BAB III
PERANCANGAN SISTEM

3.1 Deskripsi Sistem Pencarian Jalur Terpendek

Masalah yang akan dibahas adalah bagaimana membangun sistem aplikasi pencarian jalur terpendek menggunakan algoritma Label-Setting. Pencarian jalur terpendek di kota Bandung khususnya, menjadi topik utama yang berusaha dipecahkan dan dibahas pada tugas akhir ini. Pada dasarnya program akan mencari jalur terpendek berdasarkan nilai *bobot* atau panjanganya jalur yang dilewati dari asal ke tujuan, dan tidak tergantung dari berapa jumlah *node* atau titik yang dilewati.

3.2 Perancangan Sistem

Spesifikasi minimum untuk membangun sistem pencarian jalur terpendek ini adalah sebagai berikut :

a) Hardware

Kebutuhan *hardware* untuk sistem ini adalah :

1. Processor : Intel Pentium III 600MHz atau yang lebih baik
2. Memory : SDRAM 128MB atau lebih baik
3. 2MB harddisk free sebelum instalasi.
4. Resolusi layar sebesar 800 x 600 atau lebih tinggi dengan 256 warna.
5. Keyboard dan mouse.

b) Software

Kebutuhan dari software untuk sistem ini adalah :

1. OS : Windows XP sp1 atau lebih baik
2. Microsoft .Net Framework 2.0

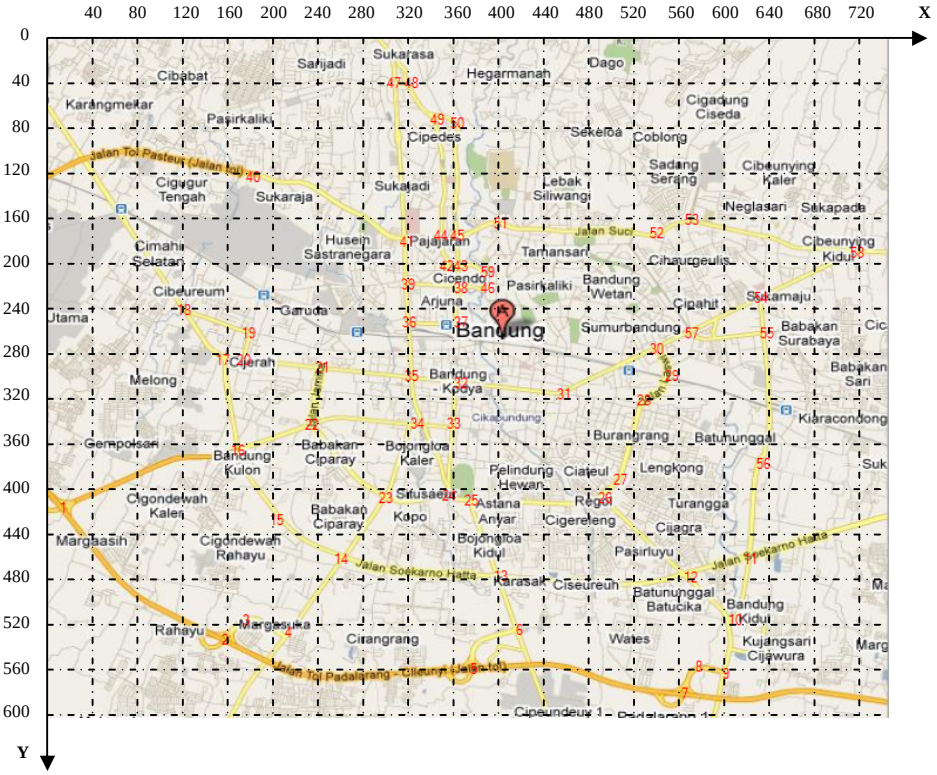
c) Pengguna sistem (user)

1. Pengguna software ini minimal dapat mengoperasikan mouse atau trackpad dan dapat meng-klik mouse kiri dan kanan.

Spesifikasi tersebut diambil dari minimum sistem yang dibutuhkan untuk menginstal software visual basic 2005.

