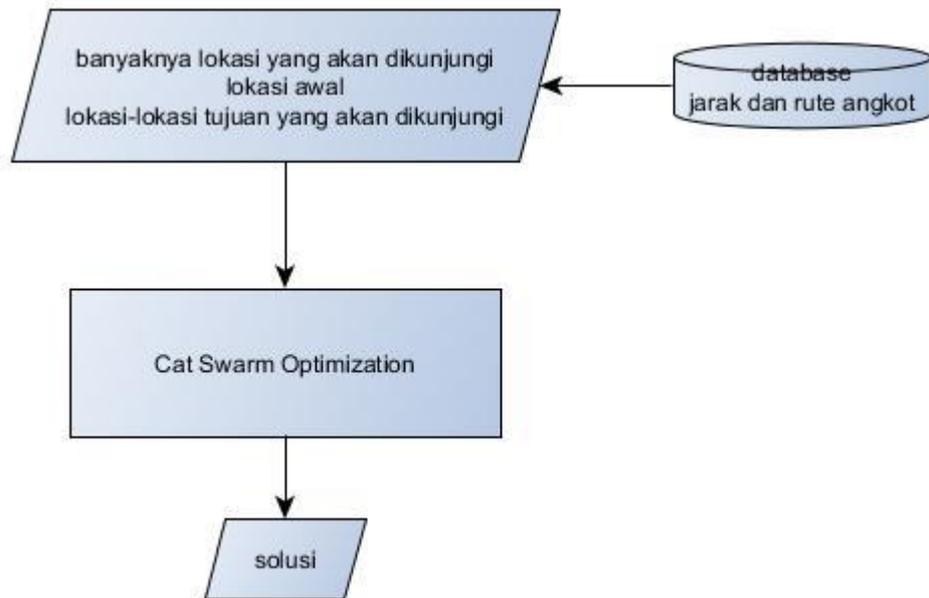


adalah lokasi awal / keberangkatan dan lokasi-lokasi yang ingin dikunjungi, dan sistem akan memberikan solusi berupa rute mana yang sebaiknya digunakan.



Gambar 3. 1 Flowchart analisis sistem

3.2.1 Data Masukan dan Sumber Data

Dalam tahap ini, jumlah tempat yang akan dikunjungi, lokasi awal dan lokasi-lokasi tujuan akan dimasukkan ke dalam sistem. Data masukannya adalah sebagai berikut

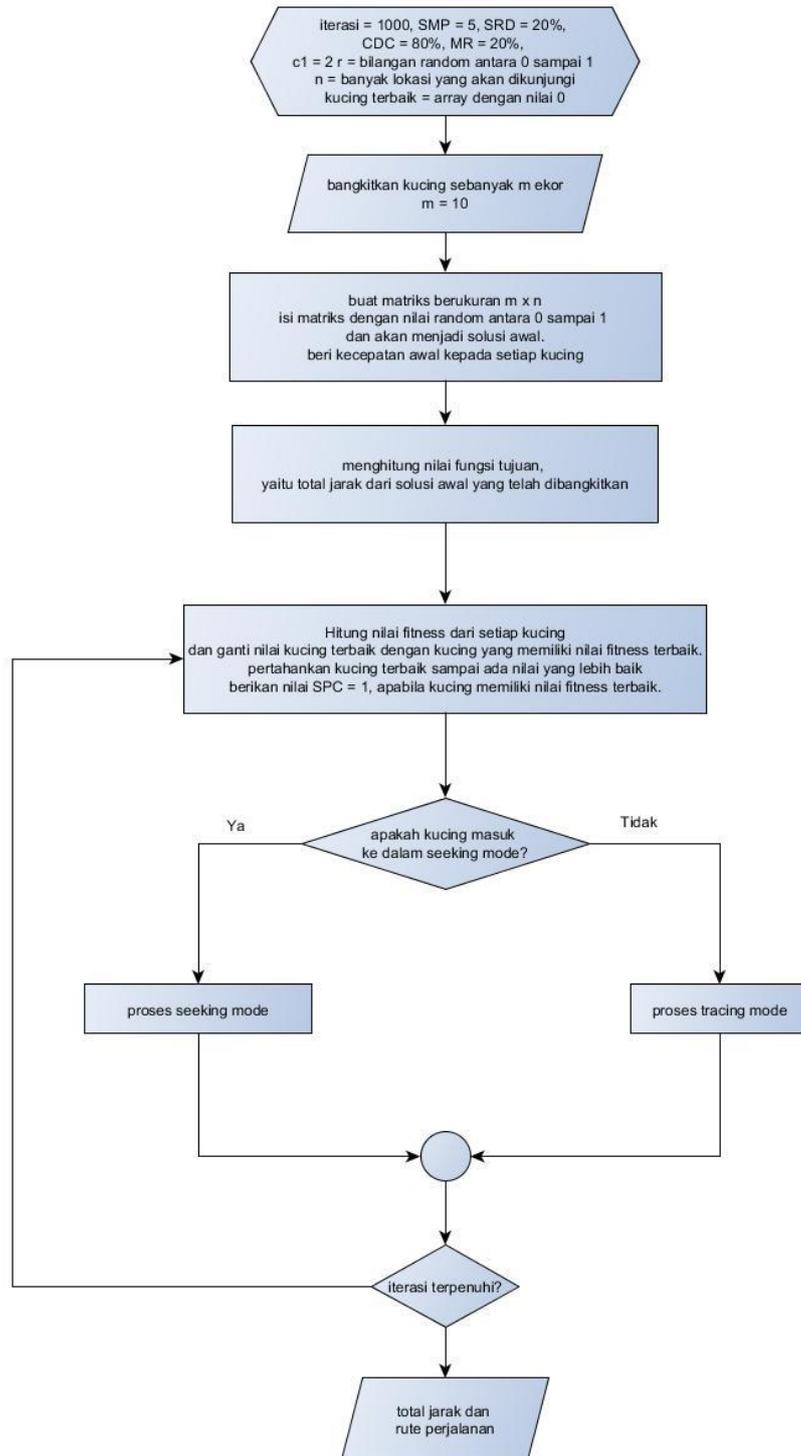
1. Bilangan bulat sebagai jumlah lokasi yang akan dikunjungi. Misal lokasi yang akan dikunjungi adalah 3 tempat, maka $n = 3$.
2. Nama tempat sebagai lokasi awal. Lokasi awalnya adalah UNIKOM.
3. Nama tempat sebagai lokasi-lokasi tujuan.
 1. Lokasi Tujuan 1 = Bandung Indah Plaza
 2. Lokasi Tujuan 2 = Cihampelas Walk
 3. Lokasi Tujuan 3 = Museum Geologi

Sumber data yang digunakan merupakan data yang diambil dari website <https://kiri.travel/#> dan <https://angkot.tibandung.com/> yang kemudian dijadikan informasi rute dari satu lokasi ke lokasi lainnya beserta jaraknya.

