BAB 3

ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Hasil penentuan jarak terdekat akan menjadi pertimbangan dalam pengambilan keputusan untuk menunjukkan jalur yang akan ditempuh. Perangkat lunak pencarian jarak terdekat sangat dibutuhkan ketika tidak ada orang untuk menanyakan jalan ke tempat yang diinginkan di kota Tasikmalaya. Algoritma semut dipakai untuk jarak terdekat menjadi solusi terbaik karena cakupan parameter algoritma ini selalu menemukan solusi yang mendekati optimal untuk semua permasalahan yang mempunyai jumlah titik banyak[4].

3.1.1 Analisis Masalah

Lokasi merupakan tempat asal dan tempat tujuan yang akan dicapai oleh pengguna jalan yang berada di kota Tasikmalaya, misalnya hotel, sekolah, bank dan lain sebagainya. Sedangkan titik merupakan persimpangan jalan yang akan dilewati untuk mencapai lokasi.

Analisis masalah yang didapat dalam pencarian jarak terdekat adalah sebagai berikut:

1. Banyaknya wisatawan ataupun masyarakat asli yang kurang mengetahui jalur-jalur terpendek di kota Tasikmalaya, sehingga kurang menghemat waktu.

2. Belum adanya perangkat lunak pada Android untuk pencarian jarak terdekat di kota Tasikmalaya.

3.1.2 Analisis Sistem Yang Sedang Berjalan

Berikut adalah bagaimana skenario pencarian jalan pada umumnya:

1. Mulai bertanya.

   Orang biasanya harus menepi dulu kemudian menanyakan tempat tujuan yang diinginkan.

2. Mencari tempat tujuan.

   Kemudian langsung melanjutkan perjalanan ke tempat yang diinginkan.

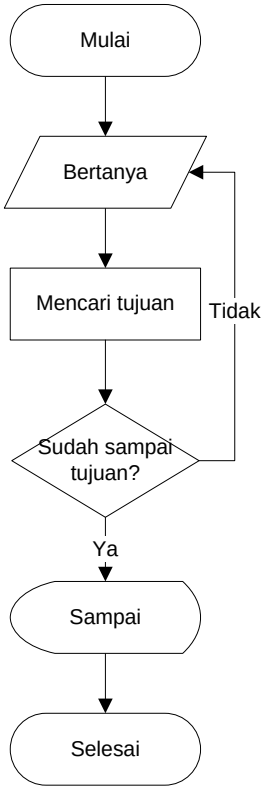
3. Tempat tujuan sudah ditemukan?

   Apabila tempat tujuan masih belum ditemukan, maka harus bertanya kembali pada orang-orang yang berada dipinggir jalan.

4. Sampai ditempat tujuan.

   Akhirnya sampai ke tempat yang diinginkan.

Untuk lebih jelasnya dapat dilihat pada *flowchart* berikut ini:



Gambar 3.1 *Flowchart* Pencarian Jalan

Masalah yang dianalisis adalah mengenai prosedur-prosedur setiap data yang dibutuhkan dan dihasilkan dari aplikasi pencarian jarak terpendek di kota Tasikmalaya. Masalah-masalah yang ada, yaitu:

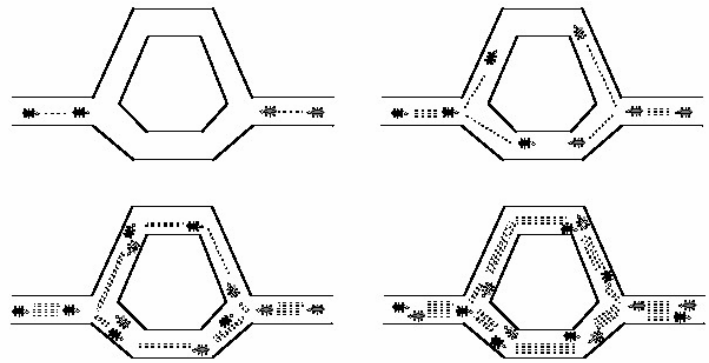
1. Masih banyak orang-orang yang belum mengetahui jarak terpendek di kota Tasikmalaya.
2. Belum adanya perangkat lunak untuk pencarian jarak terdekat di kota Tasikmalaya. Sehingga ketika tidak ada jangkauan internet, pulsa dan orang

untuk menanyakan jalan, orang-orang masih dapat mengetahui rute jarak terpendek ke tempat yang diinginkan di kota Tasikmalaya.

3. Kurang akuratnya informasi yang didapat apabila bertanya langsung pada orang, dan bisa saja orang yang ditanyai berbohong.
4. Memboros waktu dan biaya untuk mencapai tujuan, karena melewati jalan yang belum tentu terdekat jaraknya.

3.1.3 Analisis Algoritma Semut

Koloni semut merupakan algoritma yang bersifat heuristik untuk menyelesaikan masalah optimasi. Algoritma ini diinspirasikan oleh lingkungan koloni semut pada saat mencari makanan. Semut dapat mencari lintasan terpendek dari suatu sumber makanan menuju sarangnya tanpa harus melihatnya secara langsung. Semut-semut mempunyai penyelesaian yang sangat unik dan sangat maju, yaitu dengan menggunakan jejak *pheromon* pada suatu jalur untuk berkomunikasi dan membangun solusi, semakin banyak jejak *pheromon* ditinggalkan maka jalur tersebut akan diikuti oleh semut lain[1].



Gambar 3.2 Semut Dalam Proses Menemukan Sumber Makanan

Dalam algoritma semut, diperlukan beberapa variabel dan langkah-langkah untuk menentukan jarak terpendek[5][6].

Langkah 1:

a. Parameter-parameter yang diperlukan pada algoritma semut adalah sebagai berikut:

1. Intensitas jejak semut antar tempat (τ_{ij}) dan perubahannya

   τ_{ij} harus diinisialisasikan sebelum memulai siklus. τ_{ij} digunakan dalam persamaan probabilitas tempat yang akan dikunjungi. $\Delta\tau_{ij}$ diinisialisasikan setelah selesai satu siklus. $\Delta\tau_{ij}$ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya.

2. Tetapan siklus semut (Q)

   Q merupakan konstanta yang digunakan dalam persamaan untuk menentukan $\Delta\tau_{ij}$. Nilai Q ditentukan oleh pengguna.

3. Tetapan pengendali intensitas jejak semut (α)

   α digunakan dalam persamaan probabilitas tempat yang akan dikunjungi dan berfungsi sebagai pengendali intensitas jejak semut. Nilai α ditentukan oleh pengguna.

4. Tetapan pengendali visibilitas (β)

   β digunakan dalam persamaan probabilitas kota yang akan dikunjungi dan berfungsi sebagai pengendali visibilitas. Nilai β ditentukan oleh pengguna.

5. Visibilitas antar tempat (η_{ij})

   η_{ij} digunakan dalam persamaan probabilitas tempat yang akan dikunjungi. Nilai η_{ij} merupakan hasil dari $1/\eta_{ij}$ (jarak tempat).

6. Banyak semut (m).

   m merupakan banyak semut yang akan melakukan siklus dalam algoritma semut. Nilai m ditentukan oleh pengguna.

7. Tetapan penguapan jejak semut (ρ)

   ρ digunakan untuk menentukan τ_{ij} untuk siklus selanjutnya. Nilai ρ ditentukan oleh pengguna.

8. Jumlah siklus maksimum (NC_{max})

   NC_{max} adalah jumlah maksimum siklus yang akan berlangsung. Siklus akan berhenti sesuai dengan NC_{max} yang telah ditentukan atau telah konvergen. Nilai NC_{max} ditentukan oleh pengguna.

9. Pengisian koordinat tempat

   Pada pengisian koordinat tempat dapat diinputkan sesuai dengan yang kita inginkan. Dengan bertambahnya koordinat maka jalur yang ditempuh akan lebih panjang.

b. Inisialisasi tempat pertama setiap semut

Setelah inisialisasi τ_{ij} dilakukan, kemudian *m* semut ditempatkan pada tempat pertama tertentu secara acak.

Untuk nilai parameter α sebaiknya diberi nilai $0 \leq \alpha \leq 1$, hal ini dimaksudkan untuk menghindari akumulasi *pheromone* yang tidak terbatas pada sisi tersebut. Karena jumlah pheromone yang ditinggalkan tidak mungkin

bertambah kuat tetapi akan bertambah kurang. Untuk nilai parameter β sebaiknya tidak diberi nilai 0 karena jika diberi nilai 0 maka hasil yang dicapai tidak maksimum. Tidak optimum disini berarti suatu kondisi dimana panjang perjalanan yang dicapai tidak minimum.

Langkah 2:

Pengisian tempat pertama ke dalam *tabu list*. Hasil inisialisasi tempat pertama setiap semut dalam langkah 1 harus diisikan sebagai elemen pertama *tabu list*. Hasil dari langkah ini adalah terisinya *tabu list* setiap semut dengan indeks tempat tertentu, yang berarti bahwa setiap $tabu_k(1)$ bisa berisi indeks tempat antara *1* sampai *n* sebagaimana hasil inisialisasi pada langkah 1.

Langkah 3:

Penyusunan rute kunjungan setiap semut ke setiap tempat. Koloni semut yang sudah terdistribusi ke sejumlah atau setiap tempat, akan mulai melakukan perjalanan dari tempat pertama masing-masing sebagai tempat asal dan salah satu tempat-tempat lainnya sebagai tempat tujuan. Kemudian dari tempat kedua masing-masing, koloni semut akan melanjutkan perjalanan dengan memilih salah satu dari tempat-tempat yang tidak terdapat pada $tabu_k$ sebagai tempat tujuan selanjutnya. Perjalanan koloni semut berlangsung terus menerus sampai semua tempat satu persatu dikunjungi atau telah menempati $tabu_k$. Jika s menyatakan indeks urutan kunjungan, tempat asal dinyatakan sebagai $tabu_k(s)$ dan tempat-tempat lainnya dinyatakan sebagai$\{N\text{-}tabu_k\}$, maka untuk menentukan tempat

tujuan digunakan persamaan probabilitas tempat untuk dikunjungi sebagai berikut:

$$P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha}[\eta_{ij}]^{\beta}}{\sum[\tau_{ik'}]^{\alpha}[\eta_{ik'}]^{\beta}} \text{ untuk j} \in \{N\text{-}tabu_k\}$$

$$\text{k'} \in \{N\text{-}tabu_k\}$$

$$\text{dan } p_{ij}^{k} = 0, \text{ untuk j lainnya}$$

dengan i sebagai indeks tempat asal dan j sebagai indeks tempat tujuan.

Langkah 4:

a. Perhitungan panjang rute setiap semut

Perhitungan panjang rute tertutup (*length closed tour*) atau L_k setiap semut dilakukan setelah satu siklus diselesaikan oleh semua semut. Perhitungan dilakukan berdasarkan *tabu*$_k$ masing-masing dengan persamaan berikut:

$$L_k = d_{tabu_k(n),tabu_k(1)} + \sum_{s=1}^{n-1} d_{tabu_k(s),tabu_k(s+1)}$$

Dengan d_{ij} adalah jarak antara tempat i ke tempat j yang dihitung berdasarkan persamaan:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

b. Pencarian jarak terpendek

Setelah L_k setiap semut dihitung, akan diperoleh harga minimal panjang rute tertutup setiap siklus atau L_{minNC} dan harga minimal panjang jalur tertutup secara keseluruhan adalah atau L_{min}.

c. Perhitungan perubahan harga intensitas jejak kaki semut antar tempat $\Delta\tau_{ij}$

Koloni semut akan meninggalkan jejak-jejak kaki pada lintasan antar tempat yang dilaluinya. Adanya penguapan dan perbedaan jumlah semut yang lewat, menyebabkan kemungkinan terjadinya perubahan harga intensitas jejak kaki semut antar tempat. Persamaan perubahannya adalah:

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

keterangan:

m = banyak semut

τ_{ij} = panjang jalur setiap semut

dengan $\Delta\tau_{ij}^{k}$ adalah perubahan harga intensitas jejak kaki semut antar tempat setiap semut yang dihitung berdasarkan persamaan.

$$\Delta\tau_{ij}^{k} = \frac{Q}{L_k}$$

keterangan:

Q = tetapan siklus semut

L_k = *length close tour* (lct)

untuk (i.j) ∈ tempat asal dan tempat tujuan dalam $tabu_k$

$\Delta\tau_{ij}^{k}$ = 0 untuk (i.j) lainnya

Langkah 5:

1 Perhitungan harga intensitas jejak kaki semut antar tempat untuk siklus selanjutnya.

Harga intensitas jejak kaki semut antar tempat pada semua lintasan antar tempat ada kemungkinan berubah karena adanya penguapan dan perbedaan jumlah semut yang melewati. Untuk siklus selanjutnya, semut yang akan melewati lintasan tersebut harga intensitasnya telah berubah. Harga intensitas jejak kaki semut antar tempat untuk siklus selanjutnya dihitung dengan persamaan:

$$\tau_{ij} = \rho \, . \, \tau_{ij} + \Delta \tau_{ij}$$

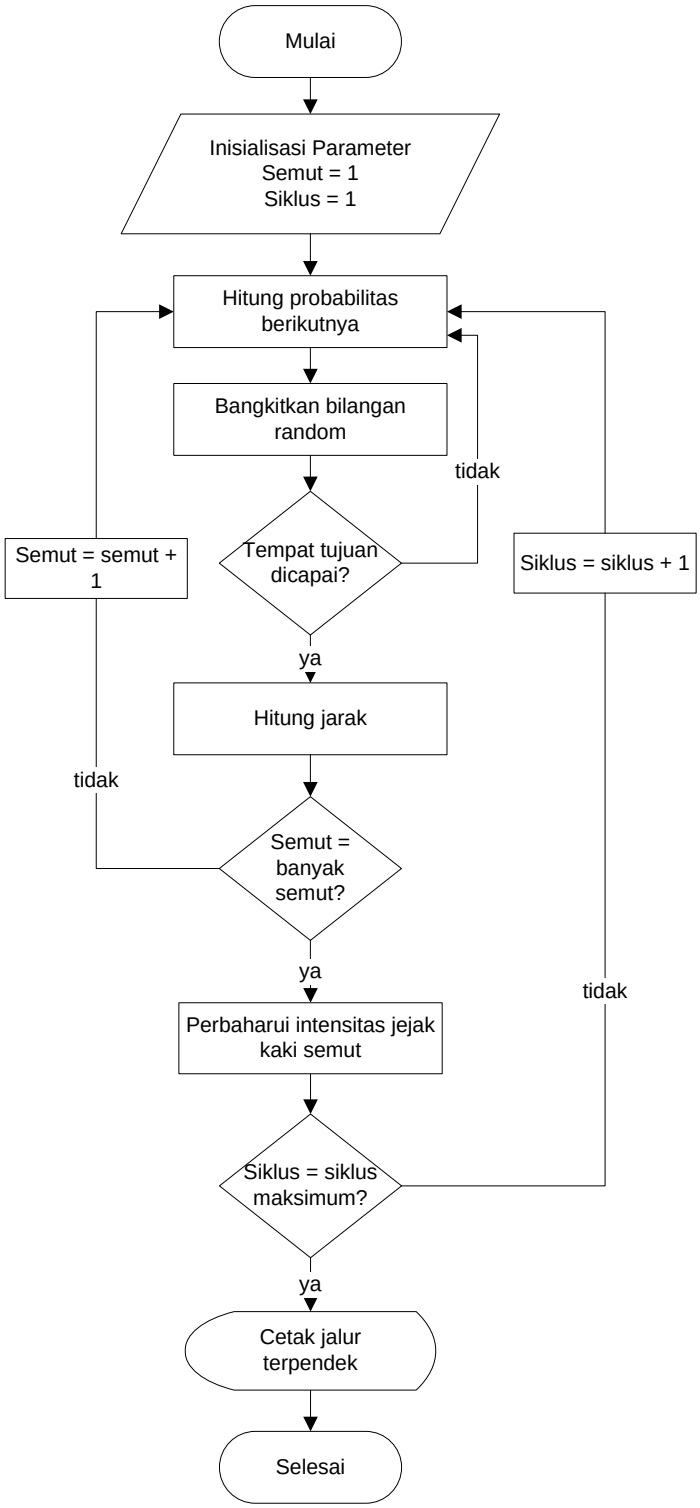
2 Atur ulang harga perubahan intensitas jejak kaki semut antar tempat.

Untuk siklus selanjutnya perubahan harga intensitas jejak semut antar tempat perlu diatur kembali agar memiliki nilai sama dengan nol.

Langkah 6:

Pengosongan *tabu list*, dan ulangi langkah 2 jika diperlukan. *Tabu list* perlu dikosongkan untuk diisi lagi dengan urutan tempat yang baru pada siklus selanjutnya, jika jumlah siklus maksimum belum tercapai atau belum terjadi konvergensi. Algoritma diulang lagi dari langkah pengisian *tabu list* dengan harga parameter intensitas jejak kaki semut antar tempat yang sudah diperbaharui.

Untuk lebih jelasnya dapat dilihat pada gambar III.3 berikut ini:



Gambar 3.3 *Flowchart* Algoritma Semut

3.1.3 Analisa Algoritma Semut

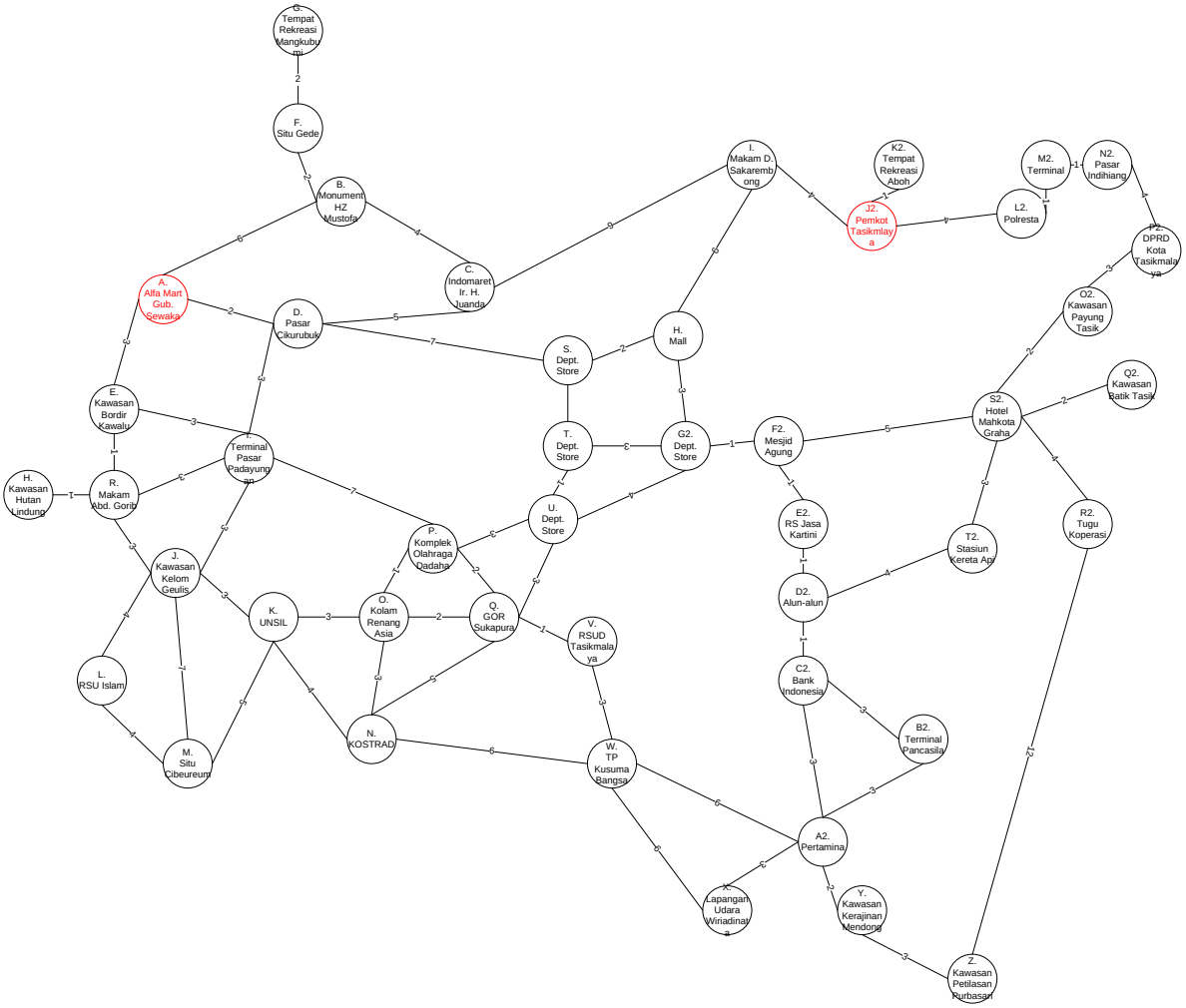
Seorang wisatawan ingin mencari jarak terpendek dari Alfa Mart Gub. Sewaka menuju Pemkot Tasikmalaya, maka dapat ditentukan:

α = 1,0 Siklus = 2

β = 0,0 Semut = 3

Q = 1,0

P = 0,5



Gambar 3.4 Contoh Pencarian

A. Siklus pertama:

Semut pertama akan melakukan pencarian, dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0^{1.0}.\left(\frac{1}{6}\right)^{0}}{\left(0^{1.0}.\frac{1}{6}^{0}\right)+\left(0^{1.0}.\frac{1}{2}^{0}\right)} = \frac{0\ .1}{0+0} = 0$$

$$P_{A\to D} = \frac{0^{1.0}.\left(\frac{1}{2}\right)^{0}}{\left(0^{1.0}.\frac{1}{6}^{0}\right)+\left(0^{1.0}.\frac{1}{2}^{0}\right)} = \frac{0\ .1}{0+0} = 0$$

$$P_{D\to C} = \frac{0^{1.0}.\left(\frac{1}{5}\right)^{0}}{\left(0^{1.0}.\frac{1}{5}^{0}\right)+\left(0^{1.0}.\frac{1}{7}^{0}\right)} = \frac{0\ .1}{0+0} = 0$$

$$P_{D\to S} = \frac{0^{1.0}.\left(\frac{1}{7}\right)^{0}}{\left(0^{1.0}.\frac{1}{5}^{0}\right)+\left(0^{1.0}.\frac{1}{7}^{0}\right)} = \frac{0\ .1}{0+0} = 0$$

$$P_{C\to I} = \frac{0^{1.0}.\left(\frac{1}{7}\right)^{0}}{0^{1.0}.\left(\frac{1}{7}\right)^{0}} = \frac{0\ .1}{0\ .1} = 0$$

$$P_{I\to J} = \frac{0^{1.0}.\left(\frac{1}{2}\right)^{0}}{0^{1.0}.\left(\frac{1}{2}\right)^{0}} = \frac{0\ .1}{0\ .1} = 0$$

Setelah semut pertama melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0 + \Delta\tau_{A,B} = 0{,}5\ .\ 0 + \frac{1.0}{19} = 0 + 0{,}05 = 0{,}05$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0 + \Delta\tau_{A,D} = 0{,}5\ .\ 0 + \frac{1.0}{16} = 0 + 0{,}06 = 0{,}06$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0 + \Delta\tau_{D,C} = 0{,}5\ .\ 0 + \frac{1.0}{16} = 0 + 0{,}06 = 0{,}06$$

$$\tau_{D,S} = \mathrm{p}\ .\ 0 + \Delta\tau_{D,C} = 0{,}5\ .\ 0 + \frac{1.0}{17} = 0 + 0{,}05 = 0{,}05$$

$$\tau_{C,I} = \mathrm{p}\ .\ 0 + \Delta\tau_{C,I} = 0{,}5\ .\ 0 + \frac{1.0}{16} = 0 + 0{,}06 = 0{,}06$$

$$\tau_{I,J} = \mathrm{p}\ .\ 0 + \Delta\tau_{C,I} = 0{,}5\ .\ 0 + \frac{1.0}{16} = 0 + 0{,}06 = 0{,}06$$

Setelah semua jumlah feromon di-*update*, sekarang giliran semut kedua yang akan melakukan pencarian. Dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0{,}05^{1.0}.\left(\frac{1}{6}\right)^{0{,}05}}{\left(0{,}05^{1.0}.\frac{1}{6}^{0{,}05}\right)+\left(0{,}05^{1.0}.\frac{1}{2}^{0{,}05}\right)} = \frac{0{,}05\ .0{,}91}{0{,}04+0{,}04} = 0{,}5$$

$$P_{A\to D} = \frac{0{,}06^{1.0}.\left(\frac{1}{2}\right)^{0{,}06}}{\left(0{,}06^{1.0}.\frac{1}{6}^{0{,}06}\right)+\left(0{,}06^{1.0}.\frac{1}{2}^{0{,}06}\right)} = \frac{0{,}06\ .0{,}95}{0{,}05+0{,}05} = 0{,}5$$

$$P_{B\to C} = \frac{0^{1.0}.\left(\frac{1}{4}\right)^{0}}{0^{1.0}.\left(\frac{1}{4}\right)^{0}} = \frac{0\ .1}{0\ .1} = 0$$

$$P_{C\to I} = \frac{0{,}06^{1.0}.\left(\frac{1}{7}\right)^{0{,}06}}{0{,}06^{1.0}.\left(\frac{1}{7}\right)^{0{,}06}} = \frac{0{,}06\ .0{,}88}{0{,}06\ .0{,}88} = 1$$

$$P_{I\to J} = \frac{0{,}06^{1.0}.\left(\frac{1}{2}\right)^{0{,}06}}{0{,}06^{1.0}.\left(\frac{1}{2}\right)^{0{,}06}} = \frac{0{,}06\ .0{,}95}{0{,}06\ .0{,}95} = 1$$