3.2.1 Perancangan Node

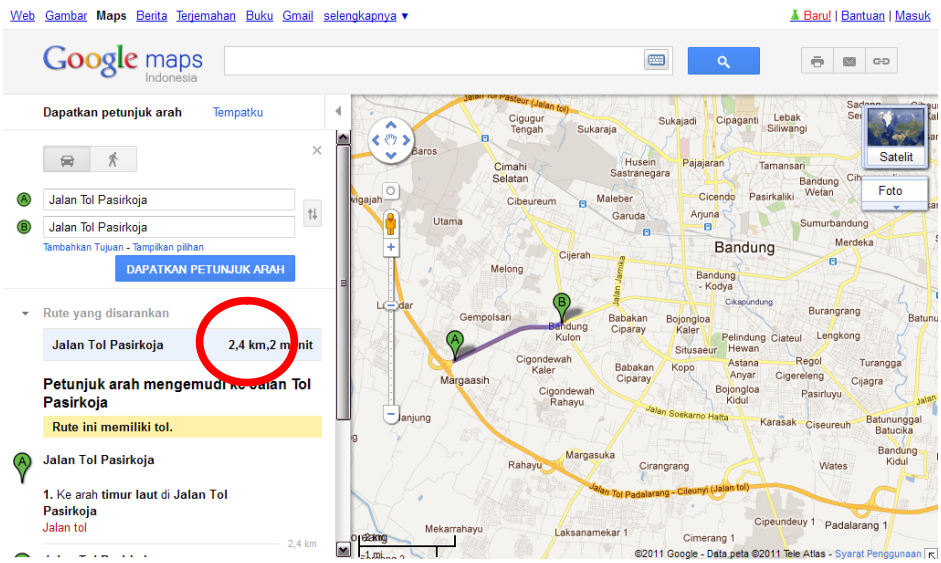
Perancangan node dan penentuan nilai bobot dilakukan secara manual, dengan cara menempatan node - node di setiap persimpangan jalan, atau juga pada jalan yang memiliki jalur yang lebih panjang dari yang lainnya. Berikut adalah gambar peta kota Bandung yang diberikan sumbu X dan Y guna menentukan koordinat node pada peta. Sebagai contoh kita menempatkan node 1 pada jalan tol pasirkoja. Node 1 berada pada titik X = 20 dan Y = 420. Setelah dicoba pada program, tidak sesuai dengan plot yang dibuat. Sehingga harus ada pembetulan nilai X dan Y pada data yang dibuat menjadi X = 13 dan Y = 408.



Gambar 3.1 Koordinat setiap node berdasarkan sumbu X dan Y

Koordinat yang telah didapatkan kemudian diuji kedalam program dan membenarkan (*fix)* node tersebut apabila keluar dari jalur pada peta. Sehingga didapatkan koordinat yang sesuai seperti pada tabel 3.1.

Untuk menentukan nilai bobot pada setiap node, digunakan fasilitas dari *google maps* seperti pada gambar 3.2.

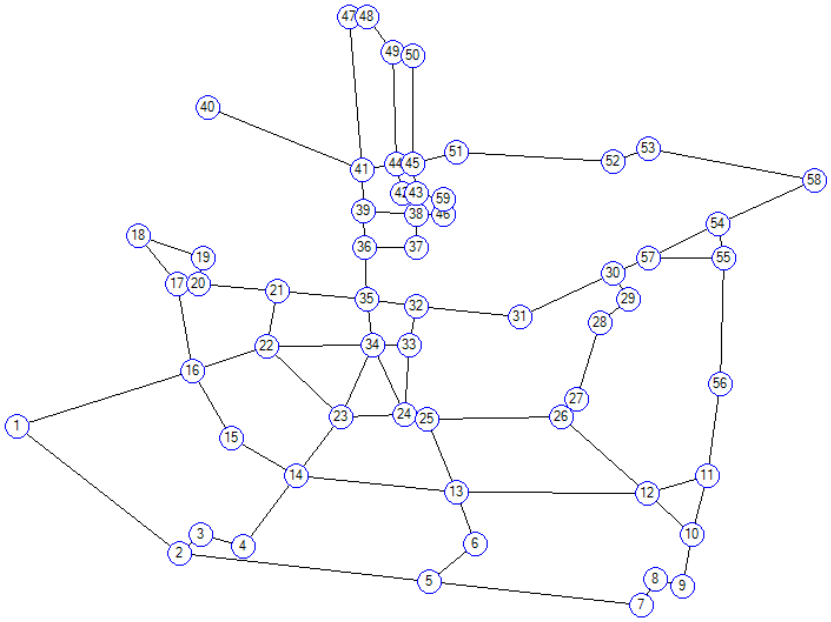


Gambar 3.2 Contoh pengambilan bobot pada setiap node

Pada gambar 3.2 titik A merupakan node 1 dan titik B merupakan node 16, dan bobot yang didapatkan pada kedua node ini adalah sebesar 2,4 Km. Pengambilan nilai bobot tidak hanya pada node 1 dan node 16 saja, tetapi pada semua link yang menghubungkan ke setiap node. Sehingga didapatkan nilai bobot pada setiap link seperti pada tabel 3.2.