Batasan data masukan banyaknya lokasi tujuan adalah minimal 2 lokasi tujuan dan maksimal 4 lokasi tujuan seperti yang dikatakan Tri Basuki Joewono[12] dengan rata-rata perjalanan perhari individu adalah 2.4 perjalanan.

3.2.2 Analisis Metode

Tahap ini merupakan tahap dimana data yang telah diambil dari database akan digunakan ke dalam Algoritma Cat Swarm Optimization.



Gambar 3. 2 Flowchart Cat Swarm Optimization

Adapun penjelasan dari flowchart pada gambar 3.2 adalah sebagai berikut

Langkah 1

Inisialisasi semua parameter yang dibutuhkan, seperti MR, SMP, SRD, CDC, MR, c_1 , r dan juga n . Nilai-nilai yang digunakan adalah seperti yang digunakan oleh Shu-Chuan Chu dan Pei-Wei Tsai [2]. Inisialisasi kucing terbaik dengan nilai 0.

Langkah 2

Buat matriks berukuran $m \times n$ yang menggambarkan m sebagai banyaknya kucing dan n sebagai banyaknya lokasi yang akan dikunjungi, m bernilai 10 dan n dimisalkan 3, maka buatlah matriks berukuran 10×3 . Lalu isi matriks tersebut dengan nilai random dengan rentang dari 0 sampai 1 dengan satu angka dibelakang koma yang selanjutnya isi dari matriks ini sebagai solusi awal yang dapat dilihat pada tabel 3.1. Kecepatan awal kucing pun berupa matriks $m \times n$ yang dapat dilihat pada tabel 3.2.

Tabel 3. 1 Populasi Kucing (solusi awal)

Kucing ke	Lokasi 1	Lokasi 2	Lokasi 3
1	0.3	0.1	0.5
2	0.4	0.5	0.2
3	0.9	0.4	0.1
4	0.6	0.2	0.7
5	0.2	0.1	0.1
6	0.2	0.3	0.9
7	0.7	0.9	0.5
8	0.5	0.2	0.6
9	0.7	0.3	0.6
10	0.4	0.1	0.2

Tabel 3. 2 Kecepatan Awal Kucing

Kucing ke-	Kecepatan lokasi 1	Kecepatan lokasi 2	Kecepatan lokasi 3
1	0.2	0.1	0.7
2	0.8	0.8	0.3
3	0.7	0.2	0.5
4	0.2	0.4	0.7
5	0.6	0.1	0.1
6	0.3	0.7	0.4
7	0.6	0.3	0.6
8	0.4	0.6	0.1
9	0.8	0.2	0.8
10	0.1	0.3	0.6

Langkah 3

Hitung nilai fungsi tujuan dari matriks solusi awal yang telah di dapat, nilai yang terdapat pada matriks bukanlah merupakan nilai yang digunakan ke fungsi tujuan, melainkan hanya probabilitas dipilihnya lokasi, nilai terbesar akan menjadi lokasi pertama yang akan dikunjungi, semakin kecil nilai semakin akhir dikunjungi. Nilai fungsi tujuan adalah total jarak dari keseluruhan perjalanan.

Jarak yang ditempuh dari lokasi satu ke lokasi lainnya yang dapat dilihat pada tabel 3.3.

Tabel 3. 3 Jarak dalam km (kilometer)

Lokasi	UNIKOM	Bandung Indah Plaza	Cihampelas Walk	Museum Geologi
UNIKOM	0	2.4	2.5	2.3

Bandung Indah Plaza	2.4	0	3.8	3
Cihampelas Walk	3.2	2.6	0	3.5
Museum Geologi	2.4	2.4	5.6	0

Cari nilai solusi awal yang lokasinya mempunyai nilai maksimal, lalu lokasi dengan nilai maksimal menjadi lokasi pertama yang dikunjungi. Berdasarkan tabel 3.1, kucing pertama memiliki solusi awal seperti pada tabel 3.4.

Tabel 3. 4 Solusi awal kucing pertama

Lokasi 1	Lokasi 2	Lokasi 3
0.3	0.1	0.5

Karena Lokasi 3 mempunyai nilai 0.5 yang lebih besar dari nilai Lokasi 1 dan Lokasi 2 maka rute yang dipilih adalah lokasi 3 → lokasi 1 → lokasi 2, dengan nilai fungsi tujuan $2.3 + 2.4 + 3.8 = 8.5$

Nilai fungsi setiap kucing dapat dilihat pada tabel 3.5.

Tabel 3. 5 Nilai fungsi Tujuan

Kucing ke-i	Nilai fungsi tujuan
1	8.5
2	8.1
3	9.7
4	8.5

5	9.7
6	10.5
7	8.1
8	8.5
9	11
10	11

Langkah 4

Hitung nilai fitness dari setiap kucing dengan cara membagi 1 dengan nilai fungsi tujuan yang didapat, dikarenakan total jarak berbanding terbalik dengan nilai fitness. Simpan posisi kucing dengan nilai fitness terbaik ke dalam memori. Berikan nilai SPC = 1 kepada kucing dengan nilai fitness terbaik, sisanya SPC = 0.

Nilai fitness dari kucing ke-1 dihitung dengan menggunakan rumus $1 / \text{nilai fungsi tujuan}$. Menurut tabel 3.5 nilai fungsi tujuan kucing ke-1 mempunyai nilai 8.5, maka nilai fitnessnya adalah $1 / 8.5 = 0.11765$.

Setelah menghitung semua nilai fitness setiap kucing, lalu berikan nilai SPC kepada kucing dengan nilai fitness terbaik.

Nilai fitness setiap kucing dapat dilihat pada tabel 3.6.

Sedangkan kucing dengan posisi terbaik dapat dilihat pada tabel 3.7.