Setelah semut kedua melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0{,}05 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}05 + \frac{1.0}{19} = 0{,}02 + 0{,}05 = 0{,}08$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0{,}05 + \Delta\tau_{A,D} = 0{,}5\ .\ 0{,}05 + \frac{1.0}{16} = 0{,}02 + 0{,}06 = 0{,}08$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0 + \Delta\tau_{D,C} = 0{,}5\ .\ 0 + \frac{1.0}{16} = 0 + 0{,}06 = 0{,}06$$

$$\tau_{C,I} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

$$\tau_{I,J} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

Setelah semua jumlah feromon di-*update*, sekarang giliran semut ketiga yang akan melakukan pencarian. Dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0{,}08^{1.0}.\left(\frac{1}{6}\right)^{0{,}08}}{\left(0{,}08^{1.0}.\frac{1}{6}^{0{,}08}\right)+\left(0{,}05^{1.0}.\frac{1}{2}^{0{,}08}\right)} = \frac{0{,}08\ .0{,}86}{0{,}06+0{,}04} = 0{,}6$$

$$P_{A\to D} = \frac{0{,}08^{1.0}.\left(\frac{1}{2}\right)^{0{,}06}}{\left(0{,}08^{1.0}.\frac{1}{6}^{0{,}08}\right)+\left(0{,}08^{1.0}.\frac{1}{2}^{0{,}08}\right)} = \frac{0{,}08\ .0{,}95}{0{,}06+0{,}07} = 0{,}53$$

$$P_{D\to C} = \frac{0{,}06^{1.0}.\left(\frac{1}{5}\right)^{0{,}06}}{\left(0{,}06^{1.0}.\frac{1}{5}^{0{,}06}\right)+\left(0{,}06^{1.0}.\frac{1}{7}^{0{,}06}\right)} = \frac{0{,}06\ .0{,}95}{0{,}05+0{,}05} = 0{,}15$$

$$P_{D\to S} = \frac{0{,}05^{1.0}.\left(\frac{1}{7}\right)^{0{,}05}}{\left(0{,}05^{1.0}.\frac{1}{5}^{0{,}05}\right)+\left(0{,}06^{1.0}.\frac{1}{7}^{0{,}05}\right)} = \frac{0{,}05\ .0{,}9}{0{,}04+0{,}04} = 0{,}5$$

$$P_{S\to H} = \frac{0^{1.0}.\left(\frac{1}{2}\right)^{0}}{0^{1.0}.\left(\frac{1}{2}\right)^{0}} = \frac{0\ .1}{0\ .1} = 0$$

$$P_{H\to I} = \frac{0^{1.0}.\left(\frac{1}{4}\right)^{0}}{0^{1.0}.\left(\frac{1}{4}\right)^{0}} = \frac{0\ .1}{0\ .1} = 0$$

$$P_{I\to J} = \frac{0{,}56^{1.0}.\left(\frac{1}{2}\right)^{0{,}56}}{0{,}56^{1.0}.\left(\frac{1}{2}\right)^{0{,}56}} = \frac{0{,}56\ .0{,}67}{0{,}56\ .0{,}67} = 1$$

Setelah semut ketiga melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0{,}6 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}6 + \frac{1.0}{19} = 0{,}3 + 0{,}05 = 0{,}35$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0{,}53 + \Delta\tau_{A,D} = 0{,}5\ .\ 0{,}53 + \frac{1.0}{17} = 0{,}26 + 0{,}05 = 0{,}31$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0{,}15 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}15 + \frac{1.0}{17} = 0{,}12 + 0{,}05 = 0{,}17$$

$$\tau_{D,S} = \mathrm{p}\ .\ 0{,}5 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}5 + \frac{1.0}{17} = 0{,}25 + 0{,}05 = 0{,}3$$

$$\tau_{S,H} = \mathrm{p}\ .\ 0 + \Delta\tau_{A,D} = 0{,}5\ .\ 0 + \frac{1.0}{17} = 0 + 0{,}05 = 0{,}05$$

$$\tau_{H,I} = \mathrm{p}\ .\ 0 + \Delta\tau_{D,C} = 0{,}5\ .\ 0 + \frac{1.0}{17} = 0 + 0{,}05 = 0{,}05$$

$$\tau_{I,J} = \mathrm{p}\ .\ 0 + \Delta\tau_{D,C} = 0{,}5\ .\ 1 + \frac{1.0}{17} = 0{,}5 + 0{,}05 = 0{,}55$$

B. Siklus kedua:

Semut pertama akan melakukan pencarian, dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0{,}35^{1.0}.\left(\frac{1}{6}\right)^{0{,}35}}{\left(0{,}35^{1.0}.\frac{1}{6}^{0{,}35}\right)+\left(0{,}35^{1.0}.\frac{1}{2}^{0{,}35}\right)} = \frac{0{,}35\ .0{,}52}{0{,}18+0{,}27} = 0{,}4$$

$$P_{A\to D} = \frac{0{,}31^{1.0}.\left(\frac{1}{2}\right)^{0{,}31}}{\left(0{,}31^{1.0}.\frac{1}{6}^{0{,}31}\right)+\left(0{,}31^{1.0}.\frac{1}{2}^{0{,}31}\right)} = \frac{0{,}31\ .0{,}8}{0{,}17+0{,}24} = 0{,}58$$

$$P_{D\to C} = \frac{0{,}17^{1.0}.\left(\frac{1}{5}\right)^{0{,}17}}{\left(0{,}17^{1.0}.\frac{1}{5}^{0{,}17}\right)+\left(0{,}17^{1.0}.\frac{1}{7}^{0{,}17}\right)} = \frac{0{,}17\ .0{,}76}{0{,}12+0{,}12} = 0{,}5$$

$$P_{D\to S} = \frac{0{,}3^{1.0}.\left(\frac{1}{7}\right)^{0{,}3}}{\left(0{,}3^{1.0}.\frac{1}{5}^{0{,}3}\right)+\left(0{,}3^{1.0}.\frac{1}{7}^{0{,}3}\right)} = \frac{0{,}3\ .0{,}55}{0{,}18+0{,}16} = 0{,}47$$

$$P_{C\to I} = \frac{0{,}56^{1.0}.\left(\frac{1}{7}\right)^{0{,}56}}{0{,}56^{1.0}.\left(\frac{1}{7}\right)^{0{,}56}} = \frac{0{,}56\ .0{,}33}{0{,}56\ .0{,}33} = 1$$

$$P_{I\to J} = \frac{0{,}55^{1.0}.\left(\frac{1}{2}\right)^{0{,}55}}{0{,}55^{1.0}.\left(\frac{1}{2}\right)^{0{,}55}} = \frac{0{,}55\ .0{,}68}{0{,}55\ .0{,}68} = 1$$

Setelah semut pertama melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0{,}4 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}4 + \frac{1.0}{19} = 0{,}2 + 0{,}05 = 0{,}25$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0{,}58 + \Delta\tau_{A,D} = 0{,}5\ .\ 0{,}58 + \frac{1.0}{16} = 0{,}29 + 0{,}06 = 0{,}35$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0{,}5 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}5 + \frac{1.0}{16} = 0{,}25 + 0{,}06 = 0{,}31$$

$$\tau_{D,S} = \mathrm{p}\ .\ 0{,}47 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}47 + \frac{1.0}{17} = 0{,}23 + 0{,}14 = 0{,}37$$

$$\tau_{C,I} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

$$\tau_{I,J} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

Setelah semua jumlah feromon di-*update*, sekarang giliran semut kedua yang akan melakukan pencarian. Dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0{,}4^{1.0}.\left(\frac{1}{6}\right)^{0{,}4}}{\left(0{,}4^{1.0}.\frac{1}{6}^{0{,}4}\right)+\left(0{,}4^{1.0}.\frac{1}{2}^{0{,}4}\right)} = \frac{0{,}4\ .0{,}48}{0{,}19+0{,}3} = 0{,}38$$

$$P_{A\to D} = \frac{0{,}58^{1.0}.\left(\frac{1}{2}\right)^{0{,}58}}{\left(0{,}58^{1.0}.\frac{1}{6}^{0{,}58}\right)+\left(0{,}58^{1.0}.\frac{1}{2}^{0{,}58}\right)} = \frac{0{,}58\ .0{,}66}{0{,}19+0{,}38} = 0{,}66$$

$$P_{D\to C} = \frac{0{,}5^{1.0}.\left(\frac{1}{5}\right)^{0{,}5}}{\left(0{,}5^{1.0}.\frac{1}{5}^{0{,}5}\right)+\left(0{,}5^{1.0}.\frac{1}{7}^{0{,}5}\right)} = \frac{0{,}5\ .0{,}44}{0{,}22+0{,}18} = 0{,}55$$

$$P_{D\to S} = \frac{0{,}47^{1.0}.\left(\frac{1}{7}\right)^{0{,}47}}{\left(0{,}47^{1.0}.\frac{1}{5}^{0{,}47}\right)+\left(0{,}47^{1.0}.\frac{1}{7}^{0{,}47}\right)} = \frac{0{,}47\ .0{,}39}{0{,}21+0{,}18} = 0{,}46$$

$$P_{C\to I} = \frac{1^{1.0}.\left(\frac{1}{7}\right)^{1}}{1^{1.0}.\left(\frac{1}{7}\right)^{1}} = \frac{1\ .0{,}14}{1\ .0{,}14} = 1$$

$$P_{I\to J} = \frac{1^{1.0}.\left(\frac{1}{2}\right)^{1}}{1^{1.0}.\left(\frac{1}{2}\right)^{1}} = \frac{1\ .0{,}5}{1\ .0{,}5} = 1$$

Setelah semut kedua melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0{,}38 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}38 + \frac{1.0}{19} = 0{,}19 + 0{,}05 = 0{,}24$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0{,}66 + \Delta\tau_{A,D} = 0{,}5\ .\ 0{,}66 + \frac{1.0}{16} = 0{,}33 + 0{,}06 = 0{,}39$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0{,}55 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}55 + \frac{1.0}{16} = 0{,}27 + 0{,}06 = 0{,}33$$

$$\tau_{D,S} = \mathrm{p}\ .\ 0{,}46 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}46 + \frac{1.0}{17} = 0{,}23 + 0{,}14 = 0{,}37$$

$$\tau_{C,I} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

$$\tau_{I,J} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

Setelah semua jumlah feromon di-*update*, sekarang giliran semut ketiga yang akan melakukan pencarian. Dapat dihitung dengan cara berikut ini:

$$P_{A\to B} = \frac{0{,}24^{1.0}.\left(\frac{1}{6}\right)^{0{,}24}}{\left(0{,}24^{1.0}.\frac{1}{6}^{0{,}24}\right)+\left(0{,}24^{1.0}.\frac{1}{2}^{0{,}24}\right)} = \frac{0{,}24\ .0{,}64}{0{,}15+0{,}18} = 0{,}45$$

$$P_{A\to D} = \frac{0{,}39^{1.0}.\left(\frac{1}{2}\right)^{0{,}39}}{\left(0{,}39^{1.0}.\frac{1}{6}^{0{,}39}\right)+\left(0{,}39^{1.0}.\frac{1}{2}^{0{,}39}\right)} = \frac{0{,}39\ .0{,}76}{0{,}18+0{,}29} = 0{,}61$$

$$P_{D\to C} = \frac{0{,}33^{1.0}.\left(\frac{1}{5}\right)^{0{,}33}}{\left(0{,}33^{1.0}.\frac{1}{5}^{0{,}33}\right)+\left(0{,}33^{1.0}.\frac{1}{7}^{0{,}33}\right)} = \frac{0{,}33\ .0{,}58}{0{,}19+0{,}17} = 0{,}52$$

$$P_{D\to S} = \frac{0{,}37^{1.0}.\left(\frac{1}{7}\right)^{0{,}37}}{\left(0{,}37^{1.0}.\frac{1}{5}^{0{,}37}\right)+\left(0{,}37^{1.0}.\frac{1}{7}^{0{,}37}\right)} = \frac{0{,}37\ .0{,}48}{0{,}2+0{,}17} = 0{,}45$$

$$P_{C\to I} = \frac{0{,}56^{1.0}.\left(\frac{1}{7}\right)^{0{,}56}}{0{,}56^{1.0}.\left(\frac{1}{7}\right)^{0{,}56}} = \frac{0{,}56\ .0{,}33}{0{,}56\ .0{,}33} = 1$$

$$P_{I\to J} = \frac{0{,}56^{1.0}.\left(\frac{1}{2}\right)^{0{,}56}}{0{,}56^{1.0}.\left(\frac{1}{2}\right)^{0{,}56}} = \frac{0{,}56\ .0{,}67}{0{,}56\ .0{,}67} = 1$$

Setelah semut ketiga melakukan pencarian, maka feromon dapat di-*update* dengan cara berikut ini:

$$\tau_{A,B} = \mathrm{p}\ .\ 0{,}45 + \Delta\tau_{A,B} = 0{,}5\ .\ 0{,}45 + \frac{1.0}{19} = 0{,}22 + 0{,}05 = 0{,}27$$

$$\tau_{A,D} = \mathrm{p}\ .\ 0{,}61 + \Delta\tau_{A,D} = 0{,}5\ .\ 0{,}61 + \frac{1.0}{16} = 0{,}3 + 0{,}06 = 0{,}36$$

$$\tau_{D,C} = \mathrm{p}\ .\ 0{,}52 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}52 + \frac{1.0}{16} = 0{,}26 + 0{,}06 = 0{,}32$$

$$\tau_{D,S} = \mathrm{p}\ .\ 0{,}45 + \Delta\tau_{D,C} = 0{,}5\ .\ 0{,}45 + \frac{1.0}{17} = 0{,}22 + 0{,}14 = 0{,}36$$

$$\tau_{C,I} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

$$\tau_{I,J} = \mathrm{p}\ .\ 1 + \Delta\tau_{C,I} = 0{,}5\ .\ 1 + \frac{1.0}{16} = 0{,}5 + 0{,}06 = 0{,}56$$

n = semut

x = siklus

Jadi, O (n.x) karena 1 semut melakukan perulangan sebanyak siklus.

