BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Tahap implementasi merupakan tahap penerjemahan perancangan berdasarkan hasil analisis ke dalam suatu bahasa pemrograman yang digunakan serta penerapan perangkat lunak yang dibangun pada lingkungan yang sesungguhnya. Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan terhadap aplikasi, sehingga pengguna dapat memberi masukan demi berkembangnya aplikasi yang telah dibangun. Adapun pembahasan untuk implementasi sistem meliputi kebutuhan perangkat keras, perangkat lunak, pengujian *alpha*, dan pengujian *beta*.

4.1.1 Implementasi Perangkat Pendukung

Perangkat keras dan perangkat lunak yang digunakan untuk melakukan implementasikan dan pengujian dari aplikasi adalah sebagai berikut:

4.1.1.1 Implementasi Perangkat Keras

Implementasi perangkat keras merupakan relisasi dari perancangan kebutuhan perangkat keras. Aplikasi keamanan sms ini diimplementasikan pada perangkat android dengan spesifikasi seperti terlihat pada tabel 4.1.

Tabel 4.1 Spesifikasi Perangkast Keras

Perangkat Keras	Spesifikasi
CPU	MTK 6589WCPU Quad-core 1.2 GHz
RAM	1 Gb
Harddisk	4 Gb
Resolusi Layar	720 x 1280 pixels

4.1.1.2 Implementasi Perangkat Lunak

Implementasi perangkat lunak merupakan proses instalasi perangkat lunak, sehingga dapat beroperasi dengan benar. Perangkat lunak yang digunakan yaitu seperti terlihat pada tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak

Perangkat Keras	Keterangan
Sistem Operasi	Windows 8 Professional
Bahasa Pemrograman	Java
Tools Pemrograman	Eclipse
Perangkat Pendukung	JDK 7
Picture Viewer	Windows Photo Viewer

4.1.2 Implementasi Aplikasi

Aplikasi keamanan sms merupakan sebuah aplikasi yang dibuat untuk membantu untuk menyamarkan isi sms agar terjamin kerahasiaan dan keamanan isi dari sms tersebut dengan menyandikan sms yang akan dikirimkan. Aplikasi kemanan sms menggunakan algoritma kunci publik RSA untuk menyadikan pesan dilengkapi dengan *management key*-nya.

4.1.3 Implementasi Antarmuka

Implementasi antarmuka aplikasi keamanan sms menggunakan algoritma RSA adalah sebagai berikut :

1. **Tampilan Buat Menu Password**

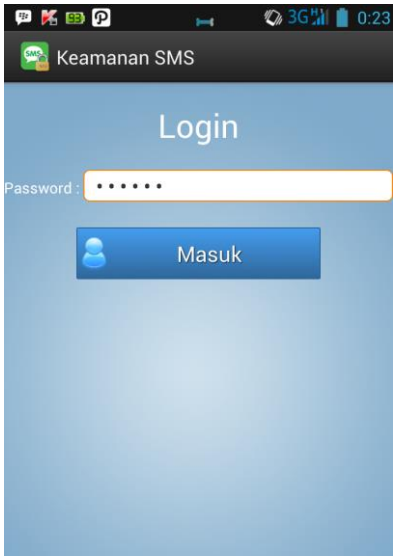
   Antarmuka tampilan buat password merupakan halaman yang akan muncul ketika pertama kali aplikasi di *install*. Dalam tampilan ini pengguna diharuskan membuat password untuk masuk ke dalam aplikasi. Tampilan buat password terlihat pada gambar 4.1.



Gambar 4.1 Tampilan Menu Buat *Password*

2. Tampilan Menu *Login*

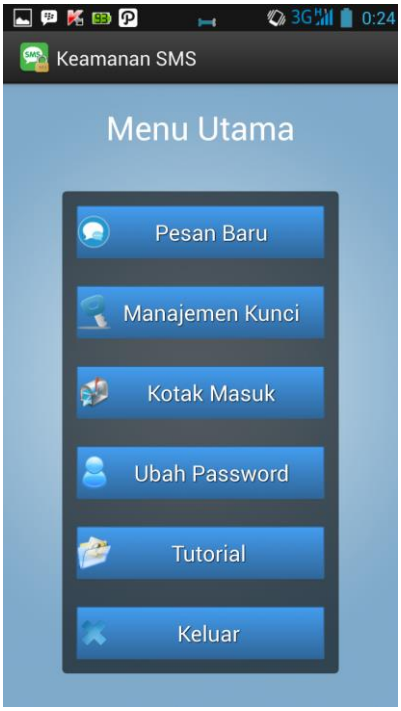
Antarmuka tampilan login merupakan halaman yang akan muncul ketika pengguna telah membuat password dang akan muncul pertama kali jika aplikasi dibuka setelah pengguna membuat password pada saat setelah instalasi. Dalam tampilan ini pengguna diharuskan memasukan password yang telah dibuat untuk dapat masuk ke dalam aplikasi. Tampilan login aplikasi keamanan SMS terlihat pada gambar 4.2.



Gambar 4.2 Tampilan Menu *Login*

3. Tampilan Halaman Menu Utama

Antarmuka tampilan halaman utama merupakan halaman utama dari aplikasi keamanan sms. Pada halaman utama ini terdapat 6 tombol yaitu pesan baru, manajemen kunci, kotak masuk, ubah password, tutorial dan keluar. Tampilan menu utama aplikasi keamanan SMS terlihat pada gambar 4.3.



