BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Pada tahap implementasi merupakan hasil penerapan algoritma semut, menjelaskan analisa data yang diperoleh dari hasil percobaan dan efektifitas hasil yang telah dilakuan dari tahap perancangan yang diterjemahkan ke dalam bahasa pemrograman. Wujud dari hasil tahap implementasi ini nantinya adalah sebuah perangkat lunak yang siap untuk diuji dan dijalankan. Dimana akan terlihat kekurangan-kekurangan pada perangkat lunak untuk selanjutnya diadakan pengembangan sistem.

Perangkat lunak pencarian jarak terdekat di kota Tasikmalaya ini dilakukan dalam lingkup perangkat keras, perangkat lunak dan jaringan internet. Spesifikasi perangkat keras yang digunakan adalah:

Tabel 4-1 Spesifikasi Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Proccessor*	Intel Core i5 2.00 GHz
2	RAM	4 Gb
3	Hardisk	1 Tb
4	Layar Monitor	Resolusi 1920 x 1090 pixel.
6	Sound Card	Realtek High Definition Audio
7	Modem	ZTE MF626

Sedangkan untuk spesifikasi perangkat lunak yang digunakan untuk implementasi perangkat lunak adalah:

Tabel 4-2 Spesifikasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 7
2	Bahasa Pemrograman (pada Android)	Java
3	Bahasa Pemrograman (pada *Web Service*)	Php

4.1.1 Implementasi Data

Tabel 4-3 Implementasi Data

No.	Nama Tabel	Nama File
1	user	tabel user.sql
2	titik	tabel titik.sql

Tabel 4-4 *Query User*

```
1  CREATE TABLE `user` (
2            `username` varchar(50) NOT NULL,
3            `password` varchar(50) NOT NULL,
4            `nama` varchar(100) NOT NULL,
5            PRIMARY KEY (`username`)
6          )ENGINE=InnoDB DEFAULT CHARSET=latin1
```

Tabel 4-5 *Query* Titik

```
1  CREATE TABLE `update_titik` (
2                  `id` int(11) NOT NULL AUTO_INCREMENT,
3                  `tgl_update` datetime NOT NULL,
4                 PRIMARY KEY (`id`)
5               )ENGINE=InnoDB     AUTO_INCREMENT=35     DEFAULT
   CHARSET=latin1
```

4.1.2 Implementasi Antarmuka

Implementasi antarmuka menggambarkan tampilan dari perangkat lunak yang dibangun, yaitu implementasi pencarian jarak terdekat di kota Tasikmalaya.

4.1.1.1 Implementasi Antarmuka pada Perangkat Lunak Android

Implementasi penggambaran pada perangkat lunak Android dapat dilihat pada tabel berikut:

Tabel 4-6 Implementasi Antarmuka pada Perangkat Lunak Android

Menu	Deskripsi	Nama File
Form Deskripsi Titik	Tampilan deskripsi lokasi.	ActivityDeskripsi.java
Form Peta Hasil Pencarian	Tampilan peta hasil pencarian jarak terdekat.	ActivityMapResult.java
Form Rute Optimasi	Tampilan penentuan tempat asal dan tempat tujuan.	ActivityOptimasi.java
Form Hasil Pencarian	Tampilan hasil pencarian jarak terdekat.	ActivityResult.java
Form Lihat Peta	Tampilan untuk melihat peta.	ActivityRute.java
Form Utama	Tampilan pada saat perangkat lunak dibuka.	ActivitySemut.java
Form Pengaturan Optimasi	Tampilan pada pengaturan.	ActivitySetting.java
Form Menu	Tampilan menu utama.	ActivityUtama.java
Class Peta	Untuk menampilkan peta.	MapView.java

Berikut adalah tampilan antarmuka pada perangkat lunak pencarian jarak terdekat di kota Tasikmalaya:

1. Form antarmuka halaman utama perangkat lunak pencarian jarak terdekat.



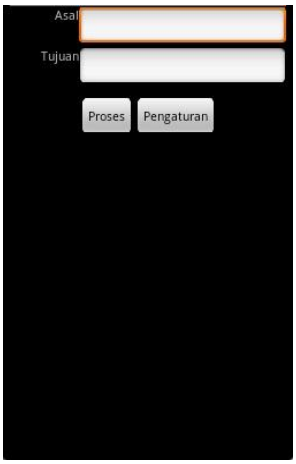
Gambar 4.1 Form Halaman Utama

2. Form antarmuka menu utama perangkat lunak pencarian jarak terdekat.



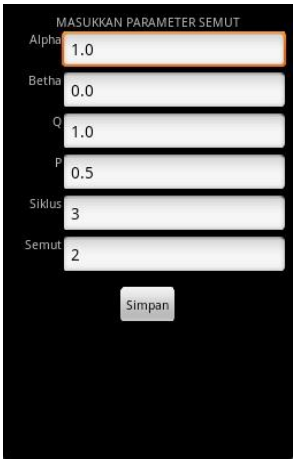
Gambar 4.2 Form Menu Utama

3. Form antarmuka menu rute optimasi perangkat lunak pencarian jarak terdekat.



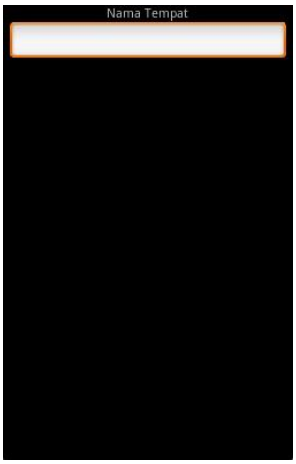
Gambar 4.3 Form Rute Optimasi

4. Form antarmuka menu pengaturan optimasi perangkat lunak pencarian jarak terdekat.



Gambar 4.4 Form Pengaturan Optimasi

5. Form antarmuka menu deskripsi titik perangkat lunak pencarian jarak terdekat.



Gambar 4.5 Form Deskripsi Titik

4.1.1.2 Implementasi Antarmuka pada *Web Service*

Implementasi penggambaran pada *Web Service* pencarian jarak terdekat dapat dilihat pada tabel berikut:

Tabel 4-7 Implementasi Antarmuka pada Perangkat *Web Service*

Menu	Deskripsi	Nama File
Form Tentang.	Tampilan deskripsi *web service*.	about.php
Form Daftar Titik	Tampilan untuk menghapus titik.	hapustitik.php
Form Utama	Tampilan pada saat *web service* dibuka.	index.php
Form Gambar Peta	Tampilan untuk melihat peta.	lihatpeta.php
Form Keluar	Tampilan untuk keluar dari *web service*.	logout.php
Form Tambah Titik	Tampilan untuk menambah titik.	manipulasititik.php
Form Masuk	Tampilan sebelum masuk.	menufront.php
Form Masuk	Tampilan sebelum	menulogin.php

	masuk.	
Form Lihat Peta	Tampilan untuk melihat peta.	peta.php
Form Daftar Titik	Tampilan setelah masuk.	titik.php

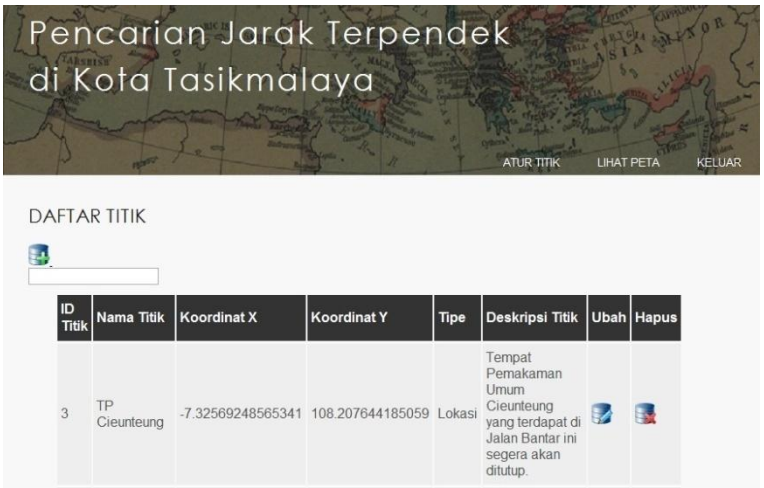
Berikut adalah tampilan antarmuka pada perangkat lunak pencarian jarak terdekat di kota Tasikmalaya:

1. Form antarmuka halaman utama *web service* pencarian jarak terdekat.



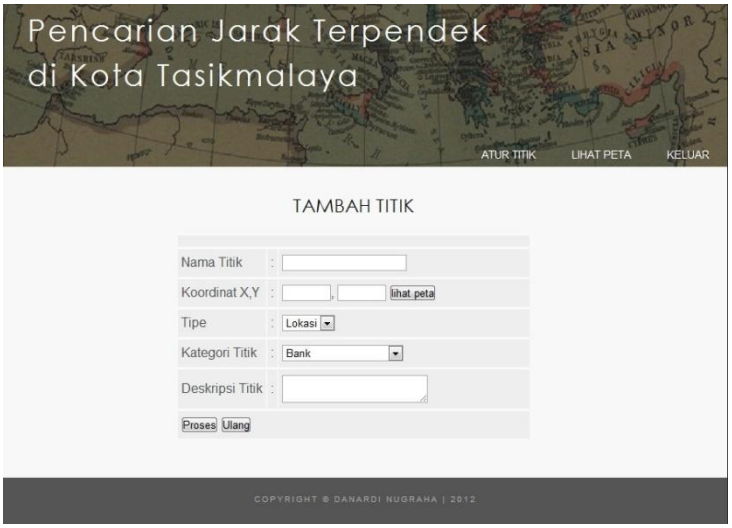
Gambar 4.6 Form Halaman Utama

2. Form antarmuka daftar titik *web server* pencarian jarak terdekat.



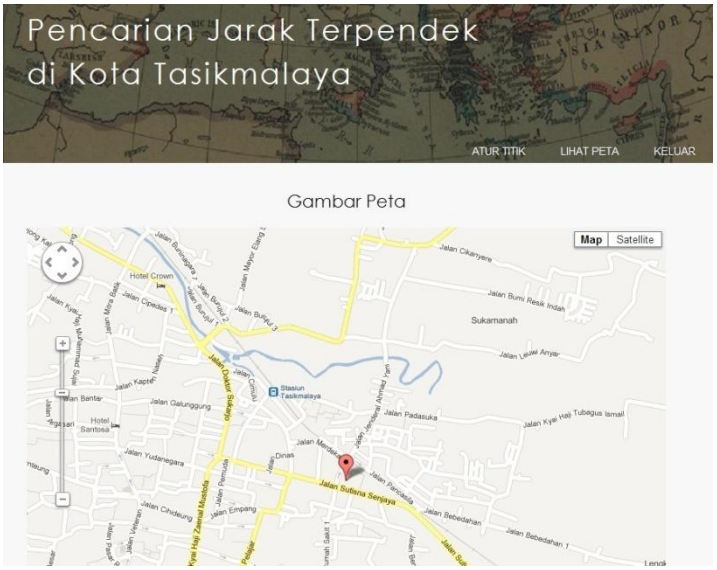
Gambar 4.7 Form Daftar Titik

3. Form antarmuka tambah titik *web service* pencarian jarak terdekat.



Gambar 4.8 Form Daftar Titik

4. Form antarmuka lihat peta *web service* pencarian jarak terdekat.



Gambar 4.9 Form Lihat Peta

4.2 Pengujian Sistem

4.2.1 Rencana Pengujian

Rencana pengujian yang akan dilakukan pada perangkat lunak pencarian jarak terdekat di kota Tasikmalaya ini dapat dilihat sebagai berikut:

Tabel 4-8 Rencana Pengujian pada Perangkat Lunak Android

No.	Proses	Jenis Pengujian
1	Optimasi Rute	*White Box*
2	Deskripsi Titik	*Black Box*
3	Pengaturan Parameter	*Black Box*

Tabel 4-9 Rencana Pengujian pada *Web Service*

No.	Proses	Jenis Pengujian
1	Login	*Black Box*
2	Tambah Titik	*Black Box*

4.2.2 Pengujian *White Box*

Pengujian *white box* bertujuan untuk mengetahui kinerja logika yang dibuat pada sebuah perangkat lunak apakah dapat berjalan dengan baik atau tidak. Pengujian ini akan digunakan pada pencarian jarak tedekat, untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis.

