BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap pengaplikasian dari tahap perancangan, pada tahap ini dipaparkan semua elemen-elemen yang akan dibangun pada sistem Ekstraksi Informasi Kemacetan Pada Media Digital.

4.1.1 Implementasi Perangkat keras

Perangkat keras yang digunakan pada saat proses implementasi dari sistem Ekstraksi Informasi Kemacetan Pada Media Digital adalah sebagai berikut:

1. Processor Intel® Core™ CPU i3 @ 2.27 GHz,
2. RAM 2 GB,
3. Hard Disk 600 GB,
4. Monitor 14 Inch,
5. *Smartphone* Android, dan
6. Modem Speedy.

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan pada saat proses implementasi dari sistem Ekstraksi Informasi Kemacetan Pada Media Digital adalah sebagai berikut:

1. Sistem Operasi Windows 7 Ultimate,
2. Sistem Operasi Android
3. Bahasa Pemrograman PHP 5.3.8,
4. Bahasa Pemrograman Android,
5. Database MySQL 5.5.16,
6. Browser Google Chrome,
7. Macromedia Dreamweaver Cs5 dan Notepad ++,
8. Eclipse Indigo dan,
9. Tools Diagram Star UML.

4.1.3 Implementasi Basis Data

Pembuatan basis data (*database*), dilakukan dengan menggunakan aplikasi MySQL. Implementasi basis data (*database*) dalam bahasa SQL adalah sebagai berikut:

Tabel 4-1 Implementasi Basis Data

No.	Nama Tabel	SQL
1	Cause	CREATE TABLE IF NOT EXISTS `cause` (`cause` varchar(65) NOT NULL, `value` int(1) NOT NULL, PRIMARY KEY (`cause`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
2	Conclusion	CREATE TABLE IF NOT EXISTS `conclusion` (`id` int(6) NOT NULL, `nilai` int(6) NOT NULL, `kesimpulan` varchar(12) NOT NULL, PRIMARY KEY (`id`)) ENGINE=MyISAM DEFAULT CHARSET=latin1;
3	Keyword	CREATE TABLE IF NOT EXISTS `keyword` (`keyword` varchar(32) NOT NULL, `value` int(1) NOT NULL, PRIMARY KEY (`keyword`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
4	tempt_data	CREATE TABLE IF NOT EXISTS `temp_data` (`temp_id` int(100) NOT NULL AUTO_INCREMENT, `temp_jalan` varchar(78) NOT NULL, `temp_jam` datetime NOT NULL, `temp_penyebab` varchar(50) DEFAULT NULL, `temp_kondisi` varchar(50) NOT NULL, PRIMARY KEY (`temp_id`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
5	Data	CREATE TABLE IF NOT EXISTS `data` (`id` int(100) NOT NULL AUTO_INCREMENT,

No.	Nama Tabel	SQL
		`jam` datetime NOT NULL, `jalan` varchar(72) NOT NULL, `penyebab` varchar(50) DEFAULT NULL, `kondisi` varchar(50) NOT NULL, `bobot` int(1) NOT NULL, PRIMARY KEY (`id`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.1.4 Implementasi Kelas *Client*

Tabel 4-2 Implementasi Kelas *Client*

No.	Nama Kelas	Nama File Fisik	Nama File Executable
1.	MainActivity	MainActivity.java	MainActivity.class
2.	HttpRequest	HttpRequest.java	HttpRequest.class
3.	MyItemizedOverlay	MyItemizedOverlay.java	MyItemizedOverlay.class
4	JSONParser	JSONParser.java	JSONParser.class
5	CheckInternetConnection	CheckInternetConnection.java	CheckInternetConnection.class

4.1.5 Implementasi Kelas *Web Service*

Tabel 4-3 Implementasi Kelas *Web Service*

No.	Nama Kelas	Nama File Fisik	Nama File Executable
1.	Cause	Cause.php	Cause.php
2.	Temp_Data	Temp_Data.php	Temp_Data.php
3.	Fetch_Url	Fetch_Url.php	Fetch_Url.php
4.	Keyword	Keyword.php	Keyword.php
5.	KNN	KNN.php	KNN.php
6.	PreProcessing	PreProcessing.php	PreProcessing.php
7.	Data	Data.php	Data.php
8.	Grab	Grab.php	Grab.php

4.1.6 Implementasi Antarmuka Pengguna (*Client*)

Implementasi antarmuka dilakukan dengan menampilkan setiap tampilan dari program yang dibangun dan pengkodeannya dalam bentuk file program.