Tabel 3. 6 Nilai fitness dan SPC

Kucing ke-i	Nilai fitness	SPC
1	0.11765	0

2	0.12346	1
3	0.10309	0
4	0.11765	0
5	0.10309	0
6	0.09524	0
7	0.12346	1
8	0.11765	0
9	0.09091	0
10	0.09091	0

Tabel 3. 7 Kucing terbaik sementara

Lokasi 1	Lokasi 2	Lokasi 3	Nilai Fitness
0.4	0.5	0.2	0.12346

Posisi kucing yang memiliki nilai fitness terbaik yang berada di memori akan terganti apabila terdapat nilai fitness dari posisi kucing lain yang lebih baik.

Langkah 5

Tentukan status kucing dengan melihat nilai MR. Jumlah kucing yang akan masuk ke proses tracing mode adalah total kucing $m \times MR$, sisanya masuk proses seeking mode. Berdasarkan parameter yang telah di inisialisasi di awal maka jumlah kucing yang masuk ke dalam tracing mode adalah $10 \times 20\% = 2$. Kucing yang masuk ke dalam tracing mode adalah kucing ke-1 dan ke-2. Lalu jumlah

kucing yang masuk proses seeking mode adalah $10 - 2 = 8$. Kucing yang masuk ke dalam seeking mode adalah kucing ke-3 sampai kucing ke-10.

Langkah 6

Pada tahap ini kucing akan diproses dengan statusnya masing-masing, kucing ke-1 dan ke-2 akan diproses sebagai tracing mode maka proses yang dilakukan adalah sebagai berikut

1. Update kecepatan setiap kucing oleh persamaan $v(k,d) = v(k,d) + r * c1 * (x(\text{best},d) - x(k,d))$

Dimana $x(\text{best}, d)$ adalah posisi kucing yang memiliki nilai fitness terbaik, $x(k, d)$ adalah posisi kucing (k) diambil contoh kucing ke-1, $c1 = 2$ dan r adalah nilai acak dalam kisaran $[0,1]$.

Berdasarkan pada tabel 3.6. kucing ke-2 yang mempunyai nilai fitness terbaik, maka kecepatan baru untuk kucing ke-1 adalah

$$V(k,d) = [0.2 \quad 0.1 \quad 0.7] + 0.2 * 2 * ([0.4 \quad 0.5 \quad 0.2] - [0.3 \quad 0.1 \quad 0.5])$$

$$V(k,d) = [0.2 \quad 0.1 \quad 0.7] + 0.2 * 2 * [0.1 \quad 0.4 \quad -0.3]$$

$$V(k,d) = [0.2 \quad 0.1 \quad 0.7] + 0.4 * [0.1 \quad 0.4 \quad -0.3]$$

$$V(k,d) = [0.2 \quad 0.1 \quad 0.7] + [0.04 \quad 0.16 \quad -0.12]$$

$$V(k,d) = [0.24 \quad 0.16 \quad 0.58]$$

Kecepatan setiap kucing yang masuk ke dalam proses tracing mode dapat dilihat pada tabel 3.8.

Tabel 3. 8 Kecepatan awal dan kecepatan baru

Kucing ke-i	Kecepatan awal	Kecepatan baru
1	[0.2 0.1 0.7]	[0.24 0.16 0.58]

2	[0.8 0.8 0.3]	[0.8 0.8 0.3]
---	---------------	---------------

2. Update posisi terbaru dengan persamaan $x(k,d) = x(k,d) + v(k,d)$

Posisi kucing ke-1 menjadi

$$x(k,d) = [0.3 \ 0.1 \ 0.5] + [0.24 \ 0.16 \ 0.58]$$

$$x(k,d) = [0.54 \ 0.26 \ 1.08]$$

Posisi awal dan posisi baru kucing yang masuk ke proses tracing mode dapat dilihat pada tabel 3.9.

Tabel 3. 9 Posisi awal kucing dan posisi baru kucing

Kucing ke-i	Posisi awal	Posisi baru
1	[0.3 0.1 0.58]	[0.54 0.26 1]
2	[0.4 0.5 0.2]	[1.2 1.3 0.5]

Sedangkan proses seeking mode adalah sebagai berikut

Kucing yang akan diproses pada seeking mode adalah kucing ke-3 sampai kucing ke-10.

1. Bangkitkan kucing tiruan sebanyak $j = \text{SMP}$, apabila SPC bernilai 1 maka $j = \text{SMP} - 1$.

Kucing ke 3 berdasarkan tabel 3.1 masuk ke proses seeking mode dan nilai SPC nya berdasarkan tabel 3.6.

maka tiruannya = $\text{SMP} = 5$.

2. Untuk setiap tiruan, berdasarkan $\text{CDC} = 80\%$ yang akan dipakai untuk menentukan berapa banyak kucing tiruan yang posisinya berubah, maka $5 \times 80\% = 4$, maka 4 dari 5 kucing tiruan pada kucing ke-3 yang akan dirubah posisinya. Posisi yang akan dirubah adalah kucing tiruan pertama

sampai kucing tiruan ke 4. Tambahkan atau kurangkan secara acak SRD = 20% dari nilai saat ini dan gantikan nilai sebelumnya.

Posisi asal kucing ke-3 adalah [0.9 0.4 0.1]

Maka nilai yang akan ditambahkan atau kurangkan adalah

$20\% \times 0.9 = 0.18$ untuk posisi lokasi 1

$20\% \times 0.4 = 0.08$ untuk posisi lokasi 2

$20\% \times 0.1 = 0.02$ untuk posisi lokasi 3

Setelah melalui proses acak antara tambahkan dan kurangkan, didapat posisi untuk kucing tiruan pertama pada kucing ke-3 adalah [0.72 0.48 0.12]. Lalu hitung nilai fungsi tujuan dari masing – masing kucing tiruan dengan menggunakan cara yang sama pada langkah ke 3. Kucing tiruan pertama dari kucing ke-3 mempunyai nilai tujuan 9.7.

3. Hitung nilai fitness dari semua kucing tiruan dengan cara yang sama pada langkah ke 4.

Fitness = 1 / fungsi tujuan.

Nilai fitness kucing tiruan pertama dari kucing ke 3 adalah

Fitness = 1 / 9.7

Fitness = 0.10309

Nilai fitness dari setiap kucing tiruan dapat dilihat pada tabel 3.10.