Dari pencarian tersebut, maka dapat disimpulkan:

Jarak terpendek dari Alfa Mart Gub. Sewaka menuju Pemkot Tasikmalaya adalah melalui Pasar Cikurubuk, Indomaret Ir. Juanda dan Makam D. Sakarembong karena menghasilkan jumlah feromon yang lebih besar daripada melalui rute lainnya

Tabel 3-1 *Pseudocode* Algoritma Semut

```
For each pair (t,v) 1. /* Initialization phase */
  τ(t,v):= τ0
End-for
For k:=1 to m do
  Let tk1 be the starting city for ant k
  Jk(tk1):= {1, ..., n} - rk1
  tk:= tk1
End-for
For i:=1 to n do
  If i<n then
     For k:=1 to m do
       Choose the next city vk according to Eq.(1) and Eq.(2)
       Jk(sk):= Jk (tk) - vk
       Tourk (i):=(tk ,vk)
     End-for
  Else
     For k:=1 to m do
       vk := tk1
       Tourk (i):=(tk ,vk)
     End-for
  End-if
  For k:=1 to m do
    τ(tk ,vk):=(1-ρ)τ(tk ,vk)+ ρτ0
     tk := vk /* New city for ant k */
  End-for
End-for
For k:=1 to m do
     Compute Lk
```

```
End-for
Compute Lbest
For each edge (t,v)
   T(tk ,vk):=(1-α)τ(tk,vk) + α(Lbest)-1
End-for
4. If (End_condition = True) then
  Print shortest of Lk
Else
  goto Phase 2
End-if
```

Berdasarkan *pseudocode* pada tabel 3.1, dapat dihitung nilai kompleksitas dari algoritma tersebut. Untuk menghitung kompleksitas suatu algoritma digunakan notasi Big O. Berikut perhitungan kompleksitas algoritmanya:

Tabel 3-2 Notasi Big O *Pseudocode* Algoritma Semut

1	For each pair (t,v) 1. /* *Initialization phase* */	$O_{(n)}$
2	τ(t,v):= T_0	$O_{(1)}$
3	End-for	
4	For k:=1 to m do	$O_{(n)}$
5	Let tk1 be the starting city for ant k	$O_{(1)}$
6	$J_k(t_{k1})$:= {1, ..., n} - r_{k1}	$O_{(1)}$
7	tk:= tk1	$O_{(1)}$
8	End-for	
9	For i:=1 to n do	$O_{(n)}$
10	If i<n then	$O_{(1)}$
11	For k:=1 to m do	$O_{(n)}$
12	Choose the next city v_k according to Eq.(1) and Eq.(2)	$O_{(1)}$
13	$J_k(s_k):= J_k (t_k) - v_k$	$O_{(1)}$
14	Tourk (i):=(t_k ,v_k)	$O_{(1)}$
15	End-for	
16	Else	
17	For k:=1 to m do	$O_{(n)}$
18	$v_k := t_{k1}$	$O_{(1)}$
19	$Tour_k$ (i):=(t_k ,v_k)	$O_{(1)}$
20	End-for	
21	End-if	
22	For k:=1 to m do	$O_{(n)}$
23	T(t_k ,v_k):=(1-ρ)T(t_k ,v_k)+ ρT_0	$O_{(1)}$
24	$t_k := v_k$ /* *New city for ant k* */	$O_{(1)}$
25	End-for	

26	End-for	
27	For k:=1 to m do	$O_{(n)}$
28	Compute L_k	$O_{(1)}$
29	End-for	
30	Compute L_{best}	$O_{(1)}$
31	For each edge (t,v)	$O_{(n)}$
32	$T(t_k, v_k):=(1-\alpha)\tau(t_k,v_k) + \alpha(L_{best})^{-1}$	$O_{(1)}$
33	End-for	
34	4. If (End_condition = True) then	$O_{(1)}$
35	Print shortest of L_k	$O_{(1)}$
36	Else	$O_{(1)}$
37	goto Phase 2	
38	End-if	

$$F(n) = (O_{(n)} \text{ x } (O_{(1)}) + (O_{(n)} \text{ x } (O_{(1)} + O_{(1)} + O_{(1)})) + (O_{(n)} \text{ x } (O_{(1)} \text{ x } (O_{(n)} \text{ x } (O_{(1)} + O_{(1)} + O_{(1)})))) + (O_{(n)} \text{ x } (O_{(1)} + O_{(1)})) + (O_{(n)} \text{ x } (O_{(1)} + O_{(1)})) + (O_{(n)} \text{ x } (O_{(1)}) + O_{(1)} + (O_{(n)} \text{ x } O_{(1)}) + (O_{(1)} \text{ x } (O_{(1)} + O_{(1)}))$$

$$F(n) = O_{(n)} + O_{(n)} + O_{(n^2)} + O_{(n)} + O_{(n)} + O_{(n)} + O_{(1)} + O_{(n)} + O_{(1)}$$

$$F(n) = O_{(n^2)}$$

3.1.4 Aturan Bisnis Lokasi/Titik

Aturan bisnis adalah aturan-aturan yang berlaku pada sistem yang sedang berjalan, sehingga kinerja perangkat lunak dapat berjalan dengan baik. Aturan-aturan yang ada pada perangkat lunak pencarian jarak terdekat di kota Tasikmalaya yaitu:

1. Jarak antara lokasi/titik dihitung berdasarkan skala peta aslinya.
2. Suatu lokasi/titik hanya dapat dikoneksikan dengan lokasi / titik yang berada disebelahnya.
3. Setiap lokasi/titik mempunyai titik koordinat masing-masing yang ditentukan berdasar pada peta aslinya.
4. Hanya admin yang dapat mengakses *web sercive*.

3.1.5 Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional merupakan bagian untuk menspesifikasikan ukuran yang harus dipenuhi oleh perangkat lunak. Untuk lebih jelasnya dapat dilihat dalam dibawahi ini.

Tabel 3-3 Analisis Kebutuhan Non Fungsional

Kriteria Kebutuhan	Tuntutan Kebutuhan
Performansi	Perangkat PC/Laptop atau mobile untuk proses update pada aplikasi ini harus terkoneksi ke internet
Batasan Memori	Perangkat PC/Laptop: - Prosessor minimum dual core - Hardddisk minimum 80GB - RAM Minimun 512MB memory Perangkat Mobile: - Memory internal minimum 10MB
Kriteria Kebutuhan	Tuntutan Kebutuhan
Modus Operasi	Perangkat PC/Laptop: - Windows XP, Vista atau windows 7 - sudah terpasang aplikasi browser Perangkat Mobile: - Sistem operasi Android - Sudah terpasang aplikasi pencarian jarak terdekat
Kebutuhan Adaptasi Lokasi	-

3.1.5.1. Analisis Kebutuhan Perangkat Lunak

Perangkat lunak yang dibutuhkan dalam menjalankan perangkat lunak ini adalah sebagai berikut:

1. Sistem operasi Windows XP/vista/7.
2. Sistem operasi Android.
3. Eclipse IDE for Java Developers.
4. Xampp win32 v.1.7.1.

3.1.5.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras yang digunakan dalam membangun perangkat lunak ini memiliki spesifikasi sebagai berikut:

1. Processor Intel Core i5
2. Memori RAM 4 Gb
3. Harddisk 1 Tb
4. Resolusi layar monitor 1920 x 1080

Berikut ini adalah spesifikasi minimum untuk menjalankan perangkat lunak ini pada komputer dan perangkat mobile:

1. Intel dual core 1,5Hz
2. Memory 512Mb
3. VGA 256Mb
4. Harddisk 10Mb space
5. Mobile dengan sistem operasi Android.

3.1.5.3. Analisis Kebutuhan Pengguna

Untuk dapat menjalankan aplikasi ini, pengguna harus memenuhi persyaratan sebagai berikut:

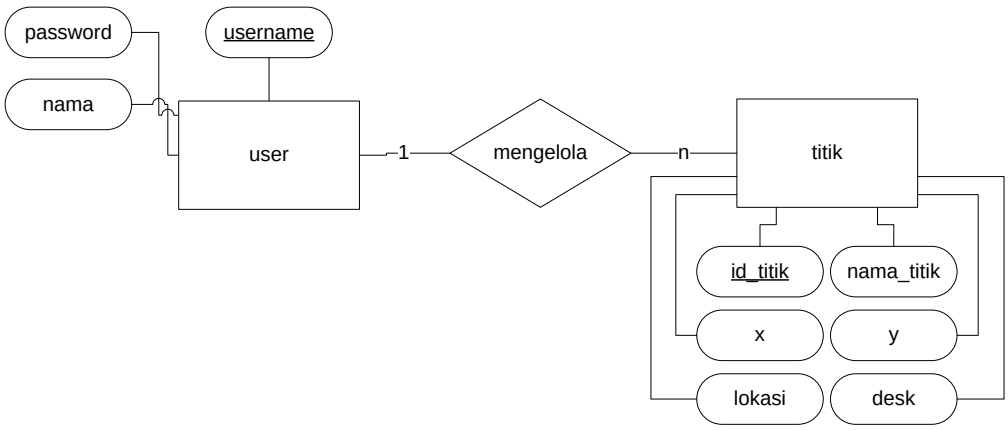
1. Mengerti penggunaan mobile dengan sistem operasi Android.
2. Mengerti secara umum penggunaan internet.
3. Minimal dapar mengatur dan memperbaiki data-data fasilitas umum di *website*.

3.1.5.4. Analisis Data

Sebelum melakukan penyusunan basis data perlu dibuat *Entity Relationship Diagram* (ERD) dari entitas yang terlibat dalam sistem pencarian jarak ini. *Entity – Relationship* adalah suatu hubungan antara dua file atau lebih yang saling berkaitan. Entitas yang saling berhubungan antara satu dengan yang lain akan membentuk suatu relasi dan isi masing-masing entitas tersebut saling melengkapi.

ERD adalah model konseptual yang mendeskripsikan hubungan antar penyimpanan (dalam DFD). ERD digunakan untuk memodelkan struktur data dan hubungan antar data. Dengan ERD kita dapat menguji model dengan mengabaikan proses yang harus dilaksanakan. ERD menggunakan notasi dan sumber untuk menggambarkan struktur dan hubungan antar data.

Untuk lebih jelasnya dapat dilihat ERD dibawah ini:



Gambar 3.5 *Entity Relationship Diagram* (ERD)

Adapun entitas-entitas lain yang terdapat dalam ERD diatas dapat dilihat pada tabel 3.3.

Tabel 3-4 Entitas dan Atribut pada ERD

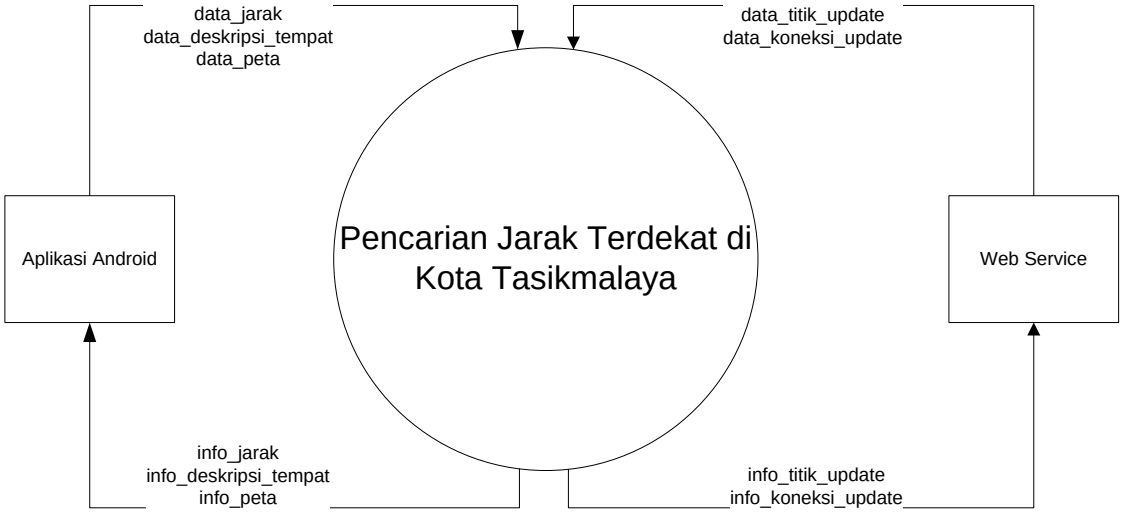
No.	Entitas	Atribut	Keterangan
1.	**user**	username password nama	Nama ID user Kata kunci ID user Nama user
2.	**titik**	id_titik nama_titik x y lokasi desk	Identifikasi titik Nama suatu tempat Titik koordinat x Titik koordinat y Jenis titik Deskripsi tempat

3.1.6 Analisis Kebutuhan Fungsional

Analsis kebutuhan fungsional mendefinisikan aksi dasar yang ada dalam perangkat lunak yang dibangun untuk menerima dan memproses masukan dan menghasilkan keluaran.