Berikut ini adalah file-file yang digunakan dalam membuka tampilan antamuka pengguna (*client*).

Tabel 4-4 Implementasi Antarmuka Pengguna(*Client*)

No	Nama Antarmuka	Deskripsi	Nama File Executable
1.	MainActivity	Menampilkan informasi kemacetan dengan menggunakan data spasial (peta)	MainActivity.class

4.2 Pengujian Sistem

Pengujian sistem merupakan tahapan untuk melakukan serangkaian tes untuk mencoba sistem yang telah dibangun dengan tujuan untuk menemukan kesalahan-kesalahan atau kekurangan-kekurangan pada perangkat lunak yang diuji.

Dalam melakukan pengujian terhadap sistem pengklasifikasian informasi kemacetan dilakukan dengan dua macam teknik pengujian, yaitu pengujian *black box* dan *white box*. Adapun pengujian *black box* berfokus pada persyaratan fungsional dari sebuah sistem yang dibangun dan menemukan kesalahan program, sedangkan pengujian *white box* berfokus kepada struktur internal (*source code*) program untuk mengetahui apakah masih terjadi *error*.

4.2.1 Skenario Pengujian Aplikasi

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada sistem pengklasifikasian informasi kemacetan.

Tabel 4-5 Rencana Pengujian

No	Kelas Uji	Butir Uji	Jenis Pengujian
1.	MainActivity	a. Mengisi nama kota dan jalan	*Black Box*
		b. Tidak mengisi nama jalan	*Black Box*
		c. Mengisi nama jalan yang tidak ada	*Black Box*
		d. Percobaan mengisi nama kota dan	*Black Box*

		jalan dengan tanpa ada koneksi internet	
2.	Knn	Algoritma Knn	*White Box*

4.2.1.1 Pengujian *Black Box*

Kasus dan hasil uji perangkat lunak yang sudah dibangun dengan menggunakan metode *black box* berdasarkan pengamatan yang dilakukan dan menghasilkan kesimpulan dari pengujian tersebut, diantaranya sebagai berikut:

Pengujian *MainActivity*

Pengujian dilakukan dengan memasukan data untuk mendapatkan hasil yang diinginkan.

Tabel 4-6 Pengujian Kelas MainActivity

Kasus dan Hasil Uji (Data Normal)			
Data Masukkan	**Keluaran yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data berupa nama kota dan jalan	Menampilkan informasi kemacetan	Nama kota dan jalan benar dan dapat menampilkan informasi kemacetan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Nama jalan tidak diisi	Tidak dapat menampilkan informasi kemacetan dan menampilkan pesan kesalahan	Sistem menampilkan pesan kesalahan “Nama jalan harus harus diisi”	[√] Diterima [ ] Ditolak
Memasukan data nama jalan yang tidak ada	Tidak dapat login/masuk dan Tidak dapat menampilkan informasi kemacetan dan menampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan “Lokasi tidak ditemukan”	[√] Diterima [ ] Ditolak
Memasukan data dengan tanpa ada	Tidak dapat menampilkan informasi kemacetan dan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

koneksi internet	menampilkan pesan kesalahan	"Tidak ada koneksi internet"	

