati. Jarak tersebut akan diproses dengan rumus yang ditentukan sehingga diperoleh tarif masing-masing jenis transportasi. Tarif dasar dan rumus perhitungan tiap jenis transportasi berbeda sehingga diperoleh tarif total yang berbeda pula.

6. Proses Login Admin

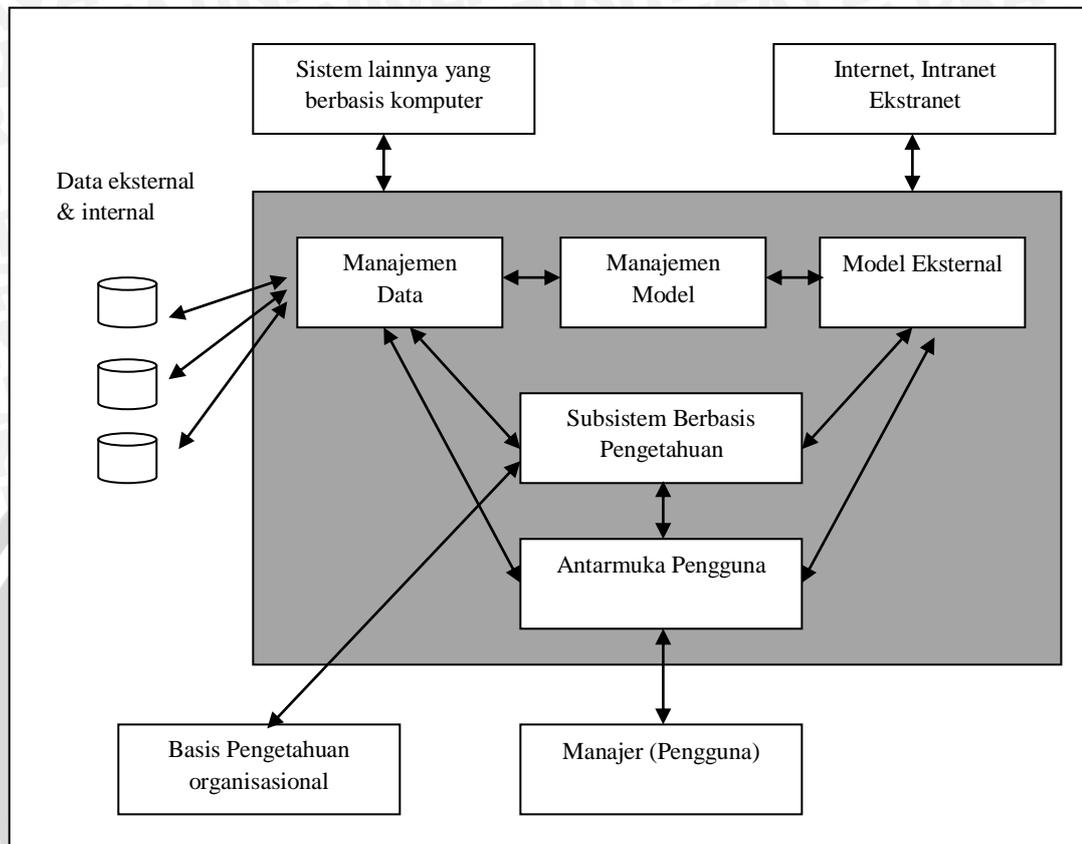
Admin atau *admin* diharuskan melakukan proses *login* terlebih dahulu dengan memasukkan *username* dan *password* sebelum melakukan proses update data.

7. Proses Update Data

Proses yang dilakukan oleh *admin* untuk melakukan *insert*, *update*, *delete* data, yaitu data *node*, jalan atau *vertex*, dan data transportasi umum.



4.2.3 Perancangan Sistem Pendukung Keputusan



Gambar 4.5 Arsitektur Sistem Pendukung Keputusan

Sumber: [KUS-07]

- **Manajemen Data**
Manajemen data, termasuk basis data, yang mengandung data berupa pengetahuan yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*. Dalam sistem ini DBMS yang digunakan yaitu *MySQL*. Pada perancangan basis data sistem ini menggunakan enam tabel yaitu tabel jalan, tabel tempat, tabel trayek, tabel angkot, tabel taksi, dan tabel ojek. Perancangan tabel *Entity Relationship Diagram* basis data sistem ini akan dibahas pada sub bab Perancangan Detail (Manajemen Data).
- **Manajemen Model**
Manajemen model, melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan. Pada sistem pendukung keputusan ini, pemodelan yang digunakan yaitu

pemodelan dengan metode *Floyd Warshall*. Permodelan dan metode yang digunakan dalam sistem ini akan dibahas pada sub bab Perancangan Detail (Manajemen Model).

- Model Eksternal

Model eksternal bersifat sebagai pendukung manajemen model. Pada sistem pendukung keputusan ini, tidak ada model eksternal yang digunakan karena hanya menggunakan pemodelan dengan metode *Floyd Warshall*.

- Subsistem Berbasis Pengetahuan

Subsistem ini mendukung semua subsistem lain atau bertindak langsung sebagai suatu komponen independen dan bersifat opsional. Pada sistem ini tidak mengikutsertakan subsistem berbasis pengetahuan sebagai bagian dari sistem pendukung keputusan.