3.1.6.1. Diagram Konteks

Diagram konteks adalah diagram yang menggambarkan masukan, proses dan keluaran secara umum yang terjadi pada sistem. Diagram konteks untuk perangkat lunak pencarian jarak terdekat dapat dilihat pada gambar:



Gambar 3.6 Diagram Konteks

3.1.6.2. DFD Level 1 untuk *Web Service*

Pada DFD Level 1 akan dibagi menjadi empat garis besar proses utama, yaitu login, atur titik, atur koneksi dan update peta.

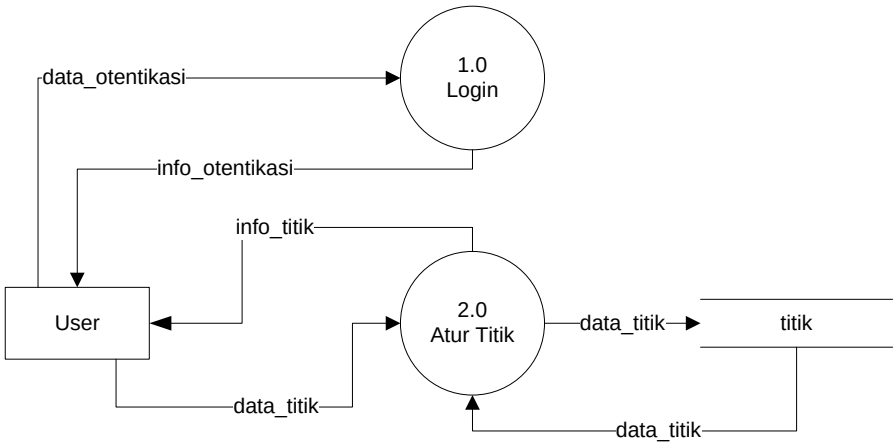
1. Login

   Pada proses pertama user melakukan login terlebih dahulu.

2. Atur titik

   Pada atur titik user dapat melakukan tambah titik, update titik dan hapus titik.

Untuk lebih jelasnya dapat dilihat gambar dibawah ini:



Gambar 3.7 *Data Flow Diagram* Level 1

3.1.6.3. DFD Level 2 Proses 2 untuk *Web Service*

Pada DFD Level 2 Proses 2 akan dibagi menjadi tiga proses, yaitu tambah titik, update titik dan hapus titik.

1. Tambah titik

   Pada proses ini user menambahkan daftar titik yang berisi ID titik, nama titik, koordinat x, koordinat y, tipe dan deskripsi.

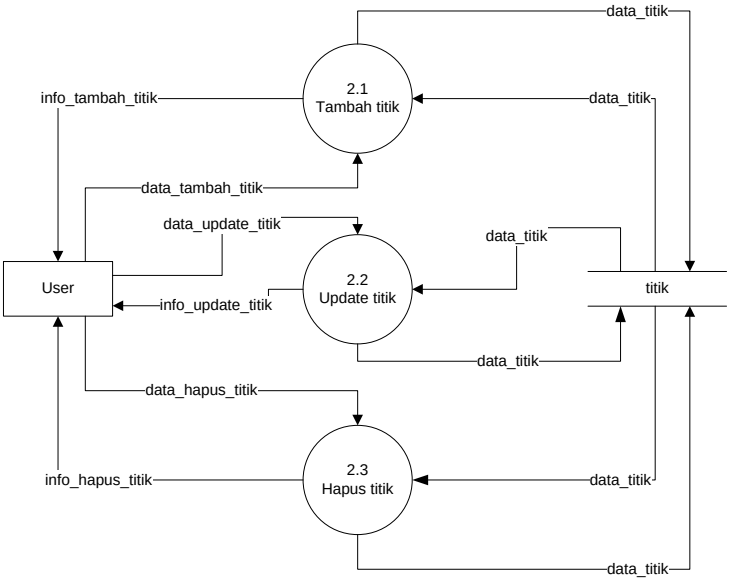
2. Update titik

   Pada proses ini user mengedit daftar titik yang berisi ID titik, koordinat x, koordinat y, tipe dan deskripsi.

3. Hapus titik

   Pada prose ini user menghapus daftar titik yang berisi ID titik, koordinat x, koordinat y, tipe dan deskripsi.

Untuk lebih jelasnya dapat dilihat gambar dibawah ini:



Gambar 3.8 *Data Flow Diagram* Level 2 Proses 1

Berikut adalah spesifikasi proses dari DFD untuk menjelaskan perilaku-perilaku yang ada dalam DFD secara lebih rinci. Spesifikasi proses pada perangkat lunak pencarian jarak terdekat dapat dilihat pada Tabel 3.5 berikut:

Tabel 3-5 Spesifikasi Proses DFD Level 1

NO.	Spesifikasi	Keterangan
1.	No	1.0
	Nama	Login
	Input	data_otentikasi
	Output	info_otentikasi
	Tujuan	user
	Logika proses	1. User akan memasukkan username dan password 2. Sistem akan masuk kedalam administrator.
2.	No	2.0
	Nama	Atur titik
	Input	data_titik
	Output	info_titik
	Tujuan	user
	Logika proses	1. User dapat mengatur daftar titik yang berisi ID titik, nama titik, koordinat x, koordinat y, tipe dan deskripsi. 2. Sistem akan menyimpan daftar titik yang telah diatur oleh user.
3.	No	2.1
	Nama	Tambah titik
	Input	Data_titik
	Output	Info_titik
	Tujuan	User
	Logika proses	1. User dapat menambah titik. 2. Sistem menyimpan titik baru yang dimasukan.
4.	No	2.2
	Nama	Update titik
	Input	Data_update_titik
	Output	Info_update_titik
	Tujuan	User
	Logika proses	1. User dapat mengupdate titik. 2. Sistem mengupdate titik yang sudah ada.
5.	No.	2.3
	Nama	Hapus titik
	Input	Data_hapus_titik
	Output	Info_hapus_titik
	Tujuan	User

	Logika proses	1. User dapat menghapus titik yang sudah ada.

3.1.6.4. Kamus Data

Kamus data membantu mendefinisikan dengan lengkap data yang mengalir diantara proses, penyimpanan data dan entitas luar. Data yang mengalir tersebut dapat berupa masukan untuk sistem atau hasil dari proses sistem. Kamus data dibuat berdasarkan arus data yang mengalir pada konteks diagram dan DFD.

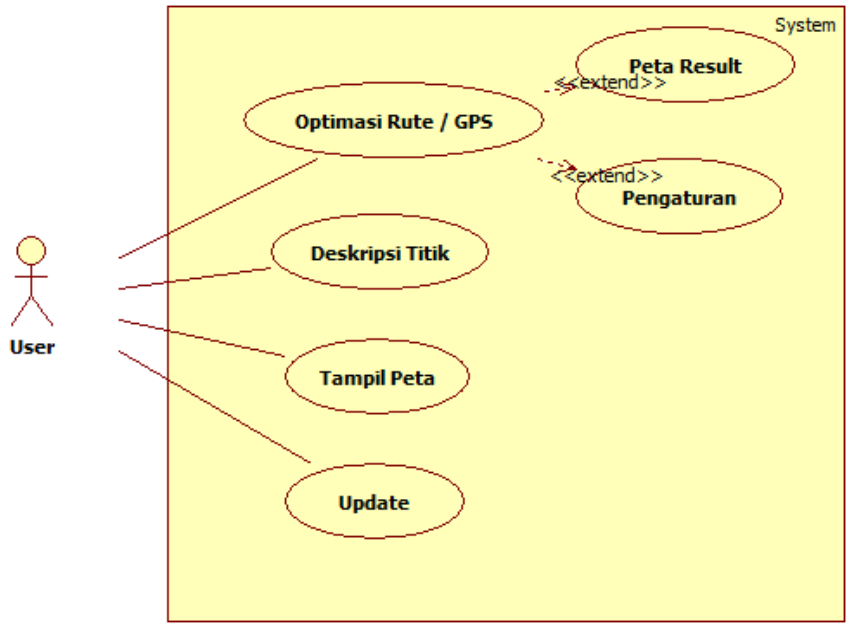
Tabel 3-6 Kamus Data

NO.	Data	Keterangan
1.	**Nama**	data_user
	Deskripsi	Berisi data user
	Struktur data	Username + password + nama
	Username	[A-Z|a-z|0-9]
	password	[A-Z|a-z|0-9]
	nama	[A-Z|a-z|0-9]
2.	**Nama**	data_titik
	Deskripsi	Berisi data titik
	Struktur data	id_titik + nama_titik + x + y + lokasi +desk
	id_titik	[0-9]
	nama_titik	[A-Z|a-z|0-9]
	X	[0-9]
	Y	[0-9]
	lokasi	[A-Z|a-z|0-9]
	desk	[A-Z|a-z|0-9]
3.	**Nama**	data_titik
	Deskripsi	Berisi data titik
	Struktur data	Id_titik + nama_titik + x + y + lokasi + desk
	Id_titik	[0-9]
	Nama_titik	[A-Z|a-z|0-9]
	x	[0-9]
	y	[0-9]
	lokasi	[0-9]
	desk	[A-Z|a-z|0-9]
4.	**Nama**	Data_hapus_titik
	Deskripsi	Berisi data hapus titik

	Struktur data	Data_hapus_titik
	Data_hapus_titik	[true|false]

3.1.6.5. *Use Case Diagram* untuk Aplikasi Android

Use case diagram perangkat lunak pencarian jarak terdekat ini dapat dilihat pada gambar 3.9.



Gambar 3.9 *Use Case* Diagram

3.1.6.6. Skenario *Use Case* untuk Aplikasi Android

Skenario *Use Case* berfungsi untuk menjelaskan lebih detail mengenai modul-modul yang terdapat dalam perangkat lunak. Dibawah ini adalah skenario *use case* yang terjadi pada sistem pencarian jarak terdekat.

Tabel 3-7 Skenario *Use Case* Optimasi Jarak

Identifikasi	
Use Case Name	Optimasi Rute
Actor	User
Use Case Description	Melakukan pencarian jarak terdekat dari titik asal ke titik tujuan
Trigger	User memasukan tempat asal dan tempat tujuan kemudian menekan tombol proses.
Trigger Type	☐ Internal ☒ External
Main Success Scenario	
1. User memilih menu optimasi rute	
2. User memasukan tempat asal & tujuan	
3. User menekan tombol Proses	
Extension	
a. Jika titik tujuan atau titik asal tidak dikenal a.1 User mendapatkan pesan tidak ada titik tujuan terdefinisi	

Tabel 3-8 Skenario *Use Case* Optimasi Jarak GPS

Identifikasi	
Use Case Name	Optimasi Rute
Actor	User
Use Case Description	Melakukan pencarian jarak terdekat ke titik tujuan
Trigger	User memasukan tempat tujuan kemudian menekan tombol proses.
Trigger Type	☐ Internal ☒ External
Main Success Scenario	
1. User memilih menu optimasi rute	
2. User memasukan tempat tujuan	

3. User menekan tombol Proses
Extension
a. Jika titik tujuan tidak dikenal a.1 User mendapatkan pesan tidak ada titik tujuan terdefinisi

Tabel 3-9 Skenario *Use Case* Deskripsi Titik

Identifikasi	
Use Case Name	Deskripsi Titik
Actor	User
Use Case Description	Menunjukan informasi tentang suatu titik di kota Tasikmalaya
Trigger	User memasukan titik yang akan dideskripsikan
Trigger Type	[ ] Internal [X] External
Main Success Scenario	
1 User memilih menu deskripsi tempat 2 User memasukkan suatu titik di kota Tasikmalaya	
Extension	
a. Jika tidak memasukkan titik a.1 User tidak akan mendapatkan deskripsi titik	

Tabel 3-10 Skenario *Use Case* Lihat Peta

Identifikasi	
Use Case Name	Tampil Peta
Actor	User
Use Case Description	Menunjukan peta Kota Tasikmalaya
Trigger	User berada di menu utama dan memilih menu lihat peta
Trigger Type	[ ] Internal [X] External
Main Success Scenario	

1. User memilih menu lihat peta
Extension

Tabel 3-11 Skenario *Use Case* Lihat Peta

Identifikasi	
Use Case Name	Update
Actor	User
Use Case Description	Sistem mengupdate titik, koneksi dan jarak
Trigger	User berada di menu utama dan memilih menu lihat peta
Trigger Type	☐ Internal ☒ External
Main Success Scenario	
1. User memilih menu update	
Extension	

Tabel 3-12 Skenario *Use Case* Pengaturan

Identifikasi	
Use Case Name	Pengaturan
Actor	User
Use Case Description	User mengatur parameter algoritma semut
Trigger	Memilih menu optimasi rute kemudian pilih pengaturan
Trigger Type	☐ Internal ☒ External
Main Success Scenario	