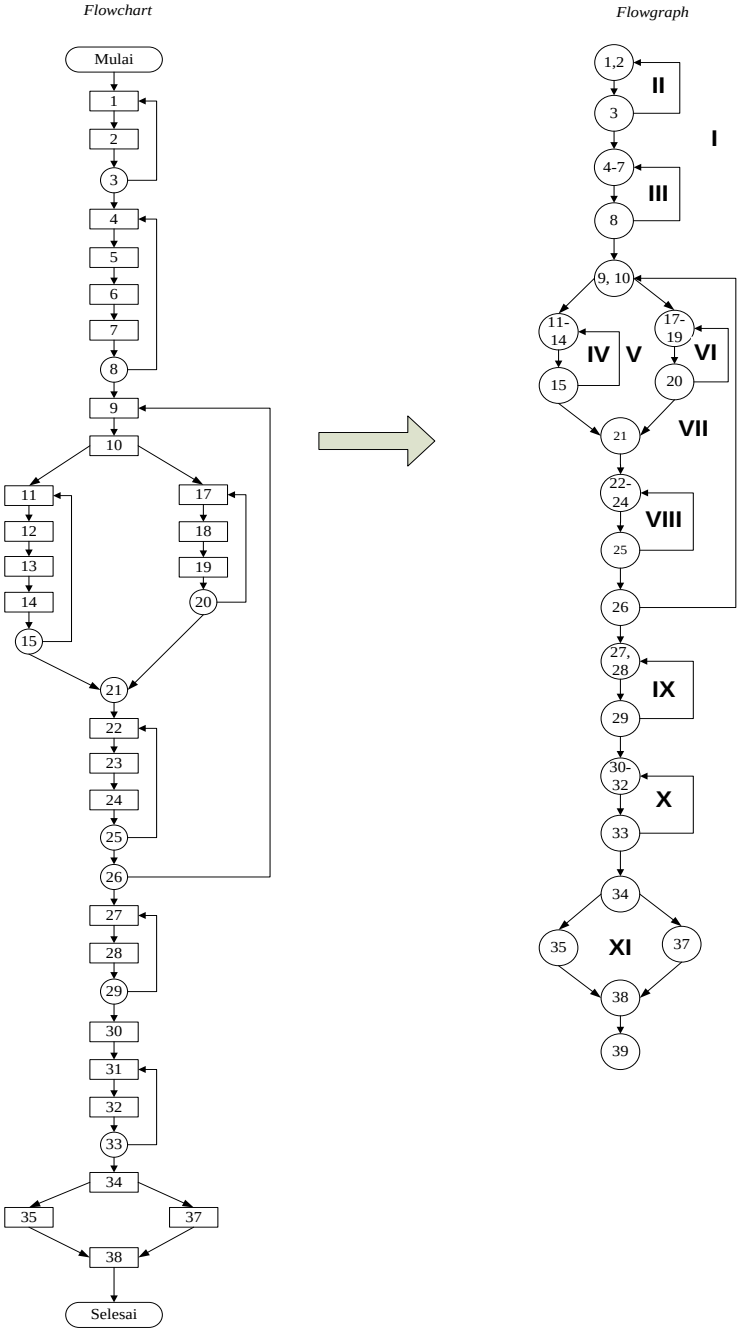
Pengujian *white box* dilakukan berdasarkan beberapa tahap yaitu:

1. Mengubah *pseudocode* menjadi *flowchart.*
2. Mengubah *flowchart* menjadi *flowgraph* ke dalam bentuk yang lebih sederhana.
3. Tahap pengujian dilakukan dengan lima cara yaitu, menghitung *region*, menghitung *Cyclomatic Complexity*, menghitung *Independent Path*, menggunakan *graph matriks*, menghitung *predicate node.*

4.2.2.1 Pengujian Optimasi Rute

Pengujian dilakukan berdasarkan *pseudocode* pada tabel 3.1.

1. Pengubahan *pseudocode* menjadi *flowchart* kemudian menjadi *flowgraph.*



Gambar 4.10 Pengubahan *Flowchart* menjadi *Flowgraph* Optimasi Rute

2. Tahap Pengujian.

   a. Region = 11

   b. $V_{(G)}$ = Edge – Node + 2

      = 31 – 22 + 2

      = 9 + 2

      = 11

   c. *Independent Path*

      1 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      2 = 1, 2, 3, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      3 = 1, 2, 3, 4, 5, 6, 7, 8, 4, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      4 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      5 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      6 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 17, 18, 19, 20, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

      7 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

8 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 22, 23, 24, 25, 26, 27, 28, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

9 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 39

10 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 30, 31, 32, 33, 34, 35, 38, 39

11 = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 37, 38, 39

d. *Predicate Node*

$V_{(G)}$ = Jumlah node yang memiliki lebih dari 1 jalur +1

= 10 + 1

= 11

e. Graph Matriks

Tabel 4-10 *Graph Matriks*

N	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	sum
1		1																					0
2	1		1																				1
3				1																			0
4			1		1																		1
5						1		1															1
6							1																0
7						1				1													1
8									1														0
9								1		1													1
10											1												0
11												1											0
12											1		1										1
13					1										1								1
14															1								0
15														1		1							1
16																	1						0
17																1		1					1
18																			1	1			1
19																					1		0
20																					1		0
21																						1	0
22																							0
	Total																						10