- Antarmuka Pengguna

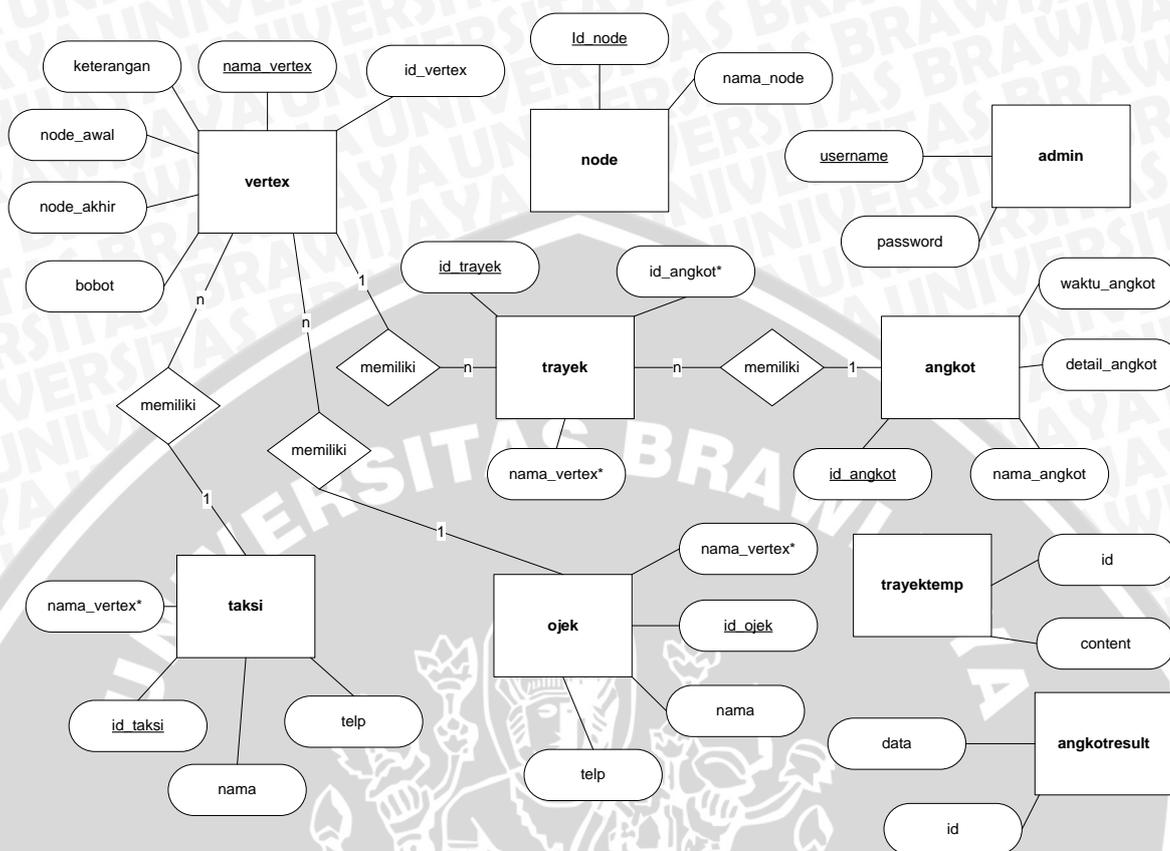
Pengguna berkomunikasi dengan memerintahkan sistem pendukung keputusan melalui subsistem antarmuka pengguna. Oleh karena itu, sistem harus menyediakan antarmuka pengguna. Perancangan antarmuka pengguna sistem akan dijelaskan pada sub bab Perancangan Detail (Antarmuka Pengguna).

4.3 Perancangan Detail

Perancangan detail merupakan perancangan sistem pendukung keputusan yang dibahas secara detail. Materi yang akan dibahas dalam subbab ini meliputi manajemen data, manajemen model, dan antarmuka pengguna, serta penjelasan di setiap poinnya.

4.3.1 Manajemen Data

Pada perancangan basis data sistem ini menggunakan enam tabel yaitu tabel jalan, tabel tempat, tabel trayek, tabel angkot, tabel taksi, dan tabel ojek. Adapun perancangan tabel *Entity Relationship Diagram* basis data sistem ini diperlihatkan pada **Gambar 4.6**.



Gambar 4.6 Entity Relationship Diagram SPK
Sumber: Perancangan

4.3.2 Manajemen Model

4.3.2.1 Gambaran Umum Manajemen Model

Sistem pendukung keputusan ini menggunakan algoritma pencarian jarak terpendek *Floyd-Warshall*. Algoritma ini menggunakan prinsip pemrograman dinamis. Ada beberapa tahap dalam memproses input dari *user* hingga menghasilkan keluaran. Tahapan tersebut antara lain:

1. Pencarian Rute Dan Jarak Terpendek Dengan Menggunakan Algoritma *Floyd-Warshall*

Setelah *user* memberikan masukan berupa data jalan atau tempat asal dan tujuan, maka secara gambaran manual akan dilakukan beberapa hal berikut:

- Peta asli ditandai dengan *node-node* antara lokasi asal sampai tujuan.
- *Node-node* dihubungkan dengan *vertex* dan diberi bobot sehingga membentuk graf.

- Graf diimplementasikan ke dalam bentuk *tree* sehingga lebih mudah diproses ke dalam rumus *Floyd-Warshall*.
- Penerapan rumus algoritma *Floyd-Warshall*

Basis:

$$f_1(s) = cx_1s \quad (4-1)$$

Rekurens:

$$f_k(s) = \min x_k \{cx_k s + f_{k-1}(x_k)\}, k = 2,3,4, \dots \quad (4-2)$$

Karena database sudah memuat data berupa pengetahuan semua *node* dan *vertex* serta bobotnya dari peta Kota Malang, maka akan secara otomatis semua *node* akan diproses menggunakan rumus algoritma *Floyd-Warshall* dari *node* awal sampai ditemukan *node* tujuan. Setelah melalui tahap tersebut, maka sistem akan menghasilkan rute dan jarak terpendek dari lokasi asal ke tujuan.