Gambar 4.3 Tampilan Menu Utama

4. Tampilan Menu Pesan Baru

Antarmuka pesan baru merupakan halaman setelah pengguna menekan button new message pada halaman utama. Pada halaman ini pengguna dapat melakukan enkripsi pesan dan pengiriman pesan terenkripsi. Tampilan pesan baru aplikasi keamanan SMS terlihat pada gambar 4.4.



Gambar 4.4 Tampilan Menu Pesan Baru

5. Tampilan Manajemen Kunci

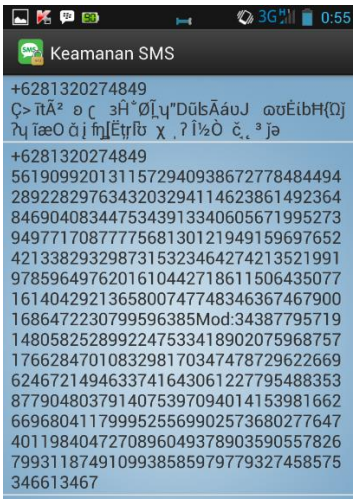
Antarmuka tampilan manajemen kunci merupakan halaman dimana pengguna dapat melakukan generate key yang digunakan dalam proses enkripsi pesan. Pengguna juga dapat mengirimkan key yang telah digenerate kepada calon penerima pesan terenkripsi. Tampilan manajemen kunci aplikasi keamanan SMS terlihat pada gambar 4.5.



Gambar 4.5 Tampilan Menu Manajemen Kunci

6. Tampilan Kotak Masuk

Antarmuka tampilan inbox merupakan halaman list kotak masuk pesan tersimpan. Tampilan kotak masuk aplikasi keamanan SMS terlihat pada gambar 4.6.



Gambar 4.6 Tampilan Menu Kotak Masuk

7. Tampilan *Detail* Kotak Masuk

Antarmuka tampilan detail kotak masuk merupakan halaman setelah pengguna memilih pesan yang akan didekripsi atau memilih key yang telah dikirimkan untuk disimpan. Tampilan *detail* kotak masuk aplikasi keamanan SMS terlihat pada gambar 4.7.



Gambar 4.7 Tampilan Detail Kotak Masuk

8. Tampilan Ubah *Password*

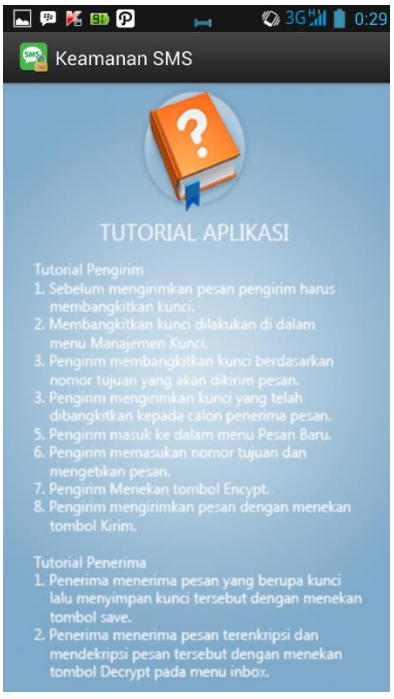
Antarmuka tampilan ubah password merupakan halaman dimana pengguna dapat merubah password untuk masuk ke aplikasi. Tampilan ubah password aplikasi keamanan SMS terlihat pada gambar 4.8.



Gambar 4.8 Tampilan Menu Ubah *Password*

9. Tampilan Tutorial

Antarmuka tampilan tutorial merupakan halaman dimana pengguna dapat melihat dan membaca cara menggunakan aplikasi. Tampilan tutorial aplikasi keamanan SMS terlihat pada gambar 4.9.



Gambar 4.9 Tampilan Tutorial

4.2 Pengujian Aplikasi

Pengujian aplikasi dilakukan untuk menentukan kesalahan atau kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibangun sudah sesuai dengan perancangan yang dibuat dan memenuhi kebutuhan pengguna.

Pengujian yang dilakukan yaitu pengujian *alpha* dan pengujian *beta*. Pengujian *alpha* dilakukan dengan pengujian *black box* dan *white box*. Pengujian *black box* digunakan untuk mengetahui kesalahan proses secara fungsional sedangkan *white box* digunakan untuk menguji performa dari metode yang digunakan. Pengujian *beta* digunakan untuk mengetahui tanggapan pengguna terhadap aplikasi dengan menggunakan kuesioner.

4.2.1 Pengujian Alpha

Pengujian fungsional yang digunakan untuk menguji aplikasi adalah metode pengujian *alpha*. Pengujian *alpha* dilakukan dengan menggunakan metode *black box* dan *white box.* Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak sedangkan *white box* berfokus pada algoritma dari metode yang digunakan. Skenario pengujian dari aplikasi keamanan sms dapat dilihat pada tabel 4.3.