Tabel 3. 10 Kucing tiruan beserta fungsi tujuan dan fitnessnya

Kucing ke-i	Tiruan kucing ke-j	Posisi awal	Posisi baru	Fungsi tujuan	fitness
3	1	[0.9 0.4 0.1]	[0.72 0.48 0.12]	9.7	0.10309
	2	[0.9 0.4 0.1]	[0.72 0.32 0.08]	9.7	0.10309
	3	[0.9 0.4 0.1]	[1 0.48 0.08]	9.7	0.10309
	4	[0.9 0.4 0.1]	[1 0.32 0.12]	9.7	0.10309

Kucing ke-i	Tiruan kucing ke-j	Posisi awal	Posisi baru	Fungsi tujuan	fitness
	5	[0.9 0.4 0.1]	[0.9 0.4 0.1]	9.7	0.10309
4	1	[0.6 0.2 0.7]	[0.72 0.24 0.56]	11	0.09091
	2	[0.6 0.2 0.7]	[0.48 0.24 0.84]	8.5	0.11765
	3	[0.6 0.2 0.7]	[0.72 0.16 0.56]	11	0.09091
	4	[0.6 0.2 0.7]	[0.48 0.24 0.56]	8.5	0.11765
	5	[0.6 0.2 0.7]	[0.6 0.2 0.7]	8.5	0.11765
5	1	[0.2 0.1 0.1]	[0.24 0.08 0.12]	11	0.09091
	2	[0.2 0.1 0.1]	[0.16 0.12 0.08]	9.7	0.10309
	3	[0.2 0.1 0.1]	[0.24 0.12 0.12]	9.7	0.10309
	4	[0.2 0.1 0.1]	[0.16 0.08 0.12]	11	0.09091
	5	[0.2 0.1 0.1]	[0.2 0.1 0.1]	9.7	0.09091
6	1	[0.2 0.3 0.9]	[0.24 0.36 0.72]	10.5	0.09524
	2	[0.2 0.3 0.9]	[0.24 0.24 1]	10.5	0.09524
	3	[0.2 0.3 0.9]	[0.24 0.24 0.72]	10.5	0.09524
	4	[0.2 0.3 0.9]	[0.16 0.36 0.72]	10.5	0.09524
	5	[0.2 0.3 0.9]	[0.2 0.3 0.9]	10.5	0.09524
7	1	[0.7 0.9 0.5]	[0.84 1 0.6]	8.1	0.12346
	2	[0.7 0.9 0.5]	[0.84 0.72 0.4]	9.7	0.10309
	3	[0.7 0.9 0.5]	[0.56 0.72 0.6]	8.4	0.11905

Kucing ke-i	Tiruan kucing ke-j	Posisi awal	Posisi baru	Fungsi tujuan	fitness
	4	[0.7 0.9 0.5]	[0.56 0.72 0.4]	8.1	0.12346
8	1	[0.5 0.2 0.6]	[0.6 0.24 0.48]	11	0.09091
	2	[0.5 0.2 0.6]	[0.4 0.16 0.72]	8.5	0.11765
	3	[0.5 0.2 0.6]	[0.6 0.16 0.48]	11	0.09091
	4	[0.5 0.2 0.6]	[0.4 0.24 0.72]	8.5	0.11765
	5	[0.5 0.2 0.6]	[0.5 0.2 0.6]	8.5	0.11765
9	1	[0.7 0.3 0.6]	[0.84 0.24 0.72]	11	0.09091
	2	[0.7 0.3 0.6]	[0.56 0.36 0.72]	8.5	0.11765
	3	[0.7 0.3 0.6]	[0.56 0.36 0.48]	11	0.09091
	4	[0.7 0.3 0.6]	[0.84 0.24 0.48]	11	0.09091
	5	[0.7 0.3 0.6]	[0.7 0.3 0.6]	11	0.09091
10	1	[0.4 0.1 0.2]	[0.32 0.12 0.24]	11	0.09091
	2	[0.4 0.1 0.2]	[0.32 0.12 0.16]	11	0.09091
	3	[0.4 0.1 0.2]	[0.48 0.08 0.24]	11	0.09091
	4	[0.4 0.1 0.2]	[0.48 0.08 0.16]	11	0.09091
	5	[0.4 0.1 0.2]	[0.4 0.1 0.2]	11	0.09091

4. Jika nilai semua fitness tidak sama hitung probabilitas terpilihnya tiruan dengan menggunakan persamaan $P(i) = (|FS(i) - FS(b)|) / (FS(\max) - FS(\min))$, karena setiap kucing tiruan pada kucing ke-3 mempunyai nilai

fitness yang sama maka semua mempunyai kemungkinan dipilih yang sama. Berbeda dengan setiap kucing tiruan dari kucing ke-4 yang mempunyai nilai fitness berbeda, maka dihitung nilai probabilitas terpilihnya. Nilai $FS(b) = FS(\min)$.

Contoh yang digunakan adalah kucing ke-4 tiruan ke-1, berdasarkan tabel 3.10. maka probabilitas dipilihnya adalah

$$P(i) = (0.09091 - 0.09091) / (0.11765 - 0.09091)$$

$$P(i) = 0 / 0.02674$$

$$P(i) = 0$$

Nilai probabilitas dari setiap kucing tiruan dapat dilihat pada tabel 3.11.

Tabel 3. 11 kucing tiruan dengan nilai probabilitas dipilihnya sebagai posisi baru

Kucing ke-i	Tiruan kucing ke-j	Posisi baru	Fungsi tujuan	fitness	Probabilitas
3	1	[0.72 0.48 0.12]	9.7	0.10309	1
	2	[0.72 0.32 0.08]	9.7	0.10309	1
	3	[1 0.48 0.08]	9.7	0.10309	1
	4	[1 0.32 0.12]	9.7	0.10309	1
	5	[0.9 0.4 0.1]	9.7	0.10309	1
4	1	[0.72 0.24 0.56]	11	0.09091	0
	2	[0.48 0.24 0.84]	8.5	0.11765	1
	3	[0.72 0.16 0.56]	11	0.09091	0
	4	[0.48 0.24 0.56]	8.5	0.11765	1
	5	[0.6 0.2 0.7]	8.5	0.11765	1