2. Pencarian Trayek Angkutan Kota yang Sesuai

Proses ini dilakukan dengan mencocokkan rute terpendek yang telah dihasilkan dari tahap sebelumnya dengan data trayek angkutan kota. Setiap *vertex* yang ada di rute terpendek akan memanggil data angkot yang melewati *vertex-vertex* tersebut satu persatu. Jika pada *vertex* tersebut merupakan *vertex* awal, maka akan diambil trayek yang kesamaannya besar dengan trayek-trayek di *vertex* setelahnya. Jika pada *vertex* tersebut merupakan *vertex* di tengah, maka akan diambil trayek yang kesamaannya besar dengan trayek-trayek di *vertex* sebelumnya dan setelahnya. Jika pada *vertex* tersebut merupakan *vertex* terakhir, maka akan diambil trayek yang kesamaannya besar dengan trayek-trayek di *vertex* sebelumnya. Jika pada *vertex* tersebut tidak memiliki trayek angkot, maka akan disarankan untuk berjalan kaki hanya jika panjang *vertex*nya kurang dari 500 meter Dengan begitu akan didapatkan kombinasi trayek yang kecil sehingga meminimalisir estimasi waktu dan tarif yang akan dikeluarkan.

3. Perhitungan Estimasi Waktu Setiap Jenis Transportasi

- Estimasi Waktu Angkutan Kota

$$ta = \frac{s}{va} + (\sum o \times 3) \quad (4-3)$$

ta = estimasi waktu angkot (menit)

s = panjang jalan (meter)

va = kecepatan rata-rata angkot (meter/menit) = 300 meter/menit (18Km/jam)

o = jumlah oper (satu kali oper diestimasikan membutuhkan waktu 3 menit)

- Estimasi Waktu Pejalan Kaki

$$tj = \frac{s}{vj} \quad (4-4)$$

tk = estimasi waktu jalan kaki (menit)

s = panjang jalan (meter)

vj = kecepatan rata-rata (meter/menit) = 100 meter/menit

- Estimasi Waktu Ojek

$$to = \frac{s}{vo} \quad (4-5)$$

to = estimasi waktu ojek (menit)

s = panjang jalan (Km)

vo = kecepatan rata-rata (meter/menit) = 500 meter/menit (30 Km/jam)

- Estimasi Waktu Taksi

$$tt = \frac{s}{vt} \quad (4-6)$$

tt = estimasi waktu taksi (menit)

s = panjang jalan (Km)

vt = kecepatan rata-rata (meter/menit) = 400 meter/menit (24 Km/jam)

4. Perhitungan Estimasi Tarif Setiap Jenis Transportasi

- Estimasi Tarif Angkutan Kota

$$ha = \sum a \times 2500 \quad (4-7)$$

ha = estimasi harga/tarif angkot (Rupiah)

a = jumlah angkot (satu kali naik angkot dikenakan tarif Rp 2.500,-)

- Estimasi Tarif Ojek

$$ho = \left(\frac{s-500}{500} \times 1000\right) + 4000 \quad (4-8)$$

ho = estimasi harga/tarif ojek (Rupiah)

s = panjang jalan (m) (setiap 500 meter dikenakan biaya Rp 1.000,-)

Tarif 500 meter pertama = Rp 4.000,-

- Estimasi Tarif Taksi

$$ht = \left(\frac{s-100}{100} \times 300\right) + 5000 \quad (4-9)$$

ht = estimasi harga/tarif taksi (Rupiah) (dengan total minimum Rp 20.000,-)

s = panjang jalan (m) (setiap 100 meter argo bertambah Rp300,-)

Tarif pertama buka pintu (*flag fall*) 100 meter pertama = Rp 6.000,-

5. Penentuan Transportasi Rekomendasi Berdasar Prioritas

Proses ini dilakukan dengan membandingkan tarif atau waktu masing-masing jenis transportasi sesuai dengan prioritas yang dipilih oleh pengguna. Sistem ini juga akan menampilkan semua hasil perhitungan tiap jenis transportasi sebagai pertimbangan. Pada halaman transportasi ojek dan taksi pun akan ditampilkan rekomendasi pemilik kendaraan dan kontakannya untuk keperluan *advertisement*.

Selain mendapatkan informasi hasil alat transportasi rekomendasi, *user* dapat pula mengetahui informasi estimasi tarif dan waktu alat transportasi lain sebagai pertimbangan jika tidak ingin menggunakan transportasi rekomendasi. Jadi, sistem pendukung keputusan ini sudah mencakup beberapa model analisa, yaitu *optimization analysis* untuk mencari jalur terpendek, dan *what-if analysis* untuk menentukan alat transportasi umum sesuai dengan kebutuhan dan keinginan pengguna.