3.2.2 Graf

Bentuk graf yang digunakan pada sistem ini adalah berbobot dan tidak berarah seperti pada gambar 3.2.1.1.



Gambar 3.3 *Node dan link dengan 59 buah node*

Selain menempatkan node pada setiap persimpangan jalan, juga harus menentukan link dan bobot pada setiap nodenya yang terdapat pada tabel 3.1.

Tabel 3.1 *Koordinat dengan 59 node*

Node	Sumbu-X	Sumbu-Y
1	13	408
2	151	516
3	169	500
4	205	510
5	363	540
6	402	508
7	543	560
8	555	538
9	578	544
10	586	500

Node	Sumbu-X	Sumbu-Y
11	599	450
12	548	465
13	386	464
14	250	450
15	195	418
16	162	361
17	149	287
18	116	246
19	171	265
20	167	287

Node	Sumbu-X	Sumbu-Y
21	234	293
22	225	340
23	288	400
24	342	398
25	361	402
26	475	400
27	488	385
28	508	320
29	532	300
30	519	278

Node	Sumbu-X	Sumbu-Y
31	440	315
32	352	306
33	346	339
34	315	339
35	310	300
36	308	256
37	352	256
38	352	228
39	307	225
40	175	137

Node	Sumbu-X	Sumbu-Y
41	306	190
42	340	210
43	352	210
44	335	185
45	349	185
46	375	228
47	295	60
48	310	60
49	332	90
50	349	93

Node	Sumbu-X	Sumbu-Y
51	386	175
52	519	183
53	549	172
54	608	236
55	613	265
56	610	372
57	549	265
58	690	199
59	375	215

Nilai bobot didapatkan dari hasil penghitungan jarak dalam dunia nyata, dengan cara mencari besar jarak antara node ke node terdekat (jika ada *link*), dengan menggunakan layanan dari *google maps*.

Tabel 3.2 *Tabel nilai bobot dan link*

Node awal	Link ke-Node	*Bobot* (meter)
1	2	4600
1	16	2400
2	3	400
2	5	3800
3	4	600
4	14	1300
5	6	800
5	7	3400
6	13	800
7	8	200
8	9	400
9	10	800
10	11	1000
10	12	900
11	12	900
11	56	1700
12	13	2700
12	26	1700
13	14	2100

Node awal	Link ke-Node	*Bobot* (meter)
13	25	1200
14	15	1200
14	23	1100
15	16	1100
16	17	1500
16	22	1100
17	18	900
17	20	200
18	19	900
19	20	500
20	21	1100
21	22	800
21	35	1300
22	23	1500
22	34	1400
23	24	900
23	34	1200
24	25	200
24	33	1100

Node awal	Link ke-Node	*Bobot* (meter)
24	34	1200
25	26	1800
26	27	500
27	28	1100
28	29	300
29	30	500
30	31	1500
30	57	500
31	32	1400
32	33	700
32	35	600
33	34	400
34	35	800
35	36	800
36	37	600
36	39	600
37	38	600
38	39	700
38	46	300

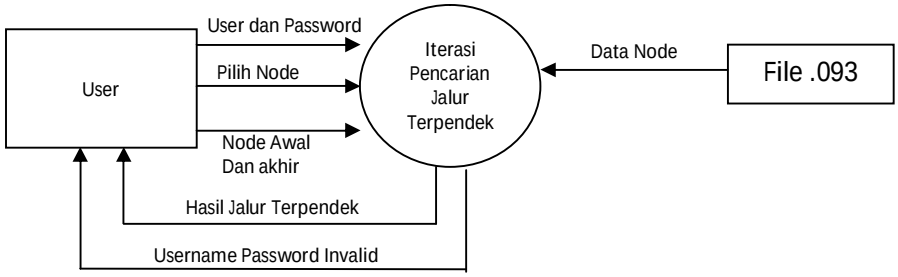
Node awal	Link ke-Node	*Bobot* (meter)
39	41	700
40	41	2400
41	44	500
41	47	2500
42	43	100
42	44	400
43	45	400
43	59	200