Tabel 4.3 Skenario Pengujian Alpha

No	Komponen Yang Diuji	Skenario Pengujian	Jenis Pengujian
1	Menu Buat Password	Instalasi Aplikasi	*Black box*
		Memasukan Password Baru	*Black box*
		Menekan Tombol Buat Password	*Black box*
2	Menu *Login*	Memasukan *Password*	*Black box*
		Menekan Tombol *Login*	*Black box*
3	Menu *Pesan Baru*	Menekan Tombol Pesan Baru	*Black box*
		Memasukan No. Tujuan	*Black box*
		Mengetikan Pesan	*Black box*
		Menekan Tombol *Encrypt*	*Black box*
		Menekan Tombol Kirim	*Black box*

4	Menu Manajemen Kunci	Memilih Tombol Manajemen Kunci	*Black box*
		Memasukan No. Penerima	*Black box*
		Menekan Tombol Bangkitkan Kunci	*Black box*
		Menekan Tombol Simpan Kunci	*Black box*
		Menekan Tombol Kirim Kunci	*Black box*
5	Menu Kotak Masuk	Menekan Tombol Kotak Masuk	*Black box*
		Memilih Pesan	*Black box*
6	Menu *Detail* Kotak Masuk	Menekan Pesan Masuk	*Black box*
		Menekan Tombol Simpan Kunci Jika Pesan Masuk Merupakan Kunci	*Black box*
		Menekan Tombol *Decrypt* Jika Pesan Masuk Merupakan Pesan Terenkripsi	*Black box*
7	Menu Ubah Password	Menekan Tombol Ubah Password	*Black box*
		Ketikan Password Baru	*Black box*
		Menekan Tombol Ubah Password	*Black box*
8	Menu Tutorial	Menekan tombol Tutorial	*Black box*
		Baca tutorial	*Black box*
9	Menu Keluar	Menekan Tombol Exit	*Black box*
		Memilih Tombol Ya	*Black box*
		Memilih Tombol Tidak	*Black box*

4.2.1.1 Kasus dan Hasil Pengujian *Black Box*

Pengujian perangkat lunak yang sudah dibangun dilakukan dengan menggunakan metode *black box*, berdasarkan pengamatan yang sudah dilakukan maka dihasilkan kesimpulan dari pengujian pada tabel 4.4.

Tabel 4.4 Pengujian *Black Box*

No	Kasus yang Diuji	Skenario dan Hasil Uji		
		Skenario Uji	Hasil yang diharapkan	Kesimpulan
1	Menu Buat *Password*	Instalasi Aplikasi	Aplikasi ter-install pada ponsel android	[√] Berhasil [ ] Tidak Berhasil
		Memasukan *Password* Baru	Sistem menerima *password* yang diketik	[√] Berhasil [ ] Tidak Berhasil

		Menekan Tombol Buat *Password*	Sistem menyimpan password dan masuk ke halaman *login*	[√] Berhasil [ ] Tidak Berhasil
2	Menu *Login*	Memasukan *Password*	Sistem menerima *password* yang diketik	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol *Login*	Sistem masuk ke halaman utama jika *password* benar dan jika *password* salah sistem akan tetap di halaman *login*	[√] Berhasil [ ] Tidak Berhasil
3	Menu Pesan Baru	Menekan Tombol Pesan Baru	Sistem menampilkan tampilan pesan baru	[√] Berhasil [ ] Tidak Berhasil
		Memasukan No. Tujuan	Sistem menampilkan nama dan nomor penerima pesan	[√] Berhasil [ ] Tidak Berhasil
		Mengetikan *Pesan*	Sistem menampilkan pesan plaintext yang diketikan pengguna	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol *Encrypt*	Sistem menampilkan *ciphertext* yang akan dikirimkan	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Kirim	Sistem mengirimkan *ciphertext* melaui sms	[√] Berhasil [ ] Tidak Berhasil
4	Menu Manajemen Kunci	Memilih Tombol Manajemen Kunci	Sistem masuk ke menu manajemen kunci	[√] Berhasil [ ] Tidak Berhasil
		Memasukan No. Penerima	Sistem menerima no. tujuan yang diketik	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Bangkitkan Kunci	Sistem membangkitkan pasangan kunci	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Simpan Kunci	Sistem menyimpan kunci ke dalam *database*	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Simpan Kunci	Sistem mengirimkan kunci pr4at kepada no. yang dituju	[√] Berhasil [ ] Tidak Berhasil
5	Menu Kotak Masuk	Menekan Tombol Kotak Masuk	Sistem menampilkan *list* kotak masuk	[√] Berhasil [ ] Tidak Berhasil
		Memilih Pesan	Sistem menampilkan *detail* kotak masuk	[√] Berhasil [ ] Tidak Berhasil
6	Menu *Detail* Pesan Masuk	Menekan Tombol Pesan Masuk	Sistem menampilkan pesan	[√] Berhasil [ ] Tidak Berhasil