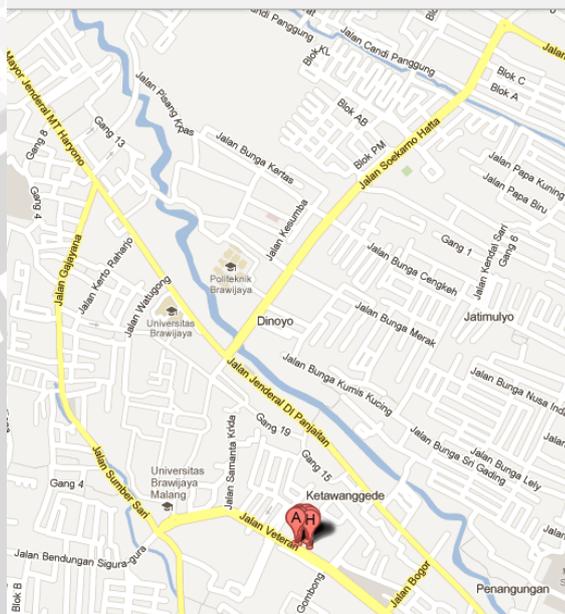
4.3.2.2 Contoh Implementasi Kasus

Berikut merupakan contoh dari perhitungan algoritma *Floyd-Warshall* yang telah diimplementasikan di dalam sistem. Pengguna dari monumen pesawat jalan Cokelat ingin menuju Malang Town Square. Maka prosesnya sebagai berikut:

Tahap 1:

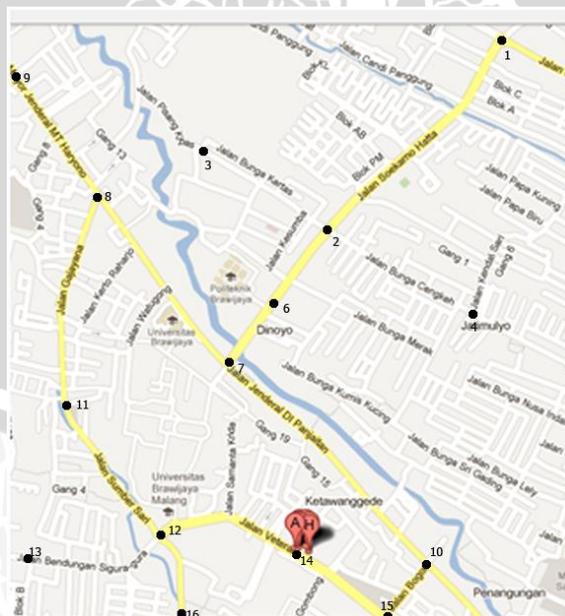
Pencarian Rute Dan Jarak Terpendek Dengan Menggunakan Algoritma *Floyd-Warshall*.

1. Peta Asli:



Gambar 4.7 Contoh Peta Malang
Sumber: map.google.com

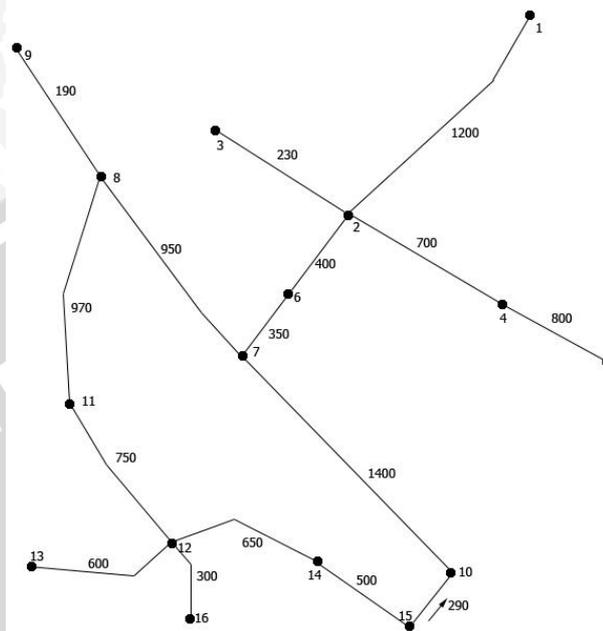
2. Diberikan *node* di setiap tikungan:



Gambar 4.8 Contoh Peta Malang dengan *Node*
Sumber: Perancangan

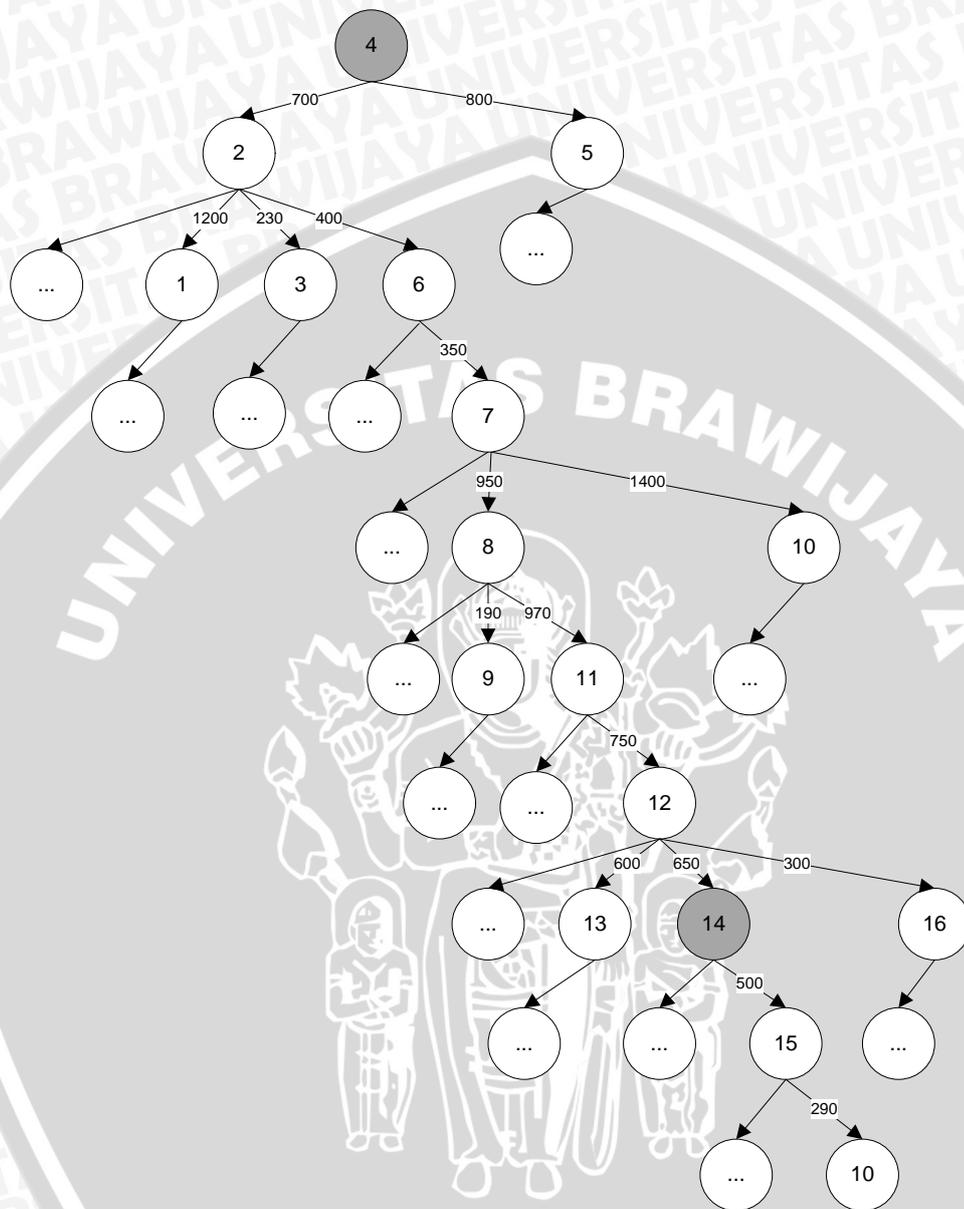


3. Membuat graf dengan *node*, *vertex*, serta bobotnya. *Node* asal adalah *node* 4. *Node* tujuan adalah *node* 14. Catatan: *node* 15 ke 10 adalah jalan searah.



Gambar 4.9 Graf Contoh Peta Malang
Sumber: Perancangan

4. Dari graf yang telah dibuat, diimplementasikan ke dalam *tree*:



Gambar 4.10 Implementasi *Tree*
Sumber: Perancangan

Keterangan:

(...) *node* sebelumnya karena jalan dua arah, terjadi *looping*.

5. Oleh sistem, *tree* yang telah dibuat dianggap sebagai tabel yang secara bertahap akan diproses menggunakan algoritma *Floyd-Warshall*.



Dari Gambar 4.11 dapat diperhatikan bahwa bobot 0 merupakan node yang tidak saling berhubungan.

Graf Awal

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	1200	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1200	0	230	700	0	400	0	0	0	0	0	0	0	0	0	0
3	0	230	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	700	0	0	800	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	800	0	0	0	0	0	0	0	0	0	0	0	0
6	0	400	0	0	0	0	350	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	350	0	950	0	1400	0	0	0	0	0	0
8	0	0	0	0	0	0	950	0	190	0	970	0	0	0	0	0
9	0	0	0	0	0	0	0	190	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	1400	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	970	0	0	0	750	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	750	0	600	650	0	300
13	0	0	0	0	0	0	0	0	0	0	0	600	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	650	0	0	500	0
15	0	0	0	0	0	0	0	0	0	290	0	0	0	500	0	0
16	0	0	0	0	0	0	0	0	0	0	0	300	0	0	0	0

Gambar 4.11 Implementasi Tabel Bobot Sebelum Proses *Floyd-Warshall*
Sumber: Perancangan

6. Penerapan algoritma *Floyd-Warshall*.

Basis:

$$f_1(s) = cx_1s$$

Rekursens:

$$f_k(s) = \min x_k \{cx_k s + f_{k-1}(x_k)\}, k = 2,3,4, \dots$$

Basis:

$$f_1(s) = cx_1s$$

s	Solusi optimum	
	$f_1(s)$	x_1
2	700	4
5	800	4



Rekurens:

Iterasi 1

$$f_2(s) = \min x_2 \{cx_2s + f_1(x_2)\}$$

s	x_2	$f_2(x_2, s) = cx_2s + f_1(x_2)$		Solusi Optimum	
		2	5	$f_2(s)$	x_2
1		1200	∞	1200	2
3		230	∞	230	2
6		1100	∞	1100	2

Iterasi 2

$$f_3(s) = \min x_3 \{cx_3s + f_2(x_3)\}$$

s	x_3	$f_3(x_3, s) = cx_3s + f_2(x_3)$			Solusi Optimum	
		1	3	6	$f_3(s)$	x_3
7		∞	∞	1450	1450	6

Iterasi 3

$$f_4(s) = \min x_4 \{cx_4s + f_3(x_4)\}$$

s	x_4	$f_4(x_4, s) = cx_4s + f_3(x_4)$		Solusi Optimum	
		7		$f_4(s)$	x_4
8		2400		2400	7
10		2850		2850	7

Iterasi 4

$$f_5(s) = \min x_5 \{cx_5s + f_4(x_5)\}$$

s	x_5	$f_5(x_5, s) = cx_5s + f_4(x_5)$		Solusi Optimum	
		8	10	$f_5(s)$	x_5
9		2590	∞	2590	8
11		3370	∞	3370	8