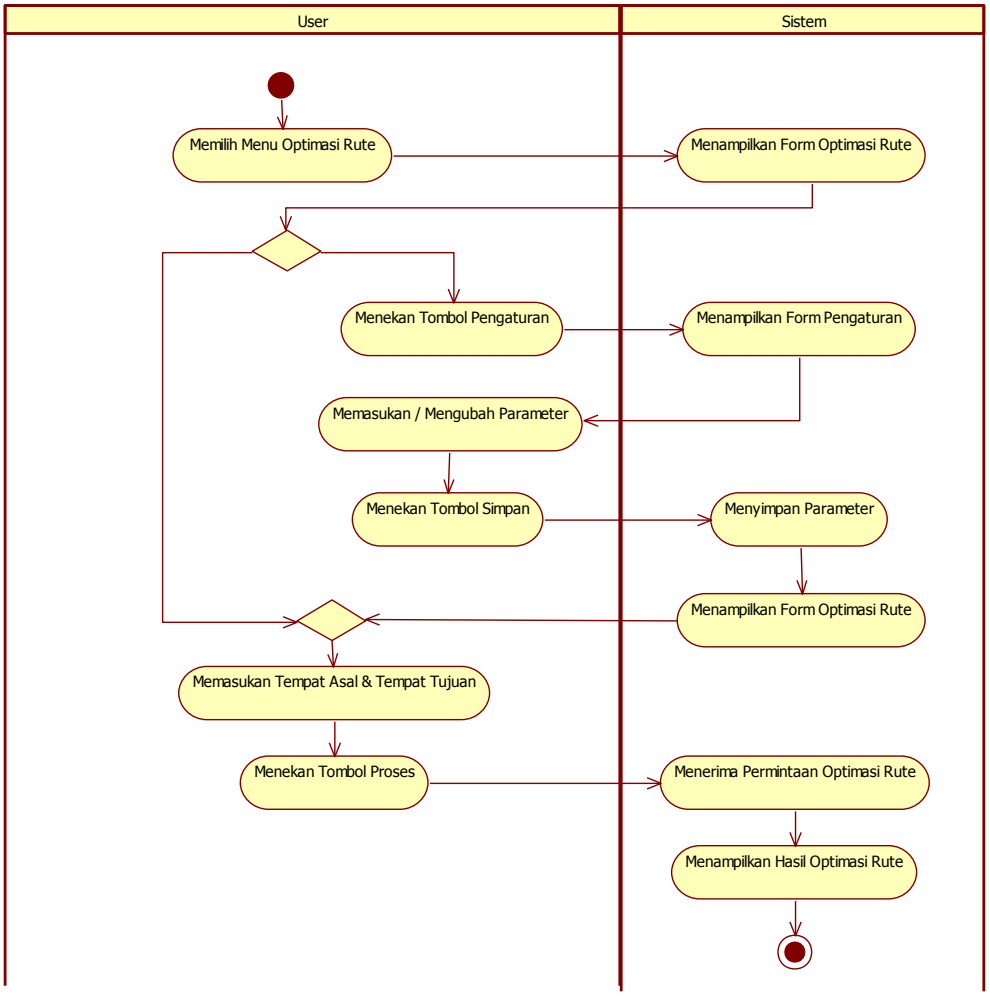
1. User memilih pengaturan
2. User mengatur parameter algoritma semut
3. User menyimpan parameter
Extension
a. Jika parameter tidak dimasukan a.1 user akan mendapatkan pesan alpha harus diisi a.2 user akan mendapatkan pesan betha harus diisi a.3 user akan mendapatkan pesan Q harus diisi a.4 user akan mendapatkan pesan P harus diisi a.5 user akan mendapatkan pesan siklus harus diisi a.6 user akan mendapatkan pesan semut harus diisi

3.1.6.7. Activity Diagram untuk Aplikasi Android

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

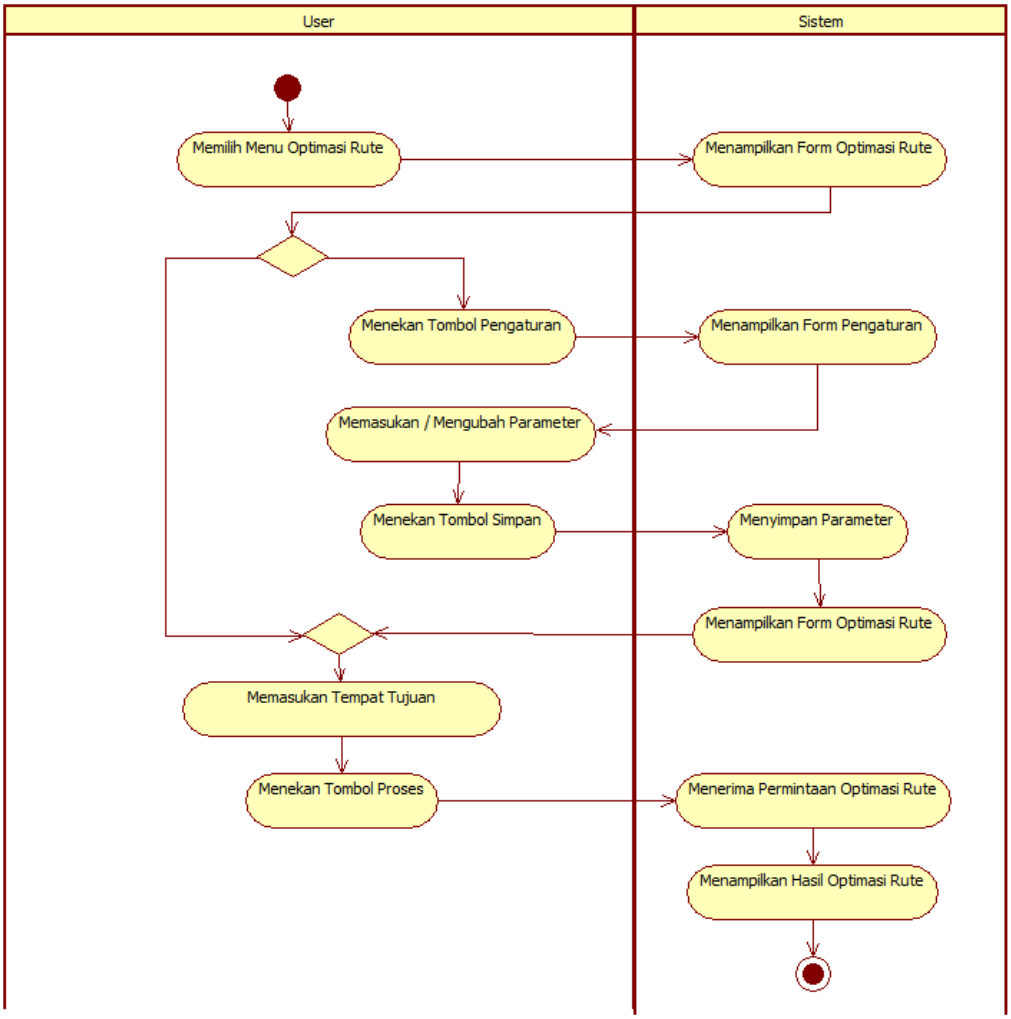
Berikut *Activity diagram* yang ada pada perangkat lunak pencarian jarak terdekat ini:

1. Optimasi Rute



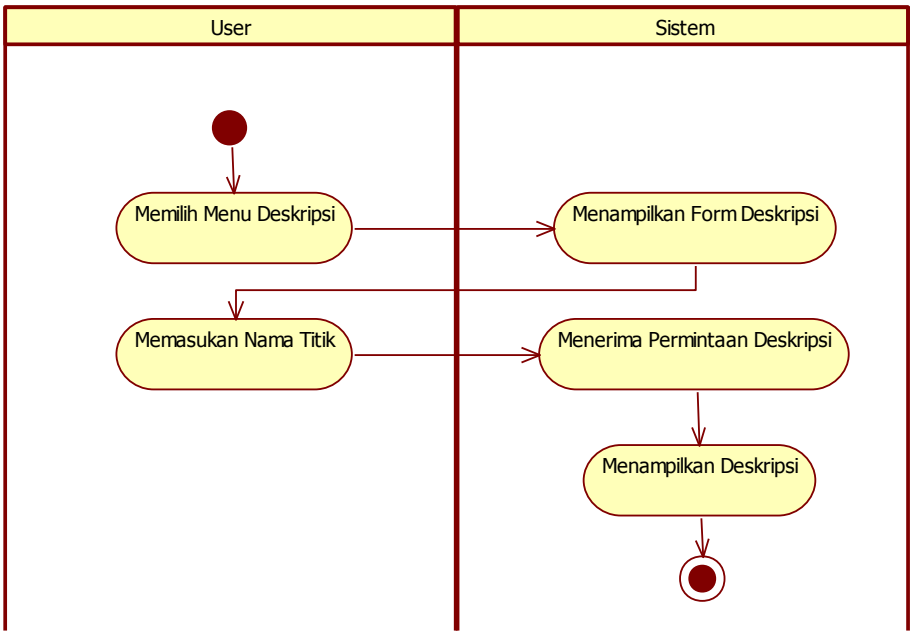
Gambar 3.10 *Activity Diagram* Optimasi Rute

2. Optimasi Rute GPS



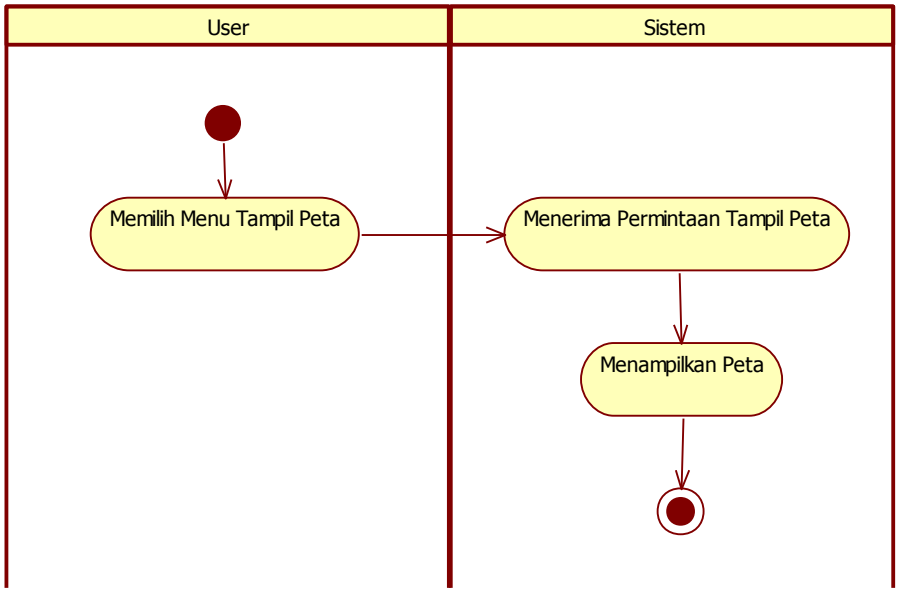
Gambar 3.11 *Activity Diagram* Optimasi Rute GPS

2. Deskripsi Titik



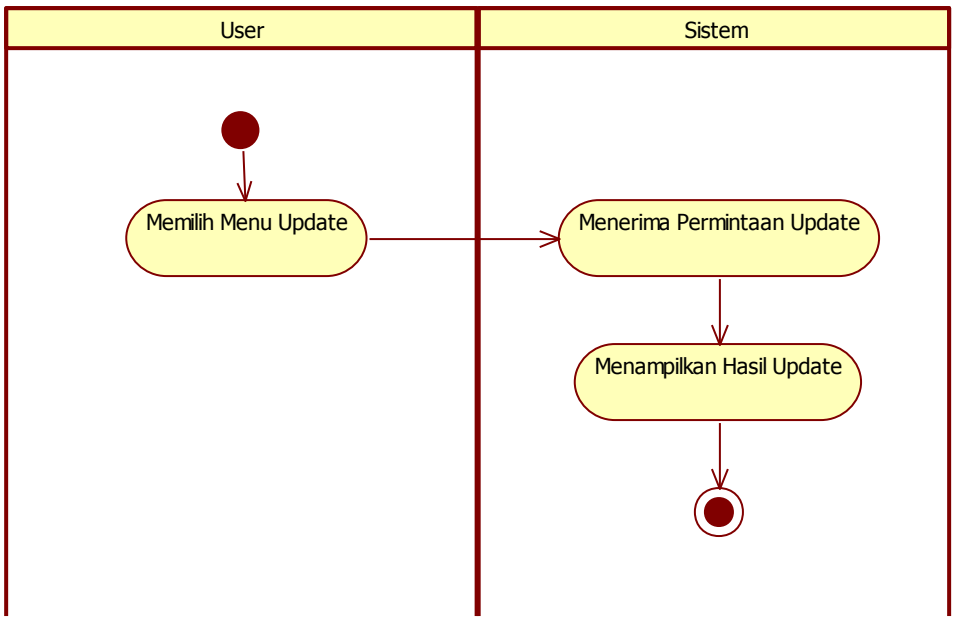
Gambar 3.12 *Activity Diagram* Deskripsi Titik

3. Tampil Peta



Gambar 3.13 *Activity Diagram* Tampil Peta

4. Update



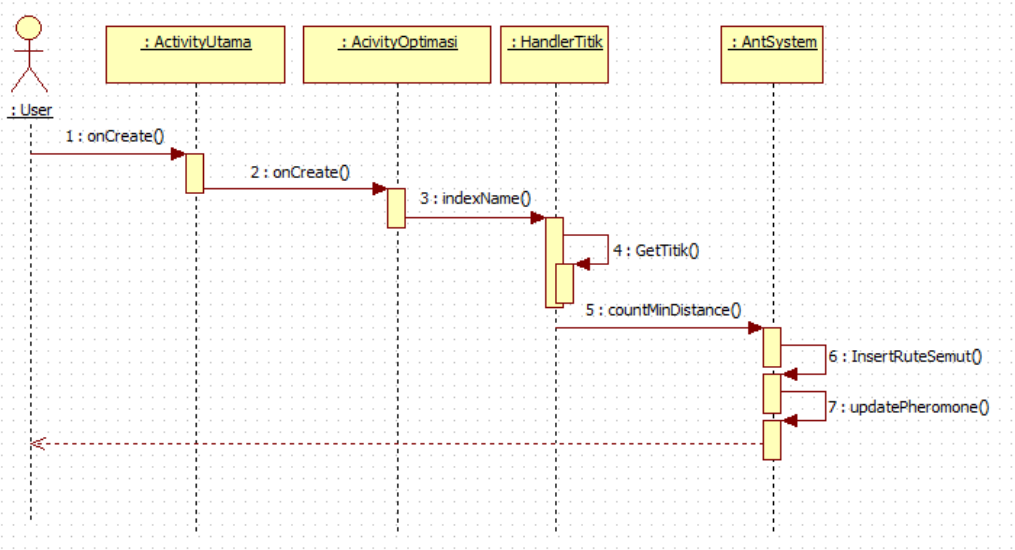
Gambar 3.14 *Activity Diagram* Update

3.1.6.8. *Sequence Diagram* untuk Aplikasi Android

Sequence diagram menggambarkan interaksi antar objek didalam dan disekitar sistem berupa message yang digambarkan terhadap waktu. *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.