Kucing ke-i	Tiruan kucing ke-j	Posisi baru	Fungsi tujuan	fitness	Probabilitas
5	1	[0.24 0.08 0.12]	11	0.09091	0
	2	[0.16 0.12 0.08]	9.7	0.10309	1
	3	[0.24 0.12 0.12]	9.7	0.10309	1
	4	[0.16 0.08 0.12]	11	0.09091	0
	5	[0.2 0.1 0.1]	9.7	0.09091	0
6	1	[0.24 0.36 0.72]	10.5	0.09524	1
	2	[0.24 0.24 1]	10.5	0.09524	1
	3	[0.24 0.24 0.72]	10.5	0.09524	1
	4	[0.16 0.36 0.72]	10.5	0.09524	1
	5	[0.2 0.3 0.9]	10.5	0.09524	1
7	1	[0.84 1 0.6]	8.1	0.12346	1
	2	[0.84 0.72 0.4]	9.7	0.10309	0
	3	[0.56 0.72 0.6]	8.4	0.11905	0
	4	[0.56 0.72 0.4]	8.1	0.12346	1
8	1	[0.6 0.24 0.48]	11	0.09091	0
	2	[0.4 0.16 0.72]	8.5	0.11765	1
	3	[0.6 0.16 0.48]	11	0.09091	0
	4	[0.4 0.24 0.72]	8.5	0.11765	1
	5	[0.5 0.2 0.6]	8.5	0.11765	1

Kucing ke-i	Tiruan kucing ke-j	Posisi baru	Fungsi tujuan	fitness	Probabilitas
9	1	[0.84 0.24 0.72]	11	0.09091	0
	2	[0.56 0.36 0.72]	8.5	0.11765	1
	3	[0.56 0.36 0.48]	11	0.09091	0
	4	[0.84 0.24 0.48]	11	0.09091	0
	5	[0.7 0.3 0.6]	11	0.09091	0
10	1	[0.32 0.12 0.24]	11	0.09091	1
	2	[0.32 0.12 0.16]	11	0.09091	1
	3	[0.48 0.08 0.24]	11	0.09091	1
	4	[0.48 0.08 0.16]	11	0.09091	1
	5	[0.4 0.1 0.2]	11	0.09091	1

5. Secara acak pilih kandidat sebagai posisi baru kucing yang mempunyai probabilitas 1.

Setelah proses pemilihan acak, maka dapat dilihat nilai posisi baru kucing pada tabel 3.12.

Tabel 3. 12 kucing dalam seeking mode dengan posisi baru dan fitnessnya

Kucing ke-i	Posisi baru	Fitness
3	[1 0.48 0.08]	0.10309
4	[0.48 0.24 0.56]	0.11765
5	[0.24 0.12 0.12]	0.10309
6	[0.2 0.3 0.9]	0.09524

7	[0.84 1 0.6]	0.12346
8	[0.5 0.2 0.6]	0.11765
9	[0.56 0.36 0.72]	0.11765
10	[0.32 0.12 0.16]	0.09091

Dari proses *tracing mode* dan *seeking mode*, ditemukan posisi kucing baru.

Posisi kucing terbaik sementara dapat dilihat pada tabel 3.13.

Tabel 3. 13 posisi kucing terbaik

Lokasi 1	Lokasi 2	Lokasi 3
0.4	0.5	0.2

Langkah 7

Apabila iterasi terpenuhi hentikan program, jika tidak ulangi langkah 4 sampai 6.

Lankah 8

Setelah ditemukan satu set solusi terbaik sistem akan menampilkan rute perjalanan terdekat dengan menggunakan angkot apa saja. Output berupa rute Lokasi awal → Lokasi 2 → Lokasi 1 → Lokasi 3

UNIKOM → Cihampelas Walk → Bandung Indah Plaza → Museum Geologi

Menggunakan angkot : Naik angkot Cicaheum - Ciroyom turun di Cihampelas Walk → naik angkot Kalapa - Ledeng turun di BIP → naik angkot St. Hall - Sadang Serang turun di Jl. WR. Supratman, dilanjutkan angkot Cicaheum - Ledeng turun di Museum Geologi.

3.2.3 Analisis Non Fungsional

Analisis kebutuhan non fungsional adalah sebuah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan digunakan untuk perangkat lunak yang dibangun.

3.2.3.1 Analisis Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak merupakan faktor-faktor yang harus dipenuhi untuk merancang sebuah perangkat lunak sehingga perangkat lunak tersebut sesuai dengan maksud dan tujuan perangkat lunak tersebut dibuat.

Kebutuhan Perangkat Lunak

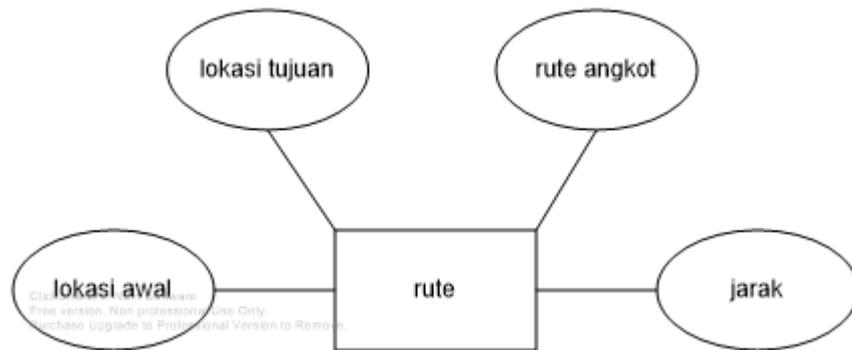
1. Sistem Operasi.
2. Web Server.
3. Text Editor.
4. Browser yang mendukung HTML 5 dan CSS3.

3.2.4 Analisis Kebutuhan Fungsional

Analisis Kebutuhan Fungsional adalah tahap analisis sistem, alat bantu yang digunakan untuk mempermudah pembuatan sistem. Pemodelan sistem pada penelitian ini menggunakan jenis pemodelan *Unified Modelling Language* (UML).