Iterasi 5

$$f_6(s) = \min x_6 \{cx_6s + f_5(x_6)\}$$

s	x_6	$f_6(x_6, s) = cx_6s + f_5(x_6)$		Solusi Optimum	
		9	11	$f_6(s)$	x_6
12		∞	4120	4120	11

Iterasi 6

$$f_7(s) = \min x_7 \{cx_7s + f_6(x_7)\}$$

s	x_7	$f_7(x_7, s) = cx_7s + f_6(x_7)$		Solusi Optimum	
		12		$f_7(s)$	x_7
13		4720		4720	12
14		4770		4770	12
16		4420		4420	12

Dari hasil pencarian jalur terpendek dari *node* 1 ke 14 menggunakan algoritma *Floyd-Warshall* (pemrograman dinamis), ditemukan bahwa jarak terpendek adalah 4.770 meter atau 4,77 Km dengan jalur (4-2-6-7-8-11-12-14).

Sistem akan menghitung semua jalur dengan algoritma *Floy-Warshall* sehingga akan terbentuk tabel bobot dan tabel hubungan antar *node*.



Tabel Bobot

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	0	1200	1430	1900	2700	1600	1950	2900	3090	3350	3870	4620	5220	5270	5770	4920
2	1200	0	230	700	1500	400	750	1700	1890	2150	2670	3420	4020	4070	4570	3720
3	1430	230	0	930	1730	630	980	1930	2120	2380	2900	3650	4250	4300	4800	3950
4	1900	700	930	0	800	1100	1450	2400	2590	2850	3370	4120	4720	4770	5270	4420
5	2700	1500	1730	800	0	1900	2250	3200	3390	3650	4170	4920	5520	5570	6070	5220
6	1600	400	630	1100	1900	0	350	1300	1490	1750	2270	3020	3620	3670	4170	3320
7	1950	750	980	1450	2250	350	0	950	1140	1400	1920	2670	3270	3320	3820	2970
8	2900	1700	1930	2400	3200	1300	950	0	190	2350	970	1720	2320	2370	2870	2020
9	3090	1890	2120	2590	3390	1490	1140	190	0	2540	1160	1910	2510	2560	3060	2210
10	3350	2150	2380	2850	3650	1750	1400	2350	2540	0	3320	4070	4670	4720	5220	4370
11	3870	2670	2900	3370	4170	2270	1920	970	1160	2190	0	750	1350	1400	1900	1050
12	4620	3420	3650	4120	4920	3020	2670	1720	1910	1440	750	0	600	650	1150	300
13	5220	4020	4250	4720	5520	3620	3270	2320	2510	2040	1350	600	0	1250	1750	900
14	4140	2940	3170	3640	4440	2540	2190	2370	2560	790	1400	650	1250	0	500	950
15	3640	2440	2670	3140	3940	2040	1690	2640	2830	290	1900	1150	1750	500	0	1450
16	4920	3720	3950	4420	5220	3320	2970	2020	2210	1740	1050	300	900	950	1450	0

Gambar 4.12 Implementasi Tabel Bobot Setelah Proses Floyd-Warshall
Sumber: Perancangan

Tabel Yang Dilalui

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

Gambar 4.13 Implementasi Tabel Hubungan Antar Node
Sumber: Perancangan

Tahap 2:

Pencarian Trayek Angkutan Kota Yang Sesuai.

Dari jalur (4-2-6-7-8-11-12-14) dicari trayek angkutan kota yang juga memuat jalur tersebut. Setelah diproses maka didapatkan:

1. Jalur (4-2) menggunakan angkot ABH2 (700m).
2. Jalur (2-6) menggunakan angkot ABH2/ASD1 (400m).
3. Jalur (6-7) menggunakan angkot ADL2 (350m).
4. Jalur (7-8) menggunakan angkot ADL1 (950m).
5. Jalur (7-10) menggunakan angkot AL2/HL2/LH1 (970m).
6. Jalur (10-11) menggunakan angkot AL2/ LH1 (750m).
7. Jalur (11-13) menggunakan angkot AL2/HL2 (650m).

Proses pengambilan angkot rekomendasi:

1. Jalur (4-2) tersedia ABH2 sehingga pasti menggunakan angkot ABH2.
2. Jalur (2-6) tersedia angkot ABH2/ASD1. Karena ada angkot yang sama dengan sebelumnya, maka digunakan angkot ABH2
3. Jalur (6-7) tersedia angkot ADL2 sehingga pasti menggunakan angkot ADL2.
4. Jalur (7-8) tersedia angkot ADL1 sehingga pasti menggunakan angkot ADL1.
5. Jalur (7-10) tersedia angkot AL2/HL2/LH1. Karena AL2 dan LH1 sama dengan angkot setelahnya maka perulangan pertama diambil AL2 dan LH1. Kemudian dilakukan perulangan kembali. Karena AL2 sama dengan dua angkot setelahnya, maka diambil AL2.
6. Jalur (10-11) menggunakan angkot AL2/ LH1. Karena ada angkot yang sama dengan sebelum dan setelahnya, maka diambil AL2.
7. Jalur (11-13) menggunakan angkot AL2/HL2. Karena ada angkot yang sama dengan sebelumnya, maka diambil angkot AL2

Angkot yang direkomendasikan:

- ABH2
- ADL2
- ADL1
- AL2

Tahap 3:

Perhitungan Estimasi Waktu Masing-Masing Jenis Transportasi.