		Menekan Tombol Simpan Kunci Jika Pesan Masuk Merupakan Kunci	Sistem menyimpan kunci ke dalam *database*	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol *Decrypt* Jika Pesan Masuk Merupakan Pesan Terenkripsi	Sistem men-dekripsi pesan	[√] Berhasil [ ] Tidak Berhasil
7	Menu Ubah Password	Menekan Tombol Ubah Password	Sistem menampilkan menu ubah password	[√] Berhasil [ ] Tidak Berhasil
		Ketikan Password Baru	Sistem menerima password yang diketik	[√] Berhasil [ ] Tidak Berhasil
		Menekan Tombol Ubah Password	Sistem merubah password lama dengan password baru	[√] Berhasil [ ] Tidak Berhasil
8	Menu Tutorial	Menekan tombol Tutorial	Sistem menampilkan tutorial	[√] Berhasil [ ] Tidak Berhasil
9	Menu Keluar	Memilih Tombol Keluar	Sistem akan menampilkan pilihan pesan	[√] Berhasil [ ] Tidak Berhasil
		Memilih Tombol Yes	Sistem keluar dari aplikasi	[√] Berhasil [ ] Tidak Berhasil
		Memilih Tombol No	Sistem kembali ke halaman utama	[√] Berhasil [ ] Tidak Berhasil

4.2.1.2 Kasus dan Hasil Pengujian *White Box*

Berikut ini adalah kasus menguji perangkat lunak yang telah dibangun dengan menggunakan metode *white box.* Terdapat 3 pengujian *white box* yaitu : *encrypt*, *decrypt* dan *generate k*ey.

1. Pengujian RSA *Encryption*

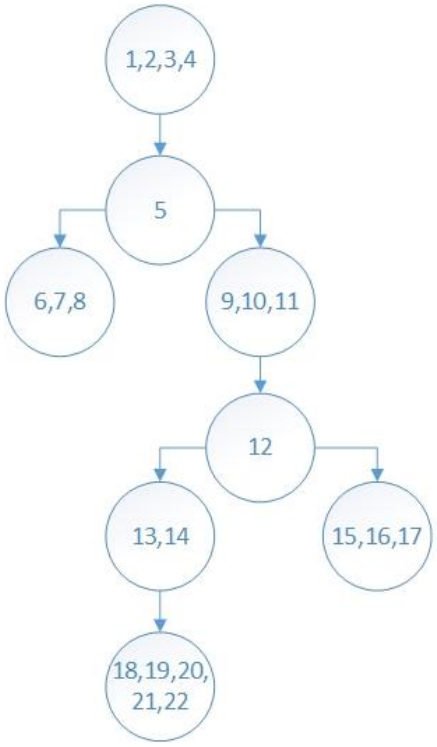
Pengujian *encryption* pada algoritma RSA terlihat pada tabel 4.5.

Tabel 4.5 Pengujian *White Box RSA Encryption*

Line	Source
1	`btnEncrypt.setOnClickListener(new OnClickListener() {`
2	`  @Override`
3	`  public void onClick(View v) {`
4	`    Key key;`
5	`      if ((key = MainActivity.dso`
6	`        .getKey(getNumber(textDestinationNumber.getText() .toString())) == null) {`
7	`        Toast.makeText(getApplicationContext(), "Not already set",`
8	`        Toast.LENGTH_LONG).show();`
9	`      } else {`
10	`        String encrypted = MainActivity.rsa.enc(editMessage`
11	`          .getText().toString(), key.getMod());`
12	`          try {`
13	`              byte[] utf16 = encrypted.getBytes("UTF-16LE");`
14	`              encrypted = new String(utf16, "UTF-16LE");`
15	`          } catch (UnsupportedEncodingException e) {`
16	`              e.printStackTrace();`
17	`              Log.d("haha", "hihi");`
	`          }`
18	`          editEncryptedMessage.setText(encrypted);`
19	`          InboxActivity.message = encrypted;`
20	`          Log.d("Encrypted", encrypted);`
21	`          DetailInboxActivity.message = encrypted;`
22	`          editPrivateKey.setText(key.getPrivateKey() + "Mod:"+ key.getMod());`
	`      }`
	`  }`
	`});`

a) Flow Graph RSA Encryption

Berdasarkan tabel 4.5 maka dapat dibentuk *flow graph* seperti pada gambar 4.10.



Gambar 4.10 *Flow Graph RSA Encryption*

Keterangan :

◯ = Menggambarkan kondisi

→ = Menggambarkan aksi

Dari gambar 4.1 dapat dihitung *cyclomatic complexity* sebagai berikut:

V(G) = E – N + 2

V(G) = 8 *edge* – 7 *node* + 2 = 3

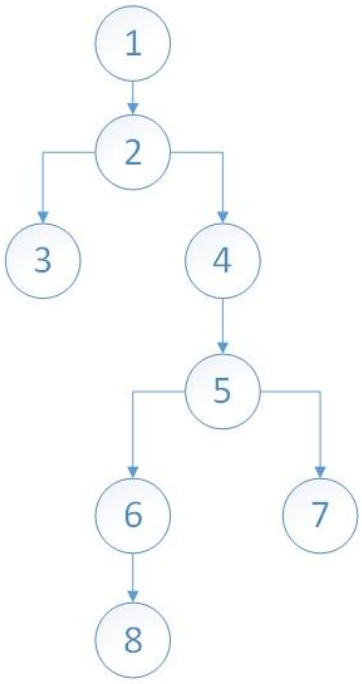
Jadi, *cyclomatic complexity* untuk gambar 4.1 adalah 3. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 3 *path* yang terdiri dari:

Path 1: 1-2-3-4-3-5-6-7-6-8

Path 2: 1-2-3-4-3-5-9-10-11-12-13-14-18-19-10-21-22

Path 3: 1-2-3-4-3-5-9-10-11-12-15-16-17

Maka gambar 4.10. disederhanakan berdasarkan kondisi atau simpul pada gambar 4.11.