3.2.4.1 Perancangan Basis Data

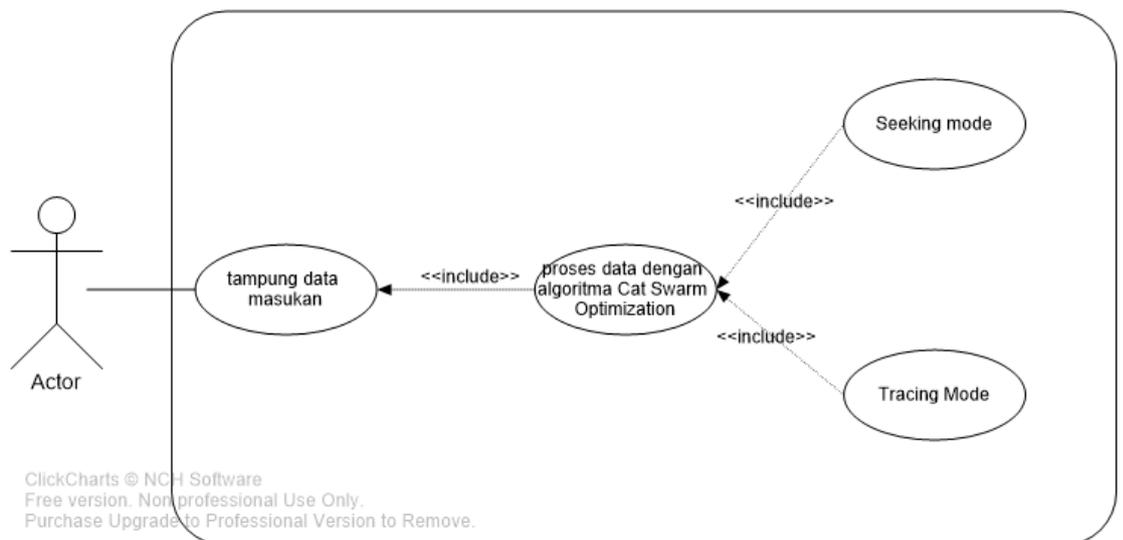
Agar data lebih mudah dipahami, data diklasifikasikan dengan basis data yang akan direpresentasikan dengan *Entity Relationship Diagram* (ERD).



Gambar 3. 3 Entity Relationship Diagram

3.2.4.2 Use Case Diagram

Use case diagram adalah sebuah diagram yang menggambarkan interaksi antara sistem dan pengguna. Adapun gambar use case diagram dapat dilihat pada gambar



Gambar 3. 4 Use Case Diagram

3.2.4.3 Use Case Scenario

Use Case Scenario adalah alur jalannya proses use case dari sisi aktor dan sistem. Berikut adalah *Use Case Scenario* dari *Use Case* yang ada.

Tabel 3. 14 Use Case Scenario Menentukan Jumlah Lokasi tujuan

Use case	Tampung data masukan
Actor	Pengguna
PreCondition	Pengguna berada pada tampilan utama
PostCondition	Menampilkan combo box pilihan lokasi
Main flow of Event	
Actor	System Response
1. Memasukkan banyaknya lokasi tujuan	2. Menampilkan combo box pilihan lokasi sesuai banyaknya lokasi tujuan
3. Memasukkan lokasi awal dan lokasi-lokasi tujuan	4. Menampung lokasi awal dan lokasi-lokasi tujuan

Tabel 3. 15 Use Case Scenario Proses data masukan dengan algoritma CSO

Use case	Proses data masukan dengan algoritma <i>Cat Swarm Optimization</i>
Actor	Pengguna
PreCondition	Memasukkan lokasi awal dan lokasi-lokasi tujuan
PostCondition	Posisi kucing akan digunakan untuk proses seeking mode dan tracing mode
Main flow of Event	
Actor	System Response

1. Menekan tombol cari jalur	2. Membuat matriks sebanyak m (banyaknya kucing) $\times n$ (banyaknya lokasi tujuan).
	3. Menghitung nilai fungsi tujuan dan fitness dari setiap kucing
	4. Memberi status untuk setiap kucing

Tabel 3. 16 Use Case Tracing Mode

Use case	Tracing Mode
Actor	Pengguna
PreCondition	Memilih lokasi awal dan lokasi lokasi tujuan
PostCondition	Posisi kucing diperbarui
Main flow of Event	
Actor	System Response
1. Menjalankan program pencarian jalur terbaik	2. Update kecepatan setiap kucing
	3. Update posisi setiap kucing
	4. Hitung nilai fungsi tujuan dan fitness

Tabel 3. 17 Use Case Seeking Mode

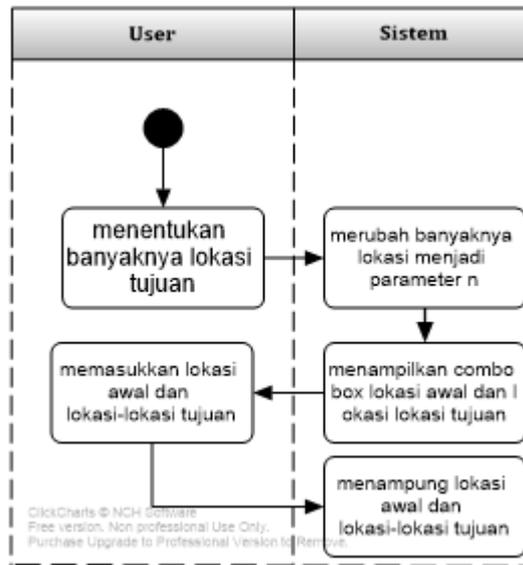
Use case	<i>Seeking Mode</i>
Actor	Pengguna
PreCondition	Memilih lokasi awal dan lokasi lokasi tujuan
PostCondition	Posisi kucing diperbarui
Main flow of Event	
Actor	System Response
1. Menjalankan program pencarian jalur terbaik	2. Bangkitkan kucing tiruan sebanyak nilai SMP
	3. Update posisi kucing tiruan
	4. Hitung nilai fungsi tujuan dan fitness kucing tiruan
	5. Gantikan posisi kucing dengan posisi kucing terpilih

3.2.4.4 Activity Diagram

Activity diagram digunakan untuk menggambarkan proses alur sistem yang bekerja. Activity diagram menunjukkan suatu alur kegiatan secara berurutan.

A. Activity Tampung Data Masukan

Berikut ini merupakan gambaran alur aktifitas saat tampung data masukan



Gambar 3. 5 Activity Diagram tampung data masukan

B. Activity Proses Data Masukan dengan Algoritma CSO

Berikut ini merupakan gambaran alur aktifitas saat program melakukan proses data masukan dengan algoritma CSO.



Gambar 3. 6 *Activity Diagram* proses data masukan dengan algoritma CSO

C. Activity Tracing Mode

Berikut ini merupakan gambaran alur aktifitas untuk proses Tracing Mode



Gambar 3.7 Activity Diagram Tracing Mode

D. Activity Seeking Mode

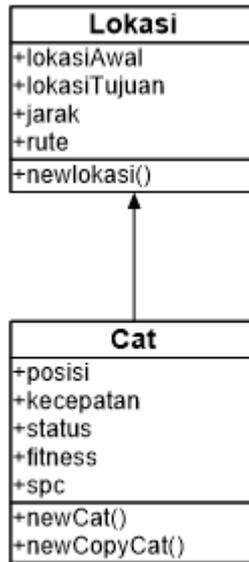
Berikut ini merupakan gambaran alur aktifitas untuk proses Seeking Mode



Gambar 3. 8 Activity Diagram Seeking Mode

3.2.4.5 Class Diagram

Class diagram menunjukkan bagaimana entitas yang berbeda berhubungan satu sama lain. Adapun *class diagram* yang digunakan pada penelitian ini dapat dilihat pada gambar 3.8.