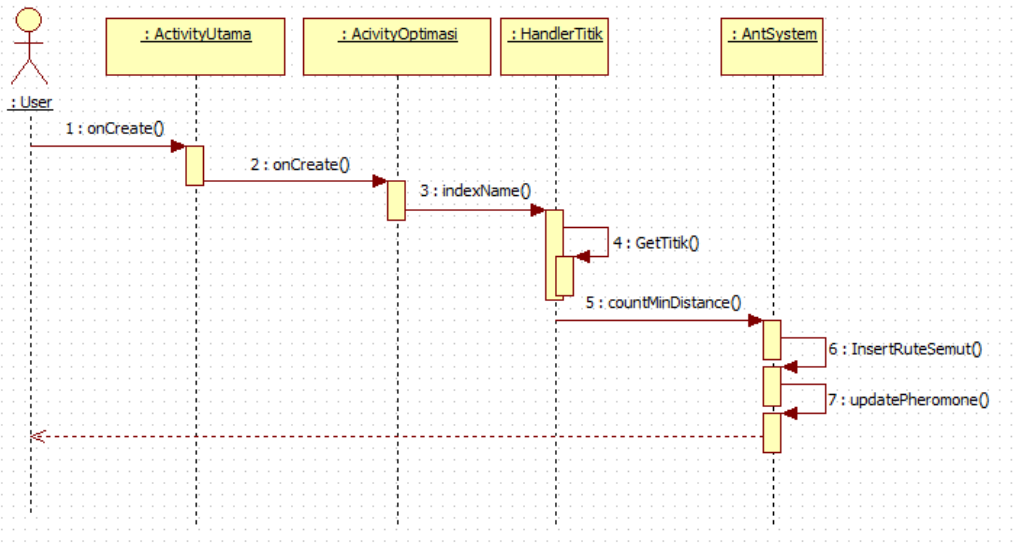
Sequence diagram pada perangkat lunak pencarian jarak terdekat ini terbagi menjadi beberapa bagian, yaitu:

1. Optimasi rute



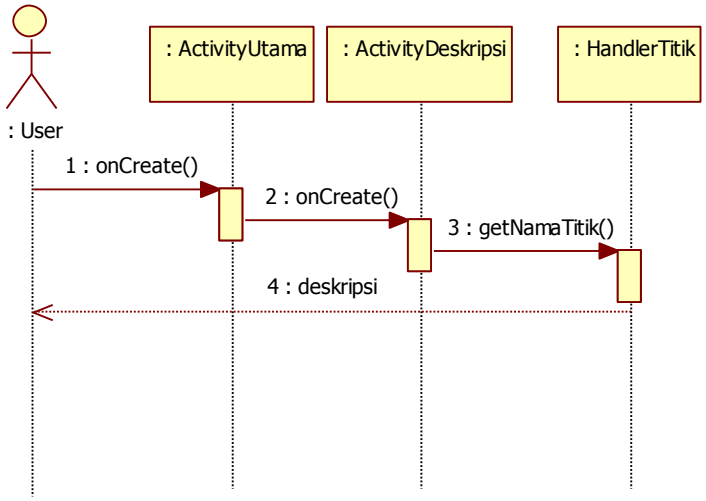
Gambar 3.15 *Sequence Diagram* Optimasi Jarak

2. Optimasi Rute GPS



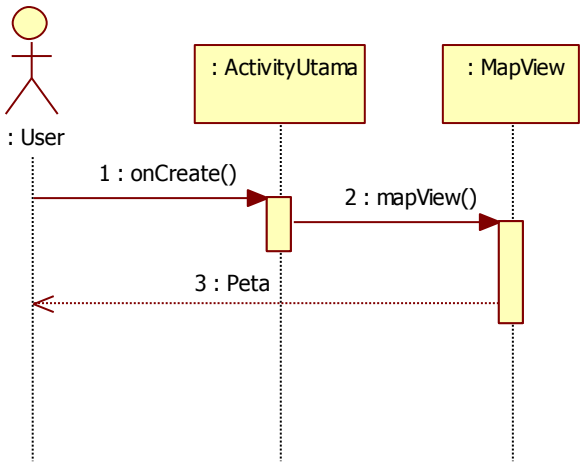
Gambar 3.16 *Sequence Diagram* Optimasi Jarak GPS

3. Deskripsi Titik



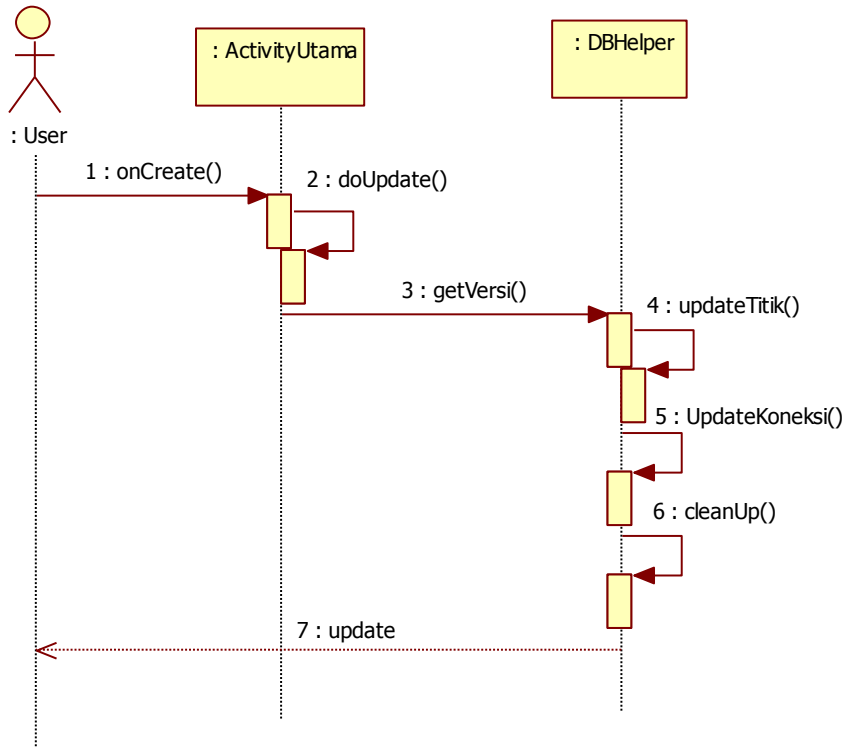
Gambar 3.17 *Sequence Diagram* Deskripsi Tempat

4. Tampil Peta



Gambar 3.18 *Sequence Diagram* Lihat Peta

5. Update

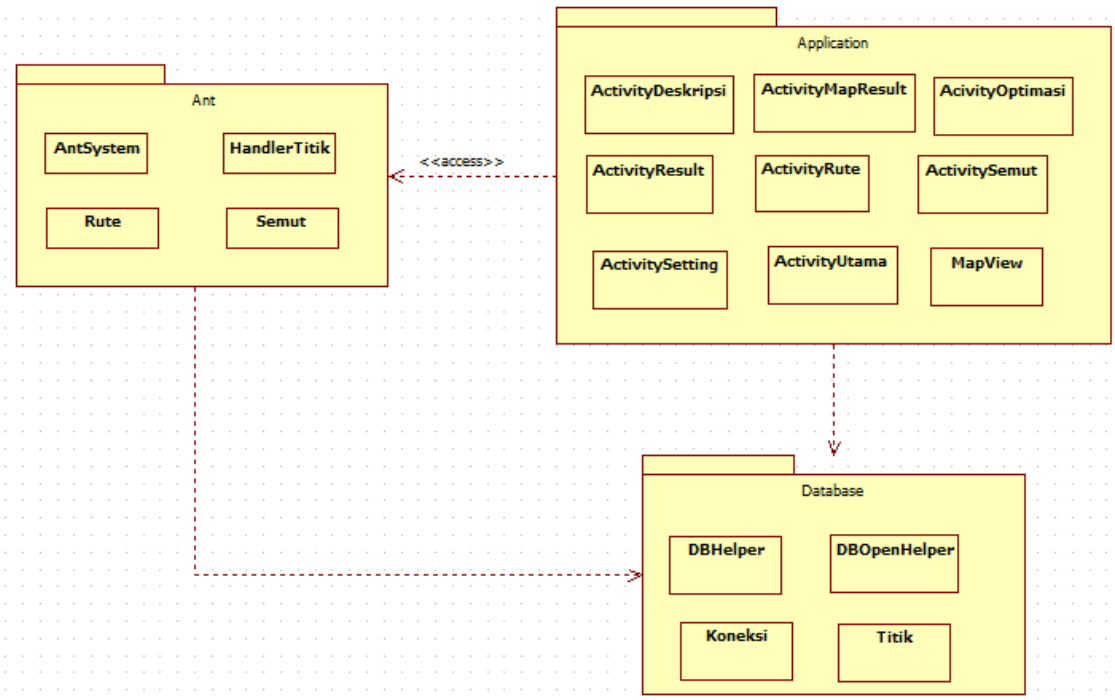


Gambar 3.19 Sequence Diagram *Update*

3.1.6.9. *Class Diagram* untuk Aplikasi Android

Class adalah sebuah spesifikasi yang jika diinstansisikan 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).

Berikut *class diagram* yang terbentuk pada perangkat lunak pencarian jarak terdekat.



Gambar 3.20 *Class Diagram*

3.2 Perancangan Sistem

Perancangan merupakan tahap lanjutan dari analisis sistem dimana pada perancangan sistem digambarkan rancangan sistem yang akan dibangun sebelum dilakukan pengkodean kedalam suatu bahasa pemrograman. Dalam perancangan sistem tidak lepas dari hasil analisis, karena dari hasil analisis sistem baru dapat dibuat suatu perancangan sistem.

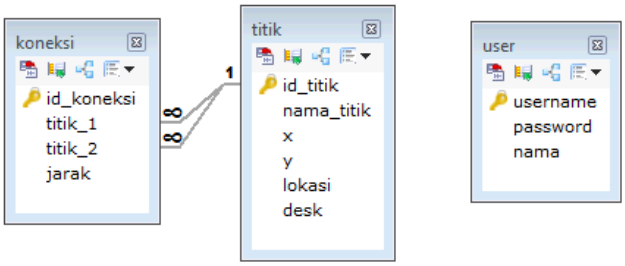
Desain umum yang akan diaplikasikan bertujuan untuk memberikan gambaran secara umum kepada penggunaan tentang sistem yang akan dibangun.

3.2.1 Perancangan Data

Perancangan data terdiri dari perancangan diagram relasi berdasarkan analisis basis data dan struktur tabel yang dibangun dalam sistem.

1.2.1.1 Diagram Relasi

Diagram relasi merupakan relasi khusus antar tabel yang berada dalam perangkat lunak yang dibangun. Berikut diagram relasi yang berada dalam perangkat lunak pencarian jarak terdekat.