Node awal	Link ke-Node	*Bobot* (meter)
44	45	200
44	49	1900
45	50	1800
45	51	600
46	59	300
47	48	100
48	49	600
49	50	300

Node awal	Link ke-Node	*Bobot* (meter)
51	52	2300
52	53	500
53	58	200
54	55	500
54	57	1100
54	58	1500
55	56	1900
55	57	1200

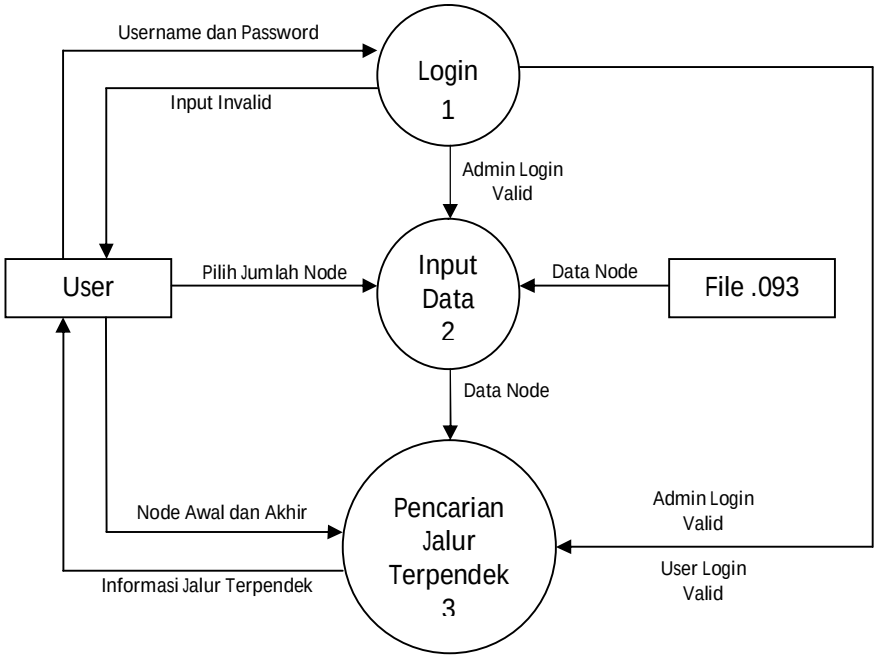
3.2.3 Proses

Proses yang dilakukan pada aplikasi ini adalah dengan cara memasukan data koordinat yang dibutuhkan dan akan menghasilkan jalur terpendek, gambar proses dapat dilihat pada gambar 3.2



Gambar 3.4 *Diagram Konteks*

3.2.4 DFD level 1



Gambar 3.5 DFD level 1

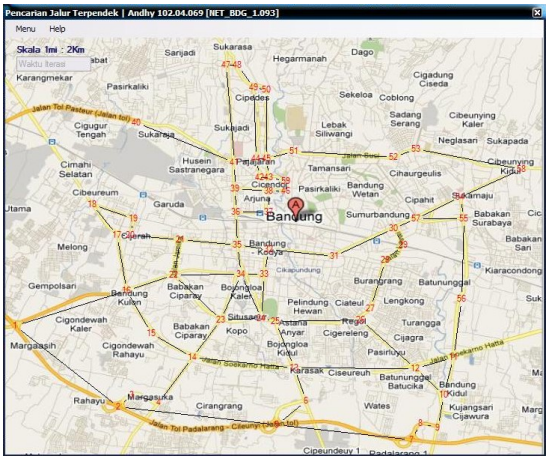
Pengguna akan diminta username dan password. Apabila memasukan username dan password “user”, maka dikategorikan sebagai user yang tidak dapat melakukan proses input data. Dan apabila username dan password “admin”, maka dikategorikan sebagai admin yang dapat melakukan proses input data. Setelah program mendapat data koordinat, *link* dan *bobot* program akan mulai menggambarkan data - data tersebut pada form. Dan ketika user memasukan tujuan asal iterasi sistem dimulai untuk mencari *path tree* dari node asal tersebut, dan ketika user menginputkan node tujuan maka sistem akan memberikan informasi hasil iterasi dari node asal ke node tujuan kepada user.