Gambar 4.11 Hasil Penyederhanaan *Flow Graph RSA Encryption*

b) Graph Matrix RSA Encryption

Graph matrix encryption dari algoritma RSA terlihat pada tabel 4.6.

Tabel 4.6 *Graph Matrix RSA Encryption*

	2	**3**	**4**	**5**	**6**	**7**	**8**	**sum**
1	1							0
2		1	1					1
3								0
4				1				0
5					1	1		1
6							1	0
Total								**2**

V(G) = Jumlah *Graph Matric* + 1

V(G) = 2 + 1

V(G) = 3

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 3, Maka dapat disimpulkan bahwa pengujian *white box* pada proses RSA *Encryption* berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

2. Pengujian RSA *Decryption*

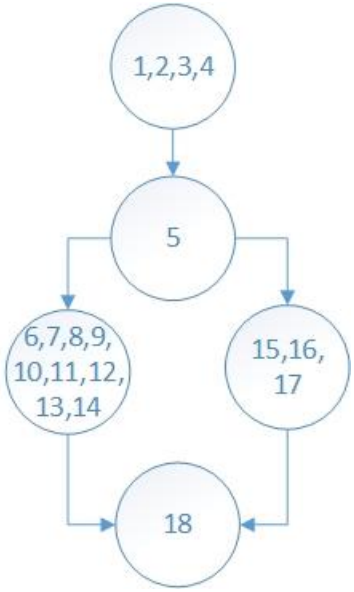
Pengujian *decryption* dari algoritma RSA terlihat pada tabel 4.7.

Tabel 4.7 Pengujian *White Box RSA Decryption*

Line	Source
1	`btnDecrypt.setOnClickListener(new OnClickListener() {`
2	`  @Override`
3	`  public void onClick(View v) {`
4	`    Key key = null;`
5	`      if ((key = MainActivity.dao.getKeyR(editSender.getText().toString())) != null) {`
6	`        Log.d("Keyz", "Private: " + key.getPrivateKey());`
7	`        Log.d("Keyz", "Mod: " + key.getMod());`
8	`        String decrypted = MainActivity.rsa`
9	`          .toSingleString(MainActivity.rsa.dec(`
10	`            editEncryptedMessageInbox.getText()`
11	`              .toString(), key.getPrivateKey(),`
12	`            key.getMod()));`
13	`        editPrivateKeyMessageInbox.setText(key.getPrivateKey());`
14	`        editDecryptedMessage.setText(decrypted);`
15	`      } else {`
16	`        Toast.makeText(getApplicationContext(),`
17	`          "Key not available", Toast.LENGTH_LONG).show();`
	`      }`
	`    }`
	`});`

a) Flow Graph RSA Decryption

Berdasarkan tabel 4.7 maka dapat dibentuk *flow graph* seperti pada gambar 4.12.



Gambar 4.12 *Flow Graph RSA Decryption*

Keterangan :

◯ = Menggambarkan kondisi

→ = Menggambarkan aksi

Dari gambar 4.3 dapat dihitung *cyclomatic complexity* sebagai berikut:

V(G) = E – N + 2

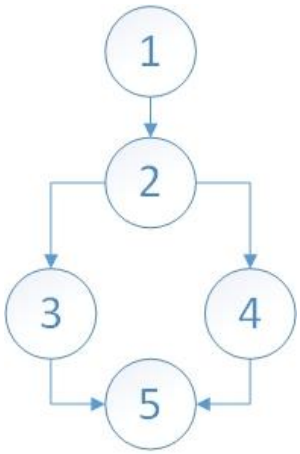
V(G) = 5 *edge* – 5 *node* + 2 = 2

Jadi, *cyclomatic complexity* untuk gambar 4.1 adalah 2. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 2 *path* yang terdiri dari:

Path 1: 1-2-3-4-3-5-6-7-6-8-9-10-11-12-13-14-18

Path 2: 1-2-3-4-3-5-15-16-17-18

Maka gambar 4.12 disederhanakan berdasarkan kondisi atau simpul pada gambar 4.13.



Gambar 4.13 Hasil Penyederhanaan *Flow Graph RSA Decryption*

b) Graph Matrix RSA Encryption

Graph matrix decryption dari algoritma RSA terlihat pada tabel 4.8.

Tabel 4.8 *Graph Matrix RSA Decryption*

	2	**3**	**4**	**5**	**sum**
1	1				0
2		1	1		1
3			1		0
4				1	0
Total					**1**

V(G) = Jumlah *Graph Matric* + 1

V(G) = 1 + 1

V(G) = 2

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 2, Maka dapat disimpulkan bahwa pengujian *white box* pada proses RSA *Decryption* berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

3. Pengujian *Generate Key*

Pengujian *generate key* pada algoritma RSA terlihat pada tabel 4.9.