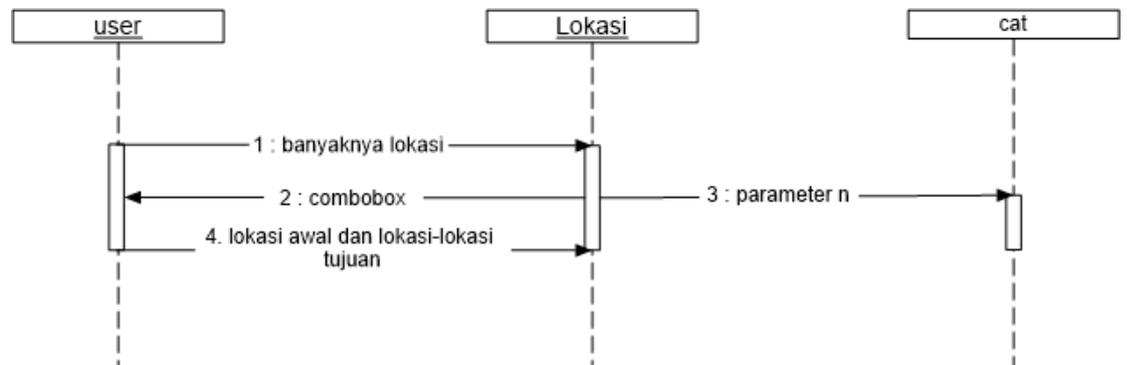
Gambar 3. 9 Class Diagram

3.2.4.6 Sequence Diagram

Sequence diagram merupakan diagram aliran pesan, yang menggambarkan pesan yang mengalir dari objek ke objek lainnya. Berikut adalah diagram aliran pesan pada penelitian ini.

A. Sequence Diagram Tampung Data Masukan

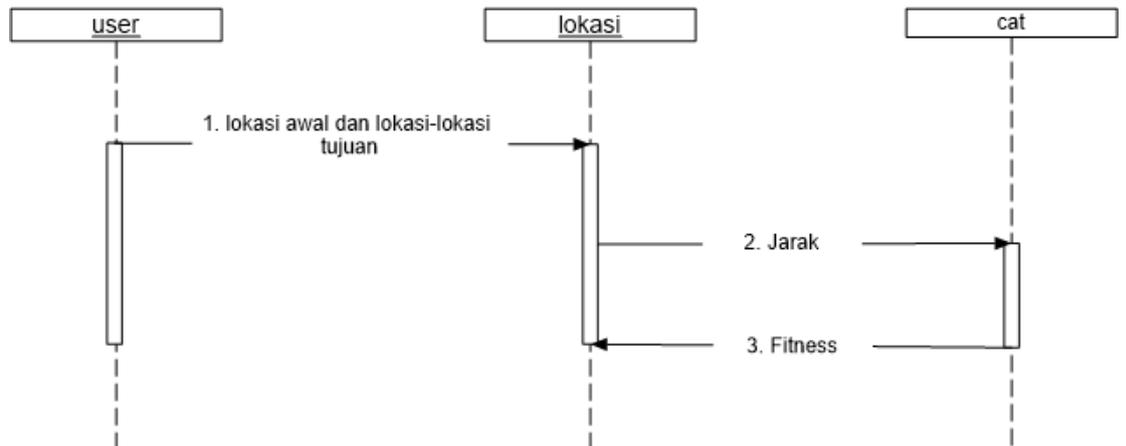
Berikut ini adalah sequence diagram untuk proses tampung data masukan



Gambar 3. 10 Sequence Diagram tampung data masukan

B. Sequence Diagram Proses data dengan algoritma CSO

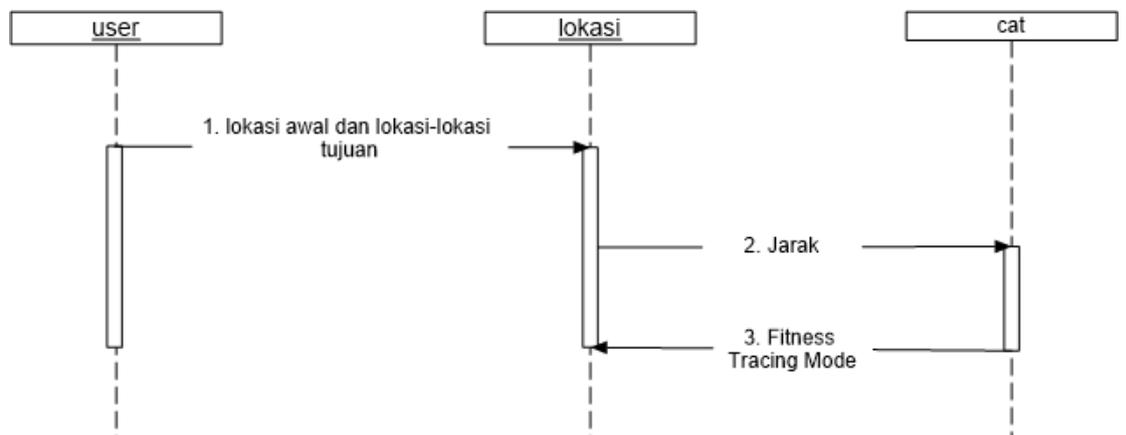
Berikut ini adalah sequence diagram untuk proses data dengan algoritma CSO.