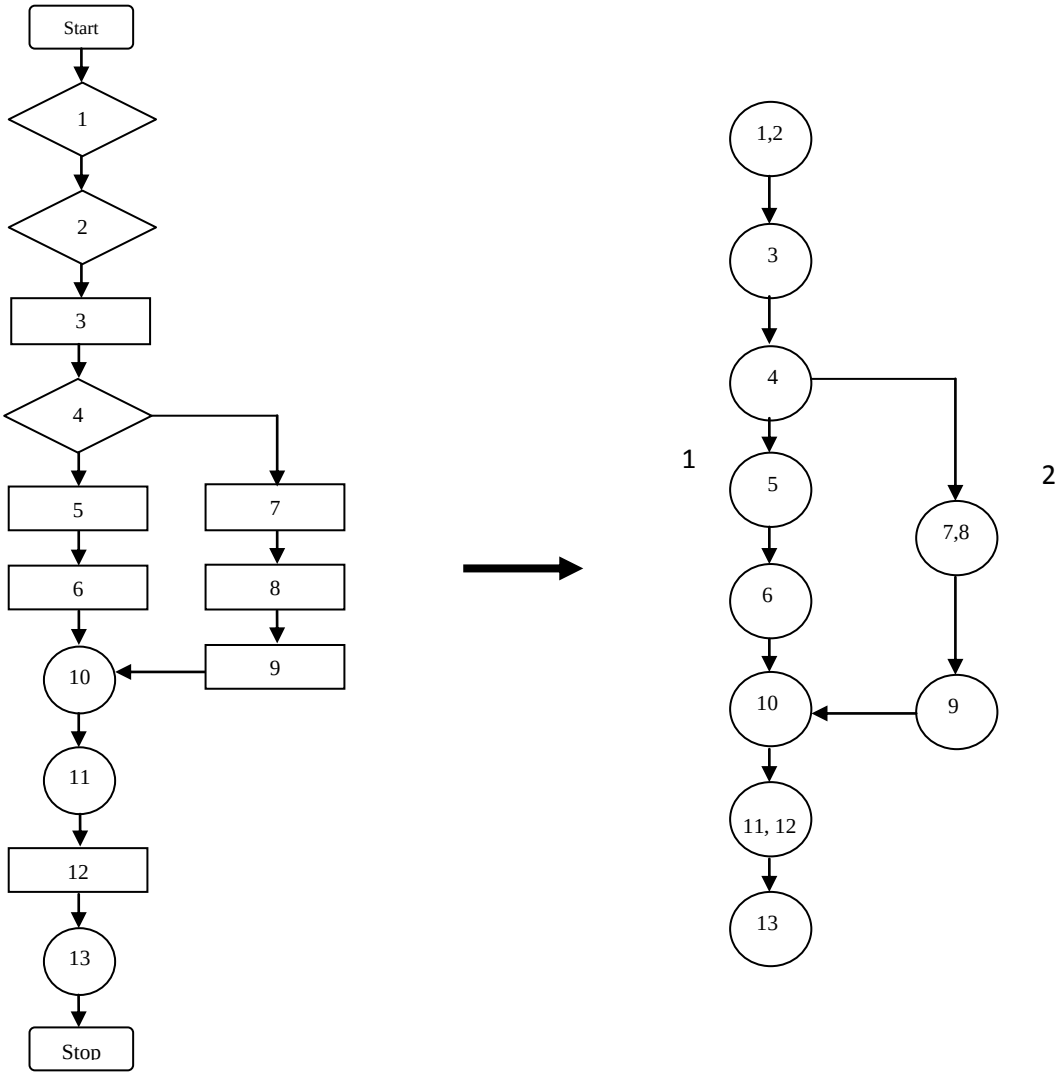
4.2.1.2 Pengujian *White Box*

Pengujian Algoritma KNN

Pengujian pada algoritma KNN menggunakan pengujian *white box*, hal tersebut dilakukan untuk mengukur kinerja logika berdasarkan *pseudocode* yang telah dibuat pada tahap analisis. Berikut adalah tahapan-tahapan pada pengujian algoritma KNN:

1. Mengubah bentuk *pseucode* menjadi *flowchart* kemudian menjadi *flowgraph*

```
1   foreach (query in querySet) do
2   foreach (training in trainingSet) do
3   map ← SortedMap(k);
4   if (query != 0) do
5   d ← EuclideanDistance(query, training);
6   map(d, training);
7   else query ← 0;
8        d ← EuclideanDistance(query, training);
9        map(d, training);
10  endif
11  endfor
12  query↑label ← majorityVote(map);
13  endfor
```