Tabel 4.9 Pengujian *White Box Generate Key*

Line	Source
1	`public void encrypt(int m){`
2	`getPQ();`
3	`computeN();`
4	`computePhi();`
5	`getE();`
6	`computeD();`
7	`computePublicKey();`
8	`computePr4ateKey();`
9	`int z=(int)(Math.pow(m, publicKey[0])%publicKey[1]);`
10	`int x=(int)(Math.pow(z, pr4ateKey[0])%pr4ateKey[1]);`
11	`}`
12	`public void getPQ(){`
13	`p=getPrimeNumber();`
14	`q=getPrimeNumber();`
15	`}`
16	`public int getPrimeNumber(){`
17	`int r;`
18	`outer:`
19	`while(true){`
20	`r=(int)(Math.random()*100);`
21	`for(int i=2; i<r; i++){`
22	`if(r%i==0){`
23	`break outer;`
24	`}`
25	`}`
26	`}`
27	`return r;`
28	`}`
29	`public void computeN(){`
30	`n=p*q;`
31	`}`
32	`public void computePhi(){`

```
33      phi=(p-1)*(q-1);
34  }

35  public void getE(){
36      int r=(int)(Math.random()*phi);
37      if(r-1>1){
38        r=r-1;
39      }
40      e=r;
41  }

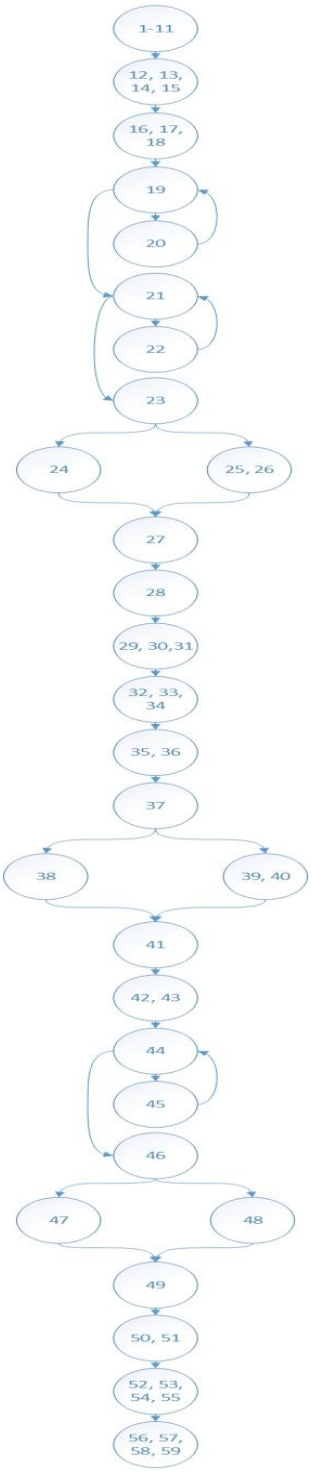
42  public void computeD(){
43      int r;
44      while(true){
45        r=(int)(Math.random()*100);
46        if((r*e)%phi==1){
47            d=r;
48            break;
49        }
50      }
51  }

52  public void computePublicKey(){
53      publicKey[0]=e;
54      publicKey[1]=n;
55  }

56  public void computePr4ateKey(){
57      pr4ateKey[0]=d;
58      pr4ateKey[1]=n;
59  }
```

a) Flow Graph Key Generate

Berdasarkan tabel 4.13. maka dapat dibentuk *flow graph* seperti pada gambar 4.14



Gambar 4.14 *Flow Graph Key Generate*

Keterangan :

◯ = Menggambarkan kondisi

→ = Menggambarkan aksi

Dari gambar 4.13 dapat dihitung *cyclomatic complexity* sebagai berikut:

V(G) = E – N + 2

V(G) = 34 *edge* – 29 *node* + 2 = 7

Jadi, *cyclomatic complexity* untuk gambar 4.13 adalah 4. Berdasarkan *cyclomatic complexity* tersebut, maka terdapat 2 *path* yang terdiri dari:

Path 1: 1-11-12-13-14-15-16-17-18-19-20-19-21-22-21-23-25-26-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-45-44-46-48-49-50-51-52-53-54-55-56-57-58-59

Path 2: 1-11-12-13-14-15-16-17-18-19-21-22-21-23-25-26-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-45-44-46-48-49-50-51-52-53-54-55-56-57-58-59

Path 3: 1-11-12-13-14-15-16-17-18-19-21-23-25-26-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-45-44-46-48-49-50-51-52-53-54-55-56-57-58-59