3.2.5 Perancangan Sistem Input/Output

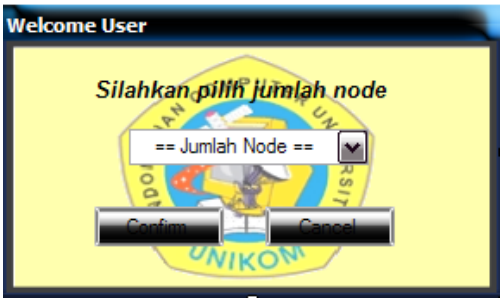
Perancangan T01

	1. Apabila pengguna memasukan username "admin" dan password "admin" maka akan me-*link* ke **T02**. 2. Apabila pengguna memasukan username "user" dan password "user" maka akan me-*link* ke **T03**. 3. Apabila pengguna gagal memasukan username dan password sebanyak 3 kali, maka program akan keluar.

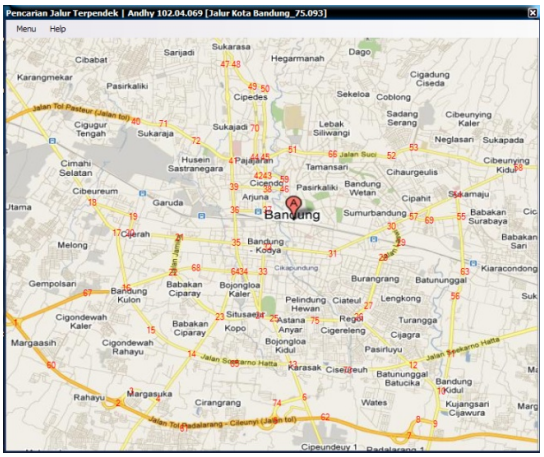
Perancangan T02

	1. Jika Menu di klik maka akan keluar submenu *Shortest Path*, *Logout* dan *Exit.* 2. Jika submenu *Shortest Path* di klik maka akan keluar submenu *Open*, yang akan mengeluarkan *browse file* yang digunakan untuk memasukan jumlah node yang diinginkan 3. Jika submenu *Logout* di klik maka program akan me-*link* ke **T01**. 4. Jika submenu *Exit* di klik maka program akan keluar / *close.* 5. Jika Help di klik maka akan keluar submenu *About* yang akan me-link ke **T05**.

Perancangan T03

	1. Disini user diminta memilih masukan banyaknya node yang akan diterapkan pada peta kota Bandung. Jika telah memilih dan meng-klik *Confirm* maka akan me-link ke **T04**. 2. Jika tombol *Cancel* ditekan maka program akan keluar / *close*.

Perancangan T04

	1. Jika Menu di klik maka akan keluar submenu *Node List, Logout* dan *Exit.* 2. Jika submenu *Node List* di klik maka akan me-link ke **T03**. 3. Jika submenu *Logout* di klik maka program akan me-*link* ke **T01**. 4. Jika submenu *Exit* di klik maka program akan keluar / *close*. 1. Jika Help di klik maka akan keluar submenu *About* yang akan me-link ke **T05**.

Perancangan T05

	1. Pengguna harus menekan *close* untuk menutup form *about*.

3.2.6 Hak akses Admin & User

Pada sistem ini terdapat 2 kategori user, diantaranya adalah :

1. Super User atau Admin

Username : admin

Password : admin

Kelebihan dari Admin adalah dapat memodifikasi isi file .093 karena admin membuka file dengan cara open file atau browse, sehingga admin mengetahui letak dari file .093 tersebut. Selain itu juga admin dapat melihat waktu proses algoritma label-setting dalam menyelesaikan perncarian jalur terpendek pada peta kota Bandung.

2. Common User atau User

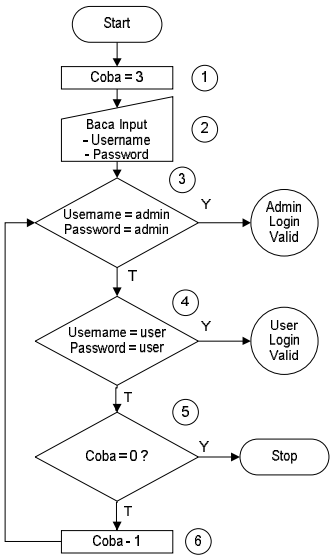
   Username : user

   Password : user

   User hanya dapat memilih jumlah node yang akan dipakai pada peta kota Bandung, dan memberikan inputan sebagai awal dan akhir node.