Gambar 4-1 Merubah flowchart menjadi flowgraph

2. Cyclometic Complexity

   V(G) = E – N + 2

   V(G) = 10 – 10 + 2

   V(G) = 2

   E = Jumlah Busur pada flow graph

   N = Jumlah Simpul pada flow graph

3. Independent Path

   Path 1 : 1,2,3,4,5,6,10,11,12,13

   Path 2 : 1,2,3,4,7,8,9,10,11,12,13

4. Hitung Jumlah region

   Jumlah Region = Jumlah Kurva Tertutup + 1 (Region Terluar)

   Jumlah Region = 1 + 1

   Jumlah Region = 2

5. Hitung jumlah predicate Node

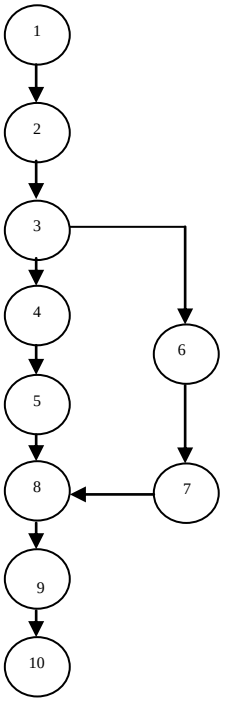
   Predicate Node = 1

   V(G) = Jumlah Predicate Node + 1

   V(G) = 1 + 1

   V(G) = 2

6. Penomoran ulang flowgraph



Gambar 4-2 Penomoran ulang flowgraph

7. Graph matriks

Tabel 4-7 Graph Matrix

Node	1	2	3	4	5	6	7	8	9	10	Jumlah
1		1									0
2			1								0
3				1							0
4					1	1					1
5								1			0
6							1				0
7								1			0
8									1		0
9										1	0
10											0
Jumlah											1

$$V(g) = x + 1$$

$$V(g) = 1 + 1$$

$$V(G) = 2$$

8. Kesimpulan Pengujian White Box

Berdasarkan pengujian pada setiap metode, dihasilkan nilai Cyclomatic Complexity yang sama yaitu dua. Maka dapat disimpulkan bahwa pengujian *white box* pada algoritma KNN berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.1.3 Pengujian Beta

Pengujian beta merupakan tahapan dimana untuk mengetahui tanggapan dan penilaian dari pengguna terhadap perangkat lunak yang telah dibangun. Untuk mengetahui hal tersebut, penulis menggunakan metode kuesioner yang telah disebarkan kepada 30 responden yang memiliki *smartphone* Android.

4.2.1.3.1 Aturan pengujian

Dalam melakukan perhitungan kuesioner, penulis menggunakan teknik *skala linkert*. Adapun aturan-aturan yang digunakan dalam membantu proses perhitungan yaitu, terdapat lima pilihan jawaban atau pernyataan yang terdiri dari sangat setuju, setuju, sedang, tidak setuju dan sangat tidak setuju. Setiap pernyataan tersebut mempunyai bobot nilai seperti berikut:

1. Sangat Setuju (A) : 5 poin (80-100) %
2. Setuju (B) : 4 poin (60-79) %
3. Sedang (C) : 3 poin (40-59) %
4. Tidak Setuju (D) : 2 poin (20-39) %
5. Sangat Tidak Setuju (E) : 1 poin (0-19) %

4.2.1.3.2 Kuesioner

Data hasil kuisioner yang telah diberikan dapat dicari presentasinya dari masing-masing jawaban dengan menggunakan rumus:

$$P = \frac{S}{skorideal} x100\%$$

Keterangan:

P = Nilai persentase yang dicari

S = Jumlah frekuensi jawaban dikalikan dengan nilai pernyataan

Skor ideal = Nilai pernyataan tertinggi dikalikan dengan jumlah sampel

Berikut ini merupakan hasil nilai persentase masing-masing jawaban yang telah dihitung dengan menggunakan rumus di atas.