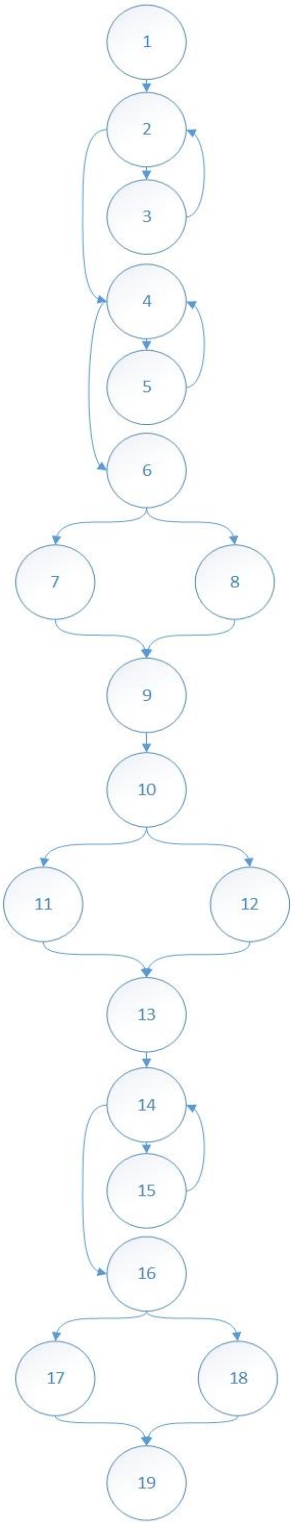
Path 4: 1-11-12-13-14-15-16-17-18-19-20-19-21-22-21-23-24-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-45-44-46-48-49-50-51-52-53-54-55-56-57-58-59

Path 5: 1-11-12-13-14-15-16-17-18-19-20-19-21-22-21-23-24-28-29-30-31-32-33-34-35-36-37-38-40-41-42-43-44-45-44-46-48-49-50-51-52-53-54-55-56-57-58-59

Path 6: 1-11-12-13-14-15-16-17-18-19-20-19-21-22-21-23-25-26-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-46-48-49-50-51-52-53-54-55-56-57-58-59

Path 7: 1-11-12-13-14-15-16-17-18-19-20-19-21-22-21-23-25-26-28-29-30-31-32-33-34-35-36-37-39-40-41-42-43-44-45-44-46-47-49-50-51-52-53-54-55-56-57-58-59

Maka gambar 4.14. disederhanakan berdasarkan kondisi atau simpul pada gambar 4.15.



Gambar 4.15 Penyederhanaan *Flow Graph Key Generate*

b) Graph Matrix Key Generate

Graph matrix key generate dari algoritma RSA terlihat pada tabel 4.10

Tabel 4.10 *Graph Matrix Key Generate*

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	**sum**
1	1																		
2		1	1																1
3	1																		
4				1	1														1
5			1																
6						1	1												1
7								1											
8								1											
9									1										
10										1	1								1
11												1							
12												1							
13													1						
14														1	1				1
15													1						
16																1	1		1
17																		1	
18																		1	
Total																			6

V(G) = Jumlah *Graph Matric* + 1

V(G) = 6 + 1

V(G) = 7

Berdasarkan pengujian yang dilakukan pada setiap metode, dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 7, Maka dapat disimpulkan bahwa pengujian *white box* pada proses *key generate* algoritma RSA berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.1.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian *alpha* dengan menggunakan metode *white box* dan *black box* dengan kasus uji diatas maka dapat disimpulkan bahwa Aplikasi Keamanan SMS ini tidak memiliki kesalahan proses dan secara fungsional pun menghasilkan aplikasi yang sudah sesuai dengan perancangan dan kebutuhan pengguna.

4.2.2 Pengujian Beta

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif yang mana melakukan pengujian secara langsung ke lapangan, dengan tujuan untuk mengetahui sejauh mana kualitas dari sistem. Pengujian *beta* dilakukan dengan penelitian terhadap responden atau calon pengguna sistem untuk mengetahui penilaian pengguna terhadap aplikasi. Teknik pengumpulan data dilakukan melalui pengisian kuesioner oleh pengguna yang terdiri dari 30 mahasiswa di lingkuan kampus Un4ersitas Komputer Indonesia.

Untuk megetahui tanggapan dan penilaian pengguna terhadap Aplikasi Keamanan SMS menggunakan Algoritma RSA berbasis android, telah disebar kuesioner kepada responden berdasarkan target pengguna, kemudian dilakukan penghitungan dengan menggunakan skala kontinum dimana data dianalisis dengan menghitung rata-rata jawaban berdasarkan skoring setiap jawaban dari responden yang kemudian berdasarkan rata-rata jawaban dari responden lalu kemudian dijumlahkan.

4.2.2.1 Data Kuesioner

Adapun data kuesioner dari aplikasi kemanan sms menggunakan algoritma RSA adalah sebagai berikut:

1. Jumlah responden yang terdiri dari 30 orang mahasiswa Un4ersitas Komputer Indonesia
2. *Point* yang menjadi bahan penilaian adalah:
   a. Membantu mengamankan pesan singkat.
   b. Tingkat keamanan yang baik
   c. Memenuhi kebutuhan yang diharapkan

Berikut adalah daftar pernyataan kepada masing-masing pengguna pada tabel 4.11.

Tabel 4.11 Daftar Pertanyaan Kuesioner

Pengguna	Pernyataan
Mahasiswa	Pesan singkat (sms) yang sifatnya rahasia harus dijaga keamanannya.
	Aplikasi Keamanan SMS membantu menjaga kerahasiaan pesan singkat.
	Aplikasi Keamanan SMS dengan baik mengenkripsi dan mendekripsi pesan.
	Aplikasi Keamanan SMS memiliki tingkat keamanan yang baik.
	Key Management di dalam aplikasi berjalan dengan baik.
	Fitur *password* di dalam aplikasi menjaga kemanan pesan di dalam ponsel.
	Aplikasi Keamanan SMS mudah digunakan.