3.2.7 Perancangan Procedural

Adapun flowchart dari perancangan sistem adalah sebagai berikut.

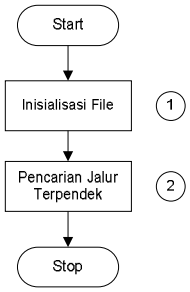


Gambar 3.6 Flowchart Program Utama

Keterangan flowchart pada gambar 3.6 adalah sebagai berikut :

1. Set coba = 3.
2. Baca input username dan password

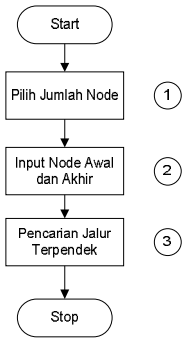
3. Cek apakah username = admin dan password = admin, jika ya admin login valid, jika tidak lanjutkan proses 4.
4. Cek apakah username = user dan password = user, jika ya user login valid, jika tidak lanjutkan proses 5.
5. Cek apakah coba = 0, jika ya hentikan eksekusi program. Jika tidak lanjutkan proses 6.
6. Coba - 1, kembali ke proses 3.



Gambar 3.7 Flowchart Menu Admin

Keterangan flowchart pada gambar 3.7 adalah sebagai berikut :

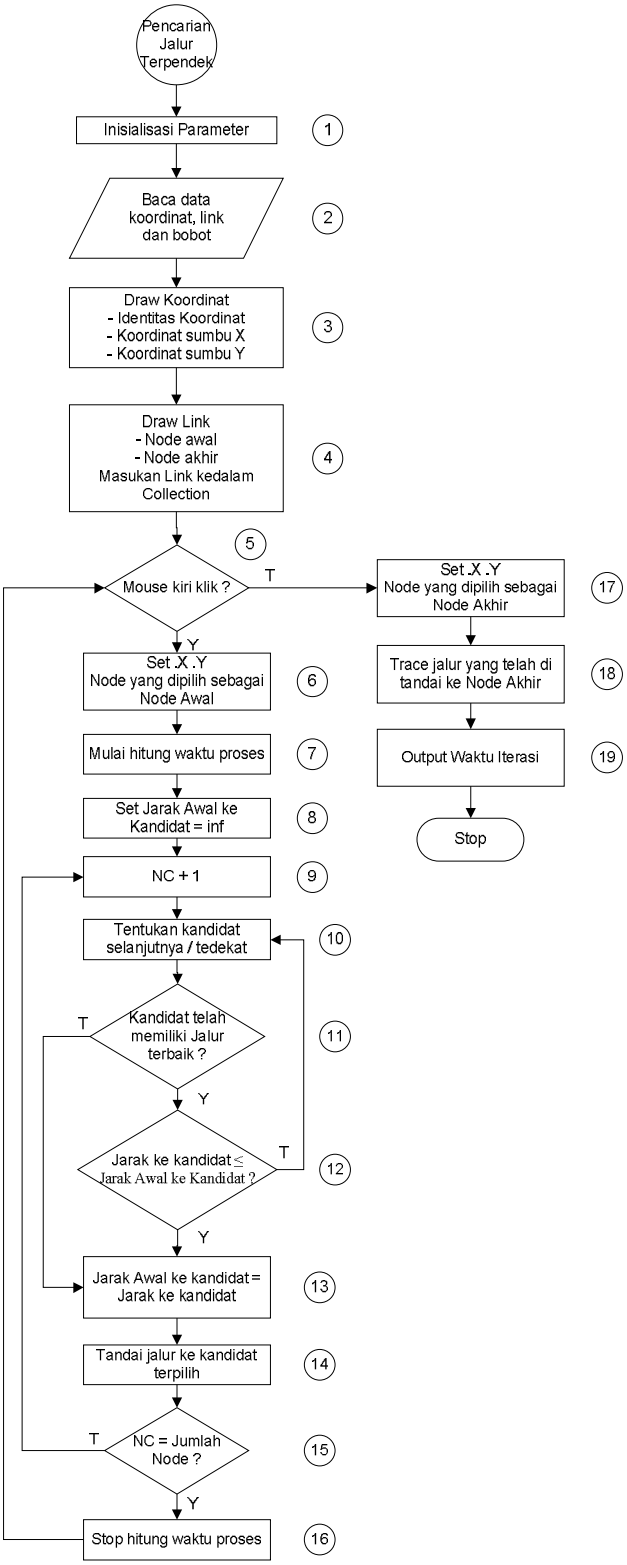
1. Inisialisai file dengan membuka *file path*.
2. Lakukan pencarian jalur terpendek.