- Estimasi Waktu Angkutan Kota

$$ta = \frac{s}{300} + (\sum o \times 3)$$

$$ta = \frac{4770}{300} + (3 \times 3) = 24,9 \text{ menit}$$

- Estimasi Waktu Ojek

$$to = \frac{s}{vj}$$

$$to = \frac{4770}{500} = 9,54 \text{ menit}$$

- Estimasi Waktu Taksi

$$tt = \frac{s}{vt}$$

$$tt = \frac{4770}{400} = 11,925 \text{ menit}$$

Tahap 4:

Perhitungan Tarif Masing-Masing Jenis Transportasi.

- Tarif Angkutan Kota

$$ha = \sum A \times 2500$$

$$ha = (3 \times 2500) = \text{Rp } 7.500,-$$

- Tarif Ojek

$$ho = \left(\frac{s-500}{500} \times 1000\right) + 4000$$

$$ho = \left(\frac{4770-500}{500} \times 1000\right) + 4000 = \text{Rp } 12.540,-$$

- Tarif Taksi

$$ho = \left(\frac{s-100}{100} \times 300\right) + 5000$$

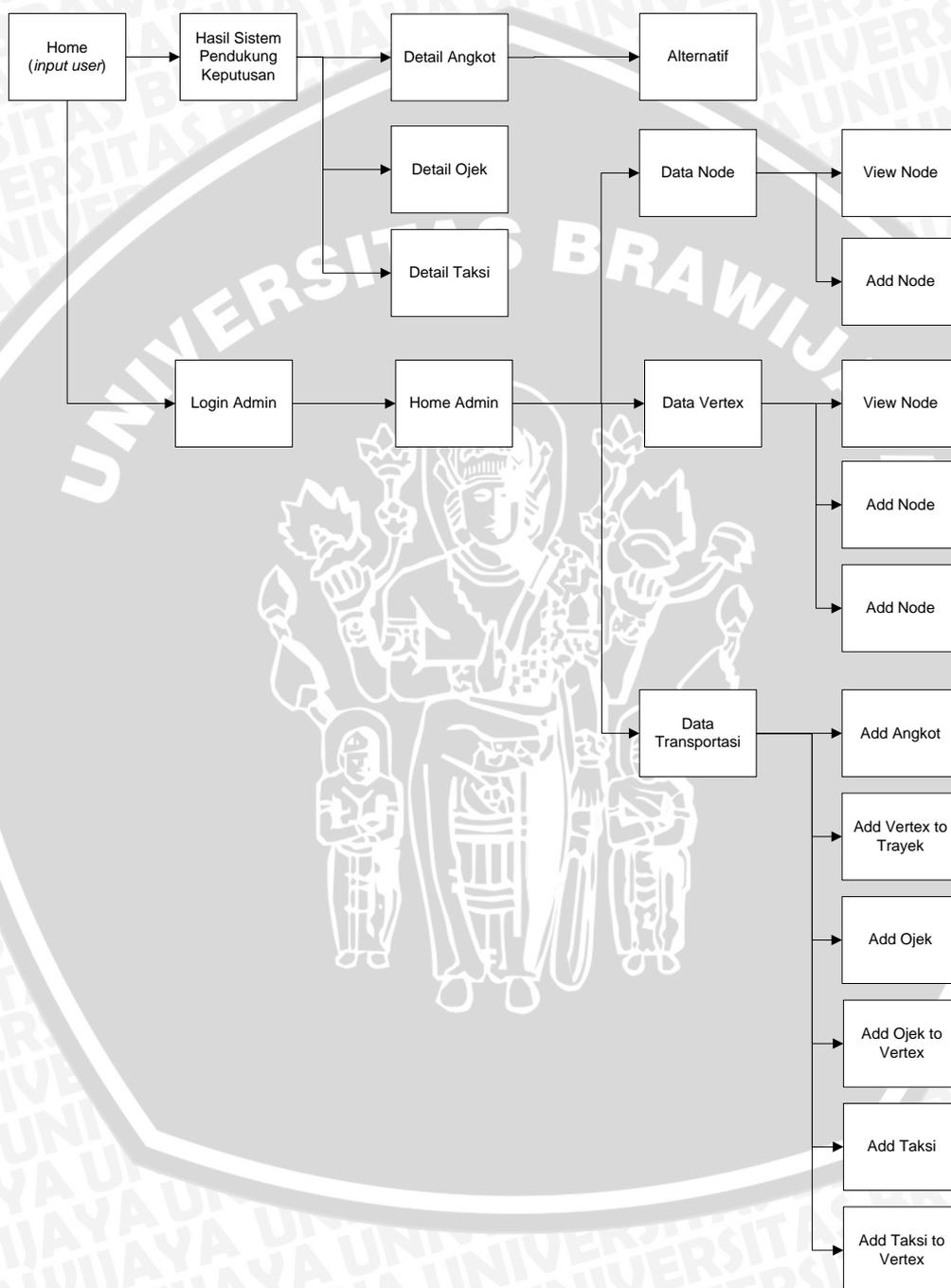
$$h = \left(\frac{4770-100}{100} \times 300\right) + 5000 = \text{Rp } 19.010,- \text{ (minimal Rp } 20.000,-)$$

6. Penentuan Transportasi Rekomendasi Berdasar Prioritas

Proses ini dilakukan dengan membandingkan tarif atau waktu masing-masing jenis transportasi sesuai dengan prioritas yang dipilih oleh pengguna. Dalam

kasus ini, jika memilih prioritas waktu, maka hasilnya adalah ojek, sedang jika prioritas tarif, maka hasilnya adalah angkot.