Gambar 3. 11 *Sequence Diagram* Proses data dengan algoritma CSO

C. Sequence Diagram Tracing Mode

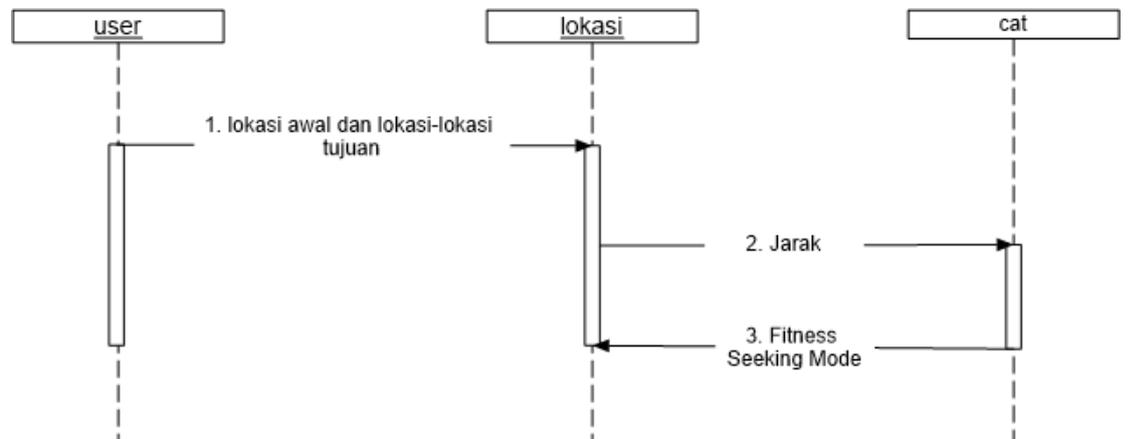
Berikut ini adalah sequence diagram untuk proses *Tracing Mode*.



Gambar 3. 12 *Sequence Diagram* Tracing Mode

D. Sequence Diagram Seeking Mode

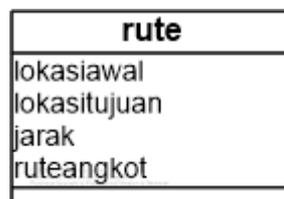
Berikut ini adalah sequence diagram untuk proses *Seeking Mode*.



Gambar 3. 13 Sequence Diagram Seeking Mode

3.2.5 Perancangan Basis Data

Perancangan basis data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Pada penelitian ini hanya menggunakan 1 tabel pada basis datanya, yaitu tabel rute yang dapat dilihat pada gambar 3.14.



Gambar 3. 14 Skema Relasi

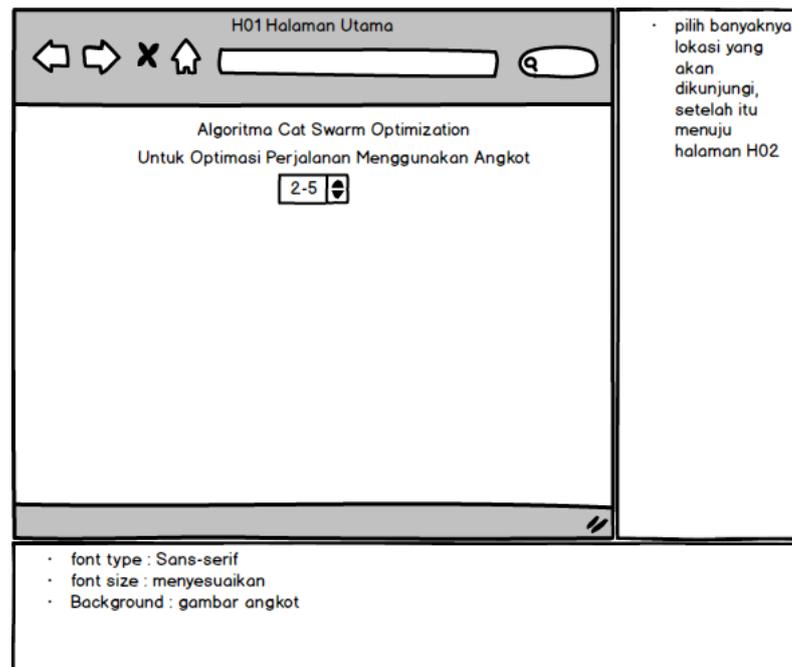
3.2.5.1 Struktur Tabel

Tabel 3. 18 Struktur Tabel rute

Nama <i>Field</i>	Tipe Data	Ukuran	Tipe Key
lokasiawal	varchar	50	
lokasitujuan	varchar	50	
jarak	float		
ruteangkot	varchar	300	

3.2.6 Perancangan Antar Muka

Pada sub bab ini akan diterangkan mengenai rancangan yang digunakan oleh penulis dalam merancang program yang akan dibuat. Perancangan program ini terdiri dari pembuatan sistem menu, desain input, dan desain output.



Gambar 3. 15 Mock up halaman utama

The image shows a web application interface for 'Algoritma Cat Swarm Optimization'. The interface is divided into several sections:

- Header:** A grey bar with the text 'H02' and navigation icons (back, forward, close, home) and a search input field.
- Title:** 'Algoritma Cat Swarm Optimization' and 'Untuk Optimasi Perjalanan Menggunakan Angkot'.
- Input:** A numeric input field containing '2-5' with up and down arrows.
- Dropdowns:** Four dropdown menus labeled 'Lokasi Awal', 'Lokasi tujuan 1', 'Lokasi tujuan 2', and 'Lokasi tujuan 3'.
- Action:** A button labeled 'Cari Jalur'.
- Footer:** A grey bar with a double-slash icon.

On the right side, there is a list of instructions:

- pilih lokasi awal dan lokasi lokasi tujuan
- klik tombol cari jalur untuk ke halaman H03

At the bottom of the interface, there are design specifications:

- font type : Sans-serif
- font size : menyesuaikan
- Background : gambar angkot

Gambar 3. 16 Halaman Pilih lokasi awal dan lokasi tujuan

H03				Menampilkan hasil keluaran berupa rute perjalanan menampilkan hasil perhitungan Cat Swarm Optimization
 <input style="width: 150px;" type="text"/> <input type="button" value="Q"/>				
Algoritma Cat Swarm Optimization Untuk Optimasi Perjalanan Menggunakan Angkot <input style="width: 50px; text-align: center; border: 1px solid black;" type="text" value="2-5"/>				
Lokasi Awal ▾	Lokasi tujuan 1 ▾	Lokasi tujuan 2 ▾	Lokasi tujuan 3 ▾	
<input type="button" value="Cari Jalur"/>				
Perjalanan ▲ Lokasi Awal -> Lokasi Tujuan		Menggunakan Angkot Naik angkot a turun di x		
Hasil Perhitungan				
				
<ul style="list-style-type: none"> • font type : Sans-serif • font size : menyesuaikan • Background : gambar angkot 				

Gambar 3. 17 Halaman Hasil perhitungan dan rute perjalanan