Gambar 3.8 Flowchart Menu User

Keterangan flowchart pada gambar 3.8 adalah sebagai berikut :

1. Pilih jumlah node yang akan digunakan pada peta.
2. Masukan input node awal dan node akhir.
3. Lakukan pencarian jalur terpendek.


Pencarian Jalur Terpendek
Inisialisasi Parameter
1
Baca data koordinat, link dan bobot
2
Draw Koordinat
- Identitas Koordinat
- Koordinat sumbu X
- Koordinat sumbu Y
3
Draw Link
- Node awal
- Node akhir
Masukan Link kedalam Collection
4
5
Mouse kiri klik ?
T
Y
Set .X .Y Node yang dipilih sebagai Node Awal
6
Mulai hitung waktu proses
7
Set Jarak Awal ke Kandidat = inf
8
NC + 1
9
Tentukan kandidat selanjutnya / tedekat
10
Kandidat telah memiliki Jalur terbaik ?
11
Jarak ke kandidat ≤ Jarak Awal ke Kandidat ?
12
Jarak Awal ke kandidat = Jarak ke kandidat
13
Tandai jalur ke kandidat terpilih
14
NC = Jumlah Node ?
15
Stop hitung waktu proses
16
Set .X .Y Node yang dipilih sebagai Node Akhir
17
Trace jalur yang telah di tandai ke Node Akhir
18
Output Waktu Iterasi
19
Stop

Gambar 3.9 Flowchart Pencarian Jalur Terpendek

Keterangan flowchart pada gambar 3.9 adalah sebagai berikut :

1. Inisialisasi parameter : dari file yang telah dibuka, program akan menginisialisasikan variable yang akan digunakan. Diantaranya adalah Jumlah Node, Jumlah Link, Koordinat dan Penomoran Node, Bobot link 1 ke link 2, dan NC yang merupakan penghitung jumlah node dari 1 sampai akhir.
2. Merupakan bagian dari inisialisasi parameter.
3. *Draw* atau gambarkan node berdasarkan koordinat dan identitas.
4. *Draw* atau gambarkan link berdasarkan node awal dan akhir, dan masukan kedalam koleksi link.
5. *Mouse event* : penggunaan mouse even bertujuan memudahkan penggunaan dari program. Program akan menunggu mouse ditekan, apabila mouse kiri ditekan maka node tersebut adalah sebagai node asal, dan apabila mouse kanan ditekan maka node tersebut sebagai node tujuan.
6. Setelah node asal ditentukan, program akan menentukan nomor dan koordinat dari node yang dipilih tersebut.
7. Awal dari proses iterasi dan mulai menghitung waktu proses.
8. Set jarak awal ke kandidat dengan nilai terbesar.
9. Tambahkan nilai NC sebagai *Node Counter* atau penghitung jumlah node
10. Tentukan kandidat terdekat selanjutnya yang dapat dilihat dari dari link yang didapat dari data inputan. Jarak link tersebut dinamai sebagai jarak ke kanditat.
11. Periksa apakah kandidat tersebut telah memiliki jalur terbaik. Jika tidak lanjutkan ke proses 11.
12. Seleksi apakah jarak ke kandidat lebih kecil atau sama dengan jarak awal. ke kandidat. Jika tidak kembali ke proses 8.
13. Jika jarak ke kandidat lebih kecil dari jarak awal kandidat, masukan nilai tersebut ke jarak awal ke kandidat.
14. Tandai link/jalur yang dipilih, yang dianggap sebagai jalur terpendek ke node tersebut.

15. Cek apakah semua node telah dilewati, NC = jumlah node. Jika tidak kembali ke proses 7.
16. Hentikan penghitungan waktu proses yang berarti proses iterasi selesai. Kembali ke proses 3.
17. Apabila mouse kanan ditekan tentukan nomor dan koordinat dari node yang dipilih sebagai node tujuan.
18. Program akan mulai mencari link dari node asal ke node tujuan berdasarkan link yang telah ditandai sebelumnya.
19. Output waktu proses hanya untuk menu admin saja, sehingga admin dapat melihat lama waktu proses iterasi dari dua buah node.