4.3.3 Antarmuka Pengguna



Gambar 4.14 Site Map Sistem Pendukung Keputusan

Sumber: Perancangan

Antarmuka pengguna pada sistem pendukung keputusan ini disiapkan untuk pengguna yang akan melakukan pencarian informasi tentang pemilihan alat transportasi Kota Malang.

a. Perancangan *Site Map* Sistem Pendukung Keputusan

Sistem ini secara bebas dapat digunakan oleh *admin* dengan login serta *user* tanpa login karena sistem hanya bertugas memberi informasi secara umum kepada masyarakat, khususnya pengguna internet. Untuk keperluan akses, sistem harus dapat menyediakan halaman yang dapat diakses oleh *user* dan *admin*.

Site map sistem pendukung keputusan ditunjukkan pada **Gambar 4.14**.

b. Perancangan Antar Muka

Perancangan antar muka dibutuhkan untuk memberikan gambaran sebenarnya dari bentuk sistem pendukung keputusan yang akan dibangun. Sistem pendukung keputusan yang akan dibangun memiliki 2 bagian penting dalam perancangan antarmukanya, yaitu untuk halaman *user* dan *admin*. Halaman untuk *user* tersebut berupa halaman utama yang digunakan untuk *input user*, halaman hasil pemrosesan sistem, dan halaman detail setiap alat transportasi. Untuk halaman *admin*, berisi *login*, halaman utama untuk *admin*, *update node*, *update vertex*, dan *update* transportasi umum. Berikut merupakan skema umum (*site map*) dari menu-menu yang nantinya terdapat dalam sistem pendukung keputusan berbasis web untuk pemilihan transportasi umum Kota Malang Berdasar Jalur Terpendek dengan Menggunakan algoritma *Floyd-Warshall*.

Pada halaman utama, *user* dapat menginputkan lokasi asal, lokasi yang ingin dituju, dan prioritas.

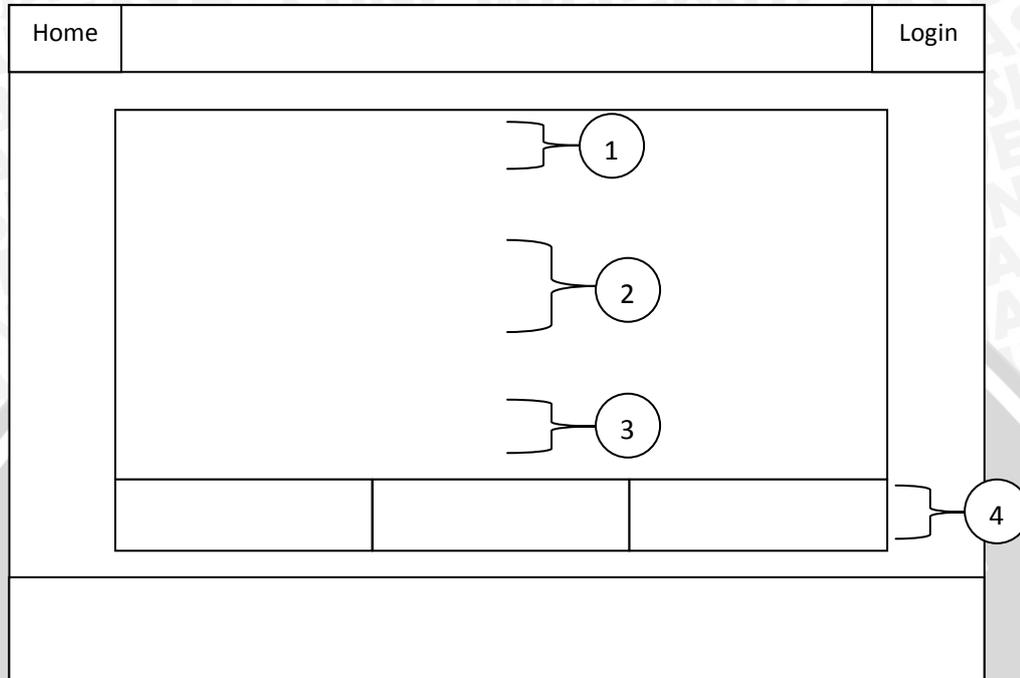
The image shows a wireframe of a web form. At the top left is a 'Home' link and at the top right is a 'Login' link. The main content area contains three input fields labeled 'Asal', 'Tujuan', and 'Prioritas'. Below these fields is a 'Submit' button. Three numbered callouts (1, 2, and 3) are placed to the right of the input fields, with arrows pointing to each field: callout 1 points to the 'Asal' field, callout 2 points to the 'Tujuan' field, and callout 3 points to the 'Prioritas' field.

Gambar 4.15 Perancangan Antar Muka Halaman Utama
Sumber: Perancangan

Keterangan gambar:

1. *User* memasukkan data lokasi asal.
2. *User* memasukkan data lokasi tujuan.
3. *User* memasukkan prioritas yang diinginkan (tarif atau waktu).

Pada halaman hasil SPK, *user* mendapatkan informasi tentang jalur terpendek, jarak terpendek, dan transportasi umum terbaik.



Gambar 4.16 Perancangan Antar Muka Halaman Hasil SPK
Sumber: Perancangan

Keterangan gambar:

1. Hasil jarak terpendek.
2. Hasil jalur terpendek.
3. Hasil transportasi umum rekomendasi yang meliputi estimasi waktu dan estimasi tarif yang akan dibayarkan.
4. Detail transportasi umum.