$V_{(G)}$ = Jumlah *Graph Matriks* + 1

= 10 + 1 = 11

3. Kesimpulan

Berdasarkan pengujian pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 11. Maka dapat disimpulkan bahwa pengujian *white box* pada proses rute optimasi berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.3 Pengujian *Black Box*

Proses pengujian *black box* adalah pengujian yang dilakukan dengan cara mencoba perangkat lunak, dengan memasukkan data ke dalam form-form yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi. Pengujian ini memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya semua persyaratan fungsional untuk suatu program.

4.2.3.1 Pengujian Deskripsi Titik

Pengujian deskripsi titik pada perangkat lunak pencarian jarak terdekat adalah sebagai berikut:

Tabel 4-11 Pengujian Deskripsi Titik

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data deskripsi titik	Mengisi data titik yang akan dideskripsikan	Dapat mengisi data titik yang akan dideskripsikan	[√] Diterima [ ] Ditolak

Contoh masukan: Indomaret H. Juanda			
Kasus dan Hasil Uji (Data salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
(kosong)	Tidak menampilkan apapun	Sistem tidak menampilkan apapun	[√] Diterima [ ] Ditolak

4.2.3.2 Pengujian Pengaturan Parameter

Pengujian pengaturan parameter pada perangkat lunak pencarian jarak terdekat adalah sebagai berikut:

Tabel 4-12 Pengujian Pengaturan Parameter

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data paramter	Mengisi data parameter	Dapat menyimpan data parameter	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
(kosong)	Tampilkan pesan “harus diisi”	Dapat menampilkan data kesalahan “harus diisi”	[√] Diterima [ ] Ditolak

4.2.3.3 Pengujian *Login*

Pengujian login pada *web service* pencarian jarak terdekat adalah sebagai berikut:

Tabel 4-13 Pengujian *Login*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data login Contoh: *username:* danardinu *password:* *080390*	Mengisi data login pada *field* yang tersedia dan berhasil mengakses masuk ke *web service*	Dapat mengisi data *login* dan dapat mengakses *web service*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
username: (kosong)	Tampilkan pesan “masuk gagal”	Dapat menampilkan data kesalahan “masuk gagal”	[√] Diterima [ ] Ditolak
password: (kosong)	Tampilkan pesan “masuk gagal”	Dapat menampilkan data kesalahan “masuk gagal”	[√] Diterima [ ] Ditolak

4.2.3.4 Pengujian Tambah Titik

Pengujian login pada *web service* pencarian jarak terdekat adalah sebagai berikut:

Tabel 4-14 Pengujian Tambah Titik

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Data yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data titik	Mengisi data titik	Dapat menyimpan data titik	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			

Data Masukan	Data yang diharapkan	Pengamatan	Kesimpulan
(kosong)	Tampilkan pesan “masukkan titik”	Dapat menampilkan data kesalahan “masukkan titik”	[√] Diterima [ ] Ditolak

4.2.3.5 Pengujian Pencarian Jarak

Pengujian pencarian jarak pada android:

4-15 Hasil Pengujian Pencarian Jarak

Tempat Asal	Tempat Tujuan	Hasil Percobaan Ke- (km)				
		1	2	3	4	5
Roemah Cucur	Bank Mega	1,4	1,4	1,4	1,4	1,4
Stasiun Tasikmalaya	Universitas Siliwangi	3,8	3,8	3,8	3,8	3,8
Pasar Cikurubuk	Mesjid Agung	3,2	3,2	3,2	3,2	3,2
RS Jasa Kartini	Mayasari Plaza	1,4	1,4	1,4	1,4	1,4
Stadion Dadaha	Stikes BTH	3,1	3,1	3,1	3,1	3,1
Bank Indonesia	Aunapiss Salon Motor	1,5	1,5	1,5	1,5	1,5
Mayasari Plaza	Alfamart Galunggung	1,1	1,1	1,1	1,1	1,1
Asia Plaza	TP Cieunteung	3,2	3,2	3,2	3,2	3,2
Yogya Toserba	Gereja Bunda Sapta	2,3	2,3	2,3	2,3	2,3
Hote Santosa	Kolam Renang Asia	3,2	3,2	3,2	3,2	3,2

Dari pengujian diatas maka dapat disimpulkan bahwa setiap jarak yang dicari hasilnya selalu sama atau konsisten.