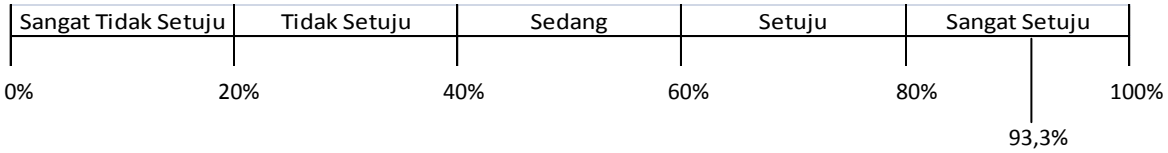
1. Apakah perangkat lunak Informasi Kemacetan tersebut mudah digunakan?

Tabel 4-8 Perhitungan Persentase Pertanyaan Pertama

Pertanyaan	Pernyataan	Skor	Frekuensi	S
1	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	0	0

Pertanyaan	Pernyataan	Skor	Frekuensi	S
	Tidak Setuju (D)	4	10	40
	Sangat Tidak Setuju (E)	5	20	100
Jumlah			30	140

$$P = \frac{140}{150} x100 = 93{,}3\%$$



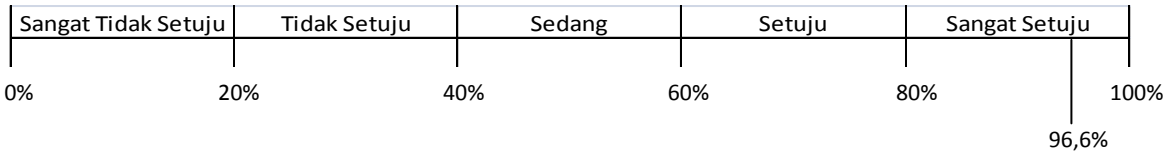
Gambar 4-3 Hasil Perhitungan Kuesioner Pertanyaan Kesatu

2. Apakah perangkat lunak informasi kemacetan menginformasikan kemacetan sesuai dengan fakta?

Tabel 4-9 Perhitungan Persentase Pertanyaan Kedua

Pertanyaan	Pernyataan	Skor	Frekuensi	S
2	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	0	0
	Tidak Setuju (D)	4	5	20
	Sangat Tidak Setuju (E)	5	25	125
Jumlah			30	145

$$P = \frac{145}{150} x100 = 96{,}6\%$$



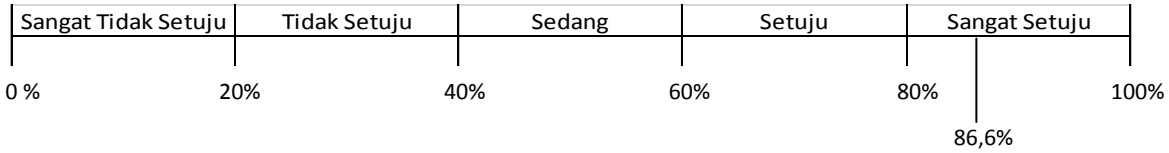
Gambar 4-4 Diagram Hasil Perhitungan Kuesioner Pertanyaan Kedua

3. Menurut anda, apakah informasi kemacetan yang menggunakan peta itu efektif?

Tabel 4-10 Perhitungan Persentase Pertanyaan Ketiga

Pertanyaan	Pernyataan	Skor	Frekuensi	S
3	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	2	6
	Tidak Setuju (D)	4	6	24
	Sangat Tidak Setuju (E)	5	22	110
Jumlah			30	130

$$P = \frac{130}{150} x100 = 86{,}6\%$$



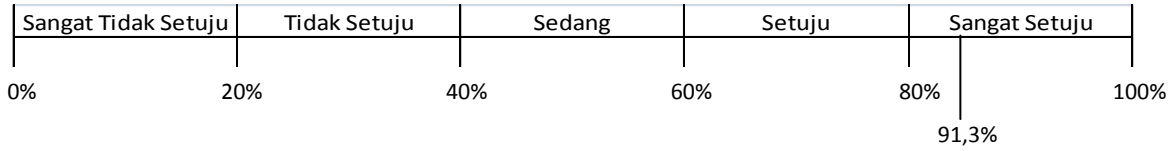
Gambar 4-5 Hasil Perhitungan Kuesioner Pertanyaan Ketiga