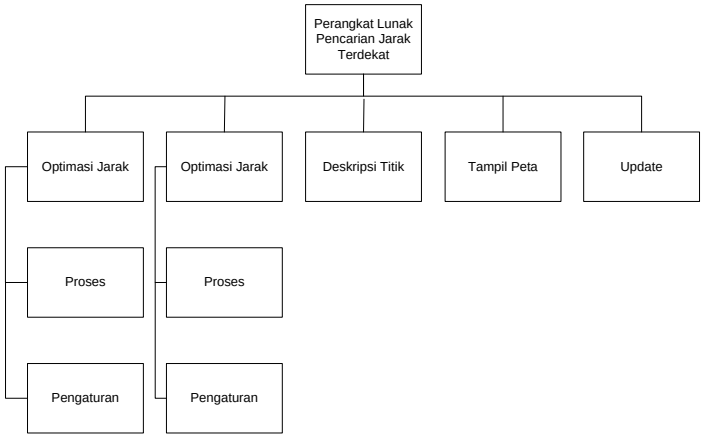
Gambar 3.21 Tabel Relasi

3.2.2 Perancangan Struktur Menu

Perancangan struktur menu pada perangkat lunak pencarian jarak terdekat ini terbagi menjadi dua, yaitu pada aplikasi android dan pada *web service.*

3.2.2.1 Aplikasi Android

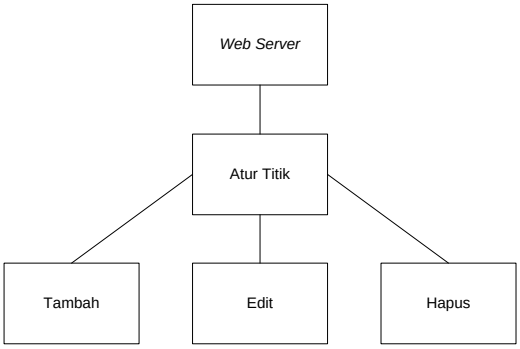
Struktur menu pada aplikasi android digambarkan dengan struktur sebagai berikut:



Gambar 3.22 Struktur Menu Aplikasi Android

3.2.2.2 *Web Service*

Struktur menu pada aplikasi android digambarkan dengan struktur sebagai berikut:



Gambar 3.23 Struktur Menu *Web Service*

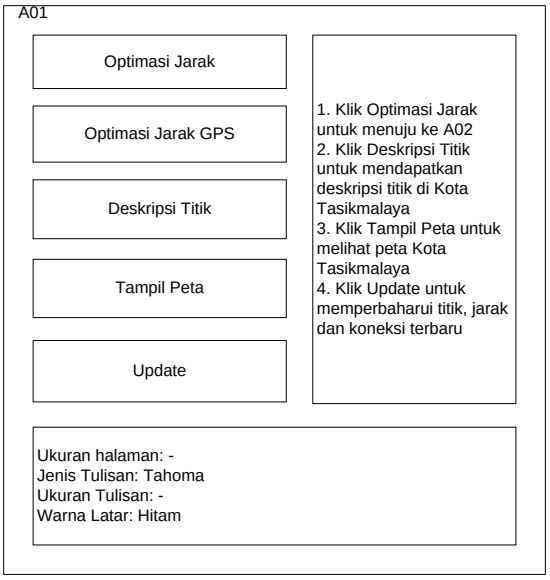
3.2.3 Perancangan Antarmuka Perangkat Lunak

Antarmuka merupakan tampilan dari suatu perangkat lunak yang berperan sebagai media komunikasi yang digunakan sebagai sarana antara perangkat lunak dan pengguna. Perancangan ini merupakan sebuah penggambaran, perencanaan dan pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan utuh.

Sistem yang dibangun diharapkan menyediakan suatu tampilan antarmuka yang mudah dipahami dan digunakan oleh pengguna.

3.2.3.1 Perancangan Antaramuka pada Aplikasi Android

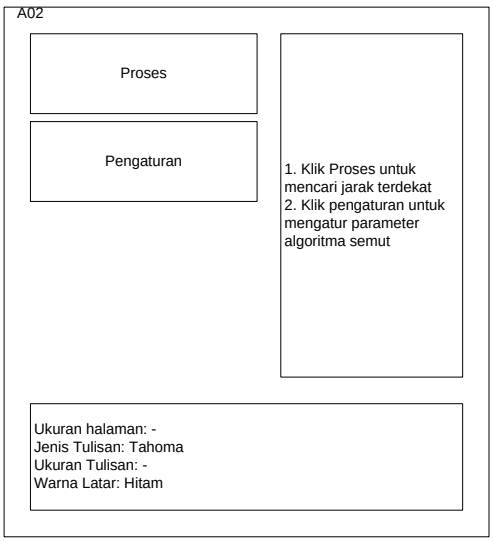
Berikut ini dapat dilihat perancangan antarmuka Ant System pada aplikasi Android. Menu Ant System merupakan halaman pertama yang akan muncul ketika perangkat lunak dieksekusi oleh pengguna. Perancangan antarmuka ini dapat dilihat digambar 3.24.



Gambar 3.24 Perancangan Antarmuka Halaman Utama

3.2.3.2 Perancangan Antarmuka Optimasi Jarak

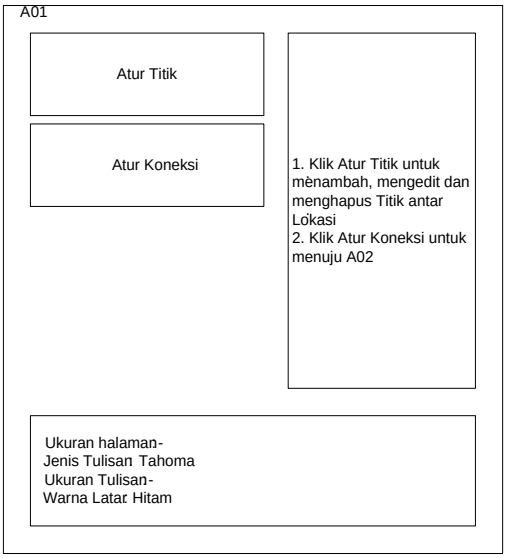
Berikut ini adalah gambar perancangan antarmuka halaman Optimasi Jarak. Halaman Optimasi Jarak ini akan muncul apabila pengguna memilih menu Optimasi Jarak. Perancangan ditampilkan pada Gambar 3.25.



Gambar 3.25 Perancangan Antarmuka Optimasi Jarak

3.2.3.3 Perancangan Antaramuka pada *Web Service*

Berikut ini dapat dilihat perancangan antarmuka pada web service. Menu Atur Titik merupakan halaman pertama yang akan muncul ketika web service dieksekusi oleh admin. Perancangan antarmuka ini dapat dilihat digambar 3.26.



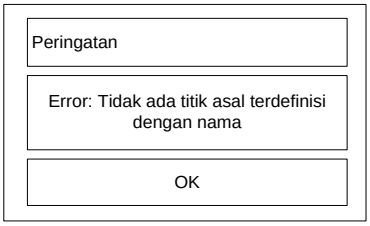
Gambar 3.26 Perancangan Antarmuka pada *Web Service*

3.2.4 Perancangan Pesan

Pesan merupakan tampilan dari suatu perangkat lunak yang berfungsi untuk menyampaikan notifikasi dan informasi kepada pengguna agar perangkat lunak lebih interaktif.

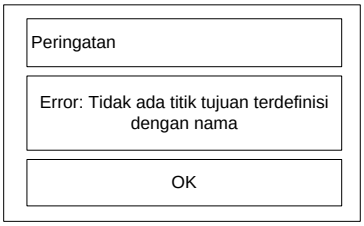
Berikut perancangan pesan yang ada pada aplikasi Android:

1 Rancangan Pesan yang akan muncul pada saat tidak ada titik asal terdefinisi.



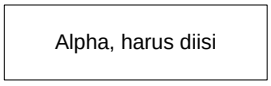
Gambar 3.27 Pesan M01

2 Rancangan Pesan yang akan muncul pada saat tidak ada titik tujuan terdefinisi.



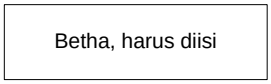
Gambar 3.28 Pesan M02

3 Rancangan Pesan yang akan muncul pada saat Alpha tidak diisi.



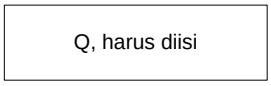
Gambar 3.29 Pesan M03

4 Rancangan Pesan yang akan muncul pada saat Betha tidak diisi.



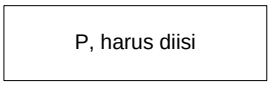
Gambar 3.30 Pesan M04

5 Rancangan Pesan yang akan muncul pada saat Q tidak diisi.



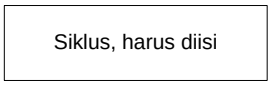
Gambar 3.31 Pesan M05

6 Rancangan Pesan yang akan muncul pada saat P tidak diisi.



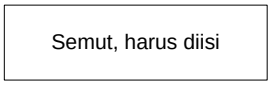
Gambar 3.32 Pesan M06

7 Rancangan Pesan yang akan muncul pada saat Siklus tidak diisi.



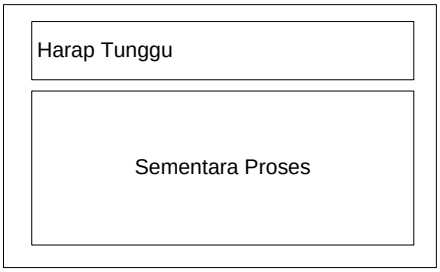
Gambar 3.33 Pesan M07

8 Rancangan Pesan yang akan muncul pada saat Semut tidak diisi.



Gambar 3.34 Pesan M08

9 Rancangan Pesan yang akan muncul pada saat Update.



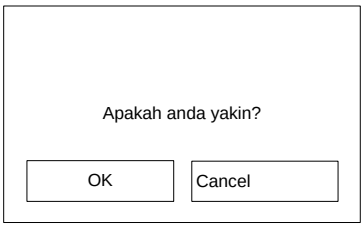
Gambar 3.35 Pesan M09

Berikut perancangan pesan yang ada pada aplikasi *Web Service*:

Tabel 3-13 Rancang pesan pada aplikasi *Web Service*

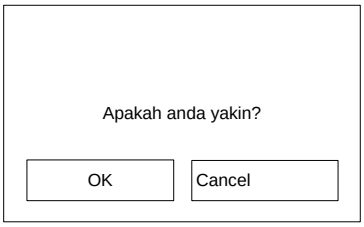
No. Pesan	Isi Pesan	Jenis Pesan	Form
M09	Masukan Nama Titik, jika tipe adalah lokasi	Rancang pesan yang akan muncul pada saat nama titik belum dimasukan	Atur titik
M10	Masukan Koordinat dari titik	Rancang pesan yang akan muncul pada saat titik koordinat belum dimasukan	Atur titik

Rancangan Pesan yang akan muncul pada saat titik akan dihapus.



Gambar 3.36 Pesan M12

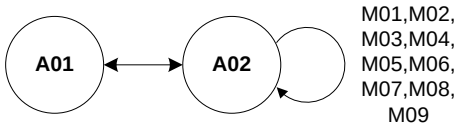
Rancangan Pesan yang akan muncul pada saat koneksi akan dihapus.



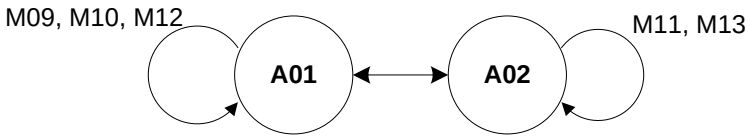
Gambar 3.37 Pesan M13

3.2.5 Jaringan Semantik

Jaringan semantik adalah diagram yang menggambarkan aliran-aliran menu dan pesan dalam sebuah program. Jaringan semantik perangkat lunak pencarian jarak terdekat dapat dilihat pada gambar dibawah ini.



Gambar 3.38 Jaringan Semantik pada Aplikasi Android



Gambar 3.39 Jaringan Semantik pada *Web Service*

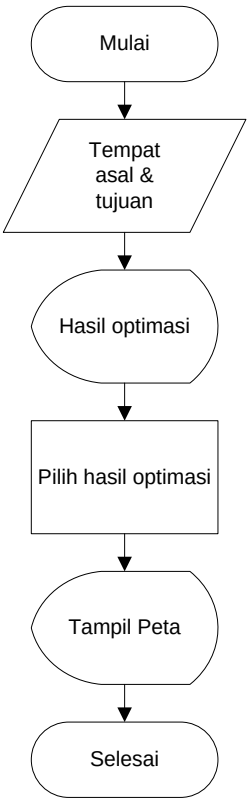
3.2.6 Perancangan Prosedural

Langkah terakhir dalam perancangan yaitu merancang prosedural. Perancangan prosedural digunakan sebagai algoritma dasar yang akan

diimplementasikan ke dalam sistem. Perancangan prosedural untuk perangkat lunak pencarian jarak terdekat adalah sebagai berikut.

3.2.6.1 Prosedural Optimasi Jarak pada Aplikasi Android

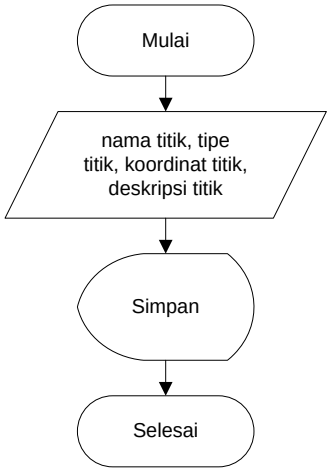
Prosedural optimasi jarak menjelaskan langkah-langkah untuk proses optimasi jarak. Dimulai dengan memasukan tempat asal dan tujuan, kemudian memilih hasil optimasi, hingga menghasilkan tampil perta. Berikut rancangan prosedural optimasi dilihat pada gambar 3.40.



Gambar 3.40 Prosedural Optimasi Jarak

3.2.6.2 Prosedural Atur Titik pada *Web Service*

Prosedural atur titik menjelaskan langkah-langkah untuk menambah, mengedit dan menghapus suatu titik. Dimulai dengan menambahkan titik baru yang berisi nama titik, tipe titik, koordinat titik dan deskripsi titik. Berikut rancangan prosedural atur titik dapat dilihat pada gambar 3.41.



Gambar 3.41 Prosedural Atur Titik