4.2.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif yang dilakukan secara langsung ke lapangan yaitu perusahaan yang bersangkutan dengan membuat kuesioner bagi pengguna biasa.

Berdasarkan hasil penelitian, perhitungan presentase untuk masing-masing jawaban pada kuesioner dapat dicari dengan rumus : Y = P / Q * 100%

P = banyaknya jawaban responden tiap soal.

Q = jumlah responden

Y = nilai presentase

Berikut ini adalah hasil persentase masing-masing jawaban yang dihitung dengan menggunakan persamaan di atas. Kuesioner ini diujikan kepada 20 orang.

1. Apakah menurut Anda tampilan perangkat lunak pencarian jarak ini sudah cukup menarik dan mudah dipahami?

A. Sangat Setuju

B. Setuju

C. Cukup Setuju

D. Kurang Setuju

E. Tidak Setuju

Tabel 4-16 Hasil Pengujian Kuesioner Soal Nomor 1

Pertanyaan	No	Keterangan	Responden	Persentase(%)
1	1	Sangat setuju	1	5
	2	Setuju	9	45
	3	Cukup setuju	9	45
	4	Kurang setuju	1	5
	5	Tidak setuju	0	0
Jumlah			**20**	**100**

2. Apakah Anda setuju perangkat lunak pencarian jarak ini memiliki navigasi yang mudah?

A. Sangat Setuju

B. Setuju

C. Cukup Setuju

D. Kurang Setuju

E. Tidak Setuju

Tabel 4-17 Hasil Pengujian Kuesioner Soal Nomor 2

Pertanyaan	No	Keterangan	Responden	Persentase(%)
2	1	Sangat setuju	5	25
	2	Setuju	11	55
	3	Cukup setuju	3	15
	4	Kurang setuju	1	5
	5	Tidak setuju	0	0
Jumlah			**20**	**100**

3. Apakah Anda setuju bahwa perangkat lunak pencarian jarak ini dapat membantu Anda untuk mendapatkan informasi tentang tempat-tempat yang sering dikunjungi di kota Tasikmalaya?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Tabel 4-18 Hasil Pengujian Kuesioner Soal Nomor 3

Pertanyaan	No	Keterangan	Responden	Persentase(%)
3	1	Sangat setuju	5	25
	2	Setuju	6	30
	3	Cukup setuju	8	40
	4	Kurang setuju	1	5
	5	Tidak setuju	0	0
Jumlah			**20**	**100**

4. Apakah Anda setuju dengan adanya perangkat lunak pencarian jarak ini dapat membantu Anda untuk menemukan hasil pencarian jarak terdekat dengan lebih cepat?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Tabel 4-19 Hasil Pengujian Kuesioner Soal Nomor 4

Pertanyaan	No	Keterangan	Responden	Persentase(%)
4	1	Sangat setuju	3	15
	2	Setuju	7	35
	3	Cukup setuju	9	45
	4	Kurang setuju	1	5
	5	Tidak setuju	0	0
Jumlah			**20**	**100**

5. Apakah Anda setuju perangkat lunak pencarian jarak ini dikembangkan lebih baik lagi?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Tabel 4-20 Hasil Pengujian Kuesioner Soal Nomor 5

Pertanyaan	No	Keterangan	Responden	Persentase(%)
5	1	Sangat setuju	6	30
	2	Setuju	11	55
	3	Cukup setuju	3	15
	4	Kurang setuju	0	0
	5	Tidak setuju	0	0
Jumlah			**20**	**100**

4.2.5 Kesimpulan Hasil Pengujian

Berdasarkan pengamatan pada hasil pengujian, dapat disimpulkan bahwa:

1 Tampilan pada perangkat lunak pencarian jarak terdekat ini dapat dengan mudah dipahami oleh pengguna.

2 Perangkat lunak pencarian jarak terdekat ini memiliki navigasi yang cukup mudah untuk dijalankan oleh pengguna.

3 Perangkat lunak pencarian jarak terdekat ini dapat membantu untuk mendapatkan informasi tentang tempat-tempat yang sering dikunjungi di kota Tasikmalaya.

4 Perangkat lunak pencarian jarak terdekat ini dapat membantu menemukan hasil pencarian jarak terdekat di kota Tasikmalaya dengan lebih cepat.

5 Perangkat lunak pencarian jarak ini bisa dikembangkan agar lebih baik lagi.