4. Apakah perangkat lunak informasi kemacetan tersebut membantu anda?

Tabel 4-11 Perhitungan Persentase Pertanyaan Keempat

Pertanyaan	Pernyataan	Skor	Frekuensi	S
4	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	2	6
	Tidak Setuju (D)	4	9	36
	Sangat Tidak Setuju (E)	5	19	95
Jumlah			30	137

$$P = \frac{137}{150} x100 = 91{,}3\%$$



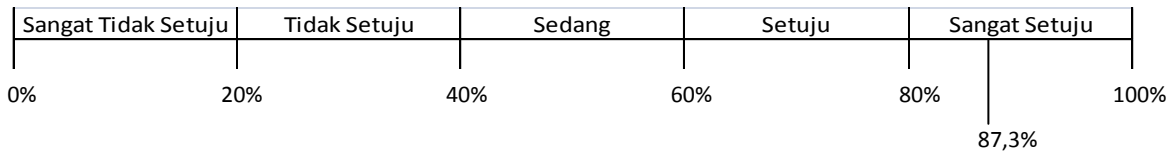
Gambar 4-6 Hasil Perhitungan Kuesioner Pertanyaan Keempat

5. Menurut anda, informasi kemacetan yang disajikan dalam perangkat lunak tersebut jelas?

Tabel 4-12 Perhitungan Persentase Pertanyaan Kelima

Pertanyaan	Pernyataan	Skor	Frekuensi	S
5	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	4	16
	Tidak Setuju (D)	4	15	60
	Sangat Tidak Setuju (E)	5	11	55
Jumlah			30	131

$$P = \frac{131}{150} x100 = 87{,}3\%$$



Gambar 4-7 Diagram Hasil Perhitungan Kuesioner Pertanyaan Kelima

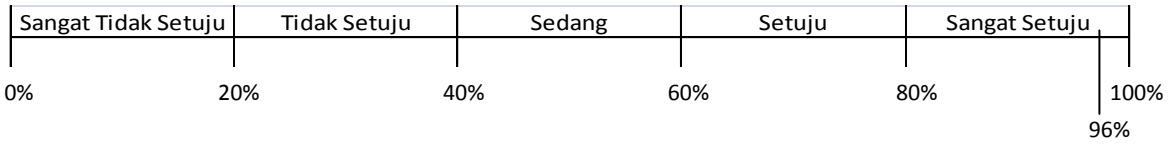
6. Puaskah anda dengan hasil informasi kemacetan yang ada pada perangkat lunak tersebut?

Tabel 4-13 Perhitungan Persentase Pertanyaan Keenam

Pertanyaan	Pernyataan	Skor	Frekuensi	S
6	Sangat Setuju (A)	1	0	0
	Setuju (B)	2	0	0
	Sedang (C)	3	0	0

Pertanyaan	Pernyataan	Skor	Frekuensi	S
	Tidak Setuju (D)	4	6	24
	Sangat Tidak Setuju (E)	5	24	120
Jumlah			30	144

$$P = \frac{144}{150} x100 = 96\%$$



Gambar 4-8 Diagram Hasil Perhitungan Kuesioner Pertanyaan Keenam

Berdasarkan hasil yang didapat, maka didapatkan kesimpulannya yaitu sebagai berikut:

1. Responden sangat setuju bahwa perangkat lunak informasi kemacetan sangat mudah digunakan.
2. Responden sangat setuju bahwa informasi kemacetan yang diinformasikan oleh perangkat lunak informasi kemacetan sangat sesuai dengan fakta yang terjadi.
3. Responden sangat setuju bahwa informasi kemacetan yang disajikan dengan menggunakan peta merupakan langkah yang sangat efektif.
4. Responden sangat setuju bahwa informasi dari perangkat lunak informasi kemacetan sangat membantu.
5. Responden setuju bahwa informasi kemacetan yang disajikan sudah jelas.
6. Responden sangat setuju bahwa hasil informasi dari perangkat lunak informasi kemacetan sangat memuaskan.