4.2.2.2 Pengolahan Data Kuesioner

Adapun skala jawaban dari kuesioner pada tabel 4.12.

Tabel 4.12 Skala Jawaban Kuisioner

Jawaban	Skor
SS = Sangat Setuju	5
S = Setuju	4
R = Ragu-ragu	3
KS = Kurang Setuju	2
TS = Tidak Setuju	1

Berikut adalah hasil pengumpulan data dari masing-masing jawaban untuk setiap pengguna. Jawaban dari masing-masing responden dapat dilihat pada tabel 4.13.

Tabel 4.13 Jawaban Kuisioner Mahasiswa

No	Pernyataan	Jawaban					Total
		SS	S	R	KS	TS	
1	Pesan singkat (sms) yang sifatnya rahasia harus dijaga keamanannya.	15	13	2			**30**
2	Aplikasi Keamanan SMS membantu menjaga kerahasiaan pesan singkat.	20	7	3			**30**
3	Aplikasi Keamanan SMS dengan baik mengenkripsi dan mendekripsi pesan.	15	10	5			**30**
4	Aplikasi Keamanan SMS memiliki tingkat keamanan yang baik.	17	11	2			**30**
5	*Key Management* di dalam aplikasi berjalan dengan baik	12	14	4			**30**
6	Fitur *password* di dalam aplikasi menjaga kemanan pesan di dalam ponsel.	20	8	2			**30**
7	Aplikasi Keamanan SMS mudah digunakan.	22	8				**30**

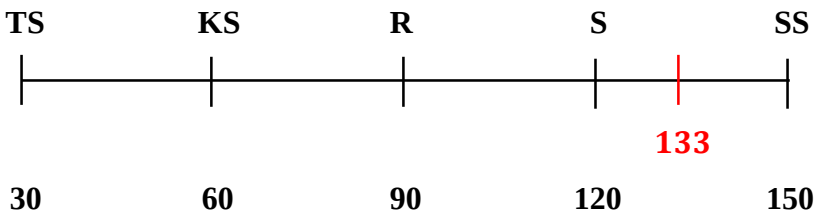
4.2.2.3 Hasil Pengolahan Kuisioner

Berikut merupakan data pengolahan kuesioner seperti pada tabel 4.14.

Tabel 4.14 Hasil Pengolahan Data Kuesioner Pertanyaan (1)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Responden)
SS = Sangat Setuju	5	15	75
S = Setuju	4	13	52
R = Ragu-ragu	3	2	6
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	133

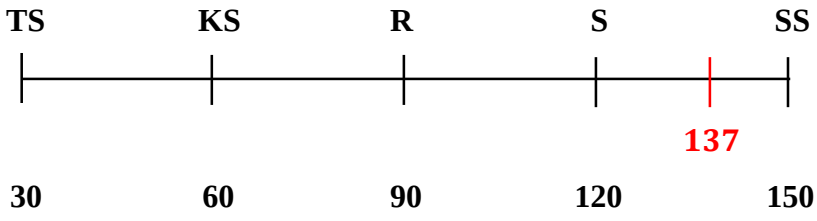
Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 133. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (1) adalah (133 : 150) x 100% = 88,7% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 133 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.15 Hasil Pengolahan Data Kuesioner Pertanyaan (2)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Responden)
SS = Sangat Setuju	5	20	100
S = Setuju	4	7	28
R = Ragu-ragu	3	3	9
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	137

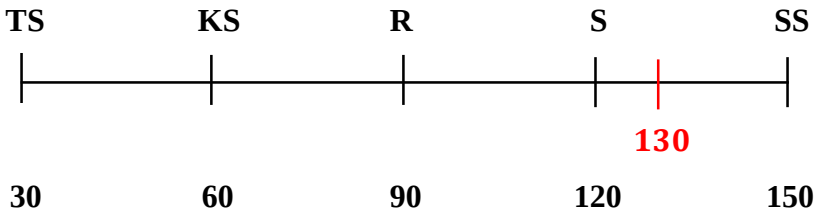
Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 137. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (2) adalah (137 : 150) x 100% = 91.3% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 137 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.16 Hasil Pengolahan Data Kuesioner Pertanyaan (3)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Responden)
SS = Sangat Setuju	5	15	75
S = Setuju	4	10	40
R = Ragu-ragu	3	5	15
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	130

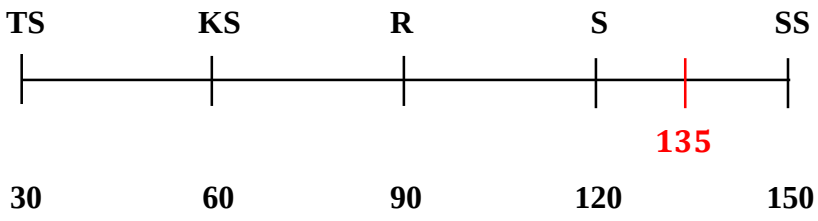
Jika semua responden (37 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 37). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 130. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (3) adalah (130 : 150) x 100% = 86,7% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 130 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.17 Hasil Pengolahan Data Kuesioner Pertanyaan (4)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Responden)
SS = Sangat Setuju	5	17	85
S = Setuju	4	11	44
R = Ragu-ragu	3	2	6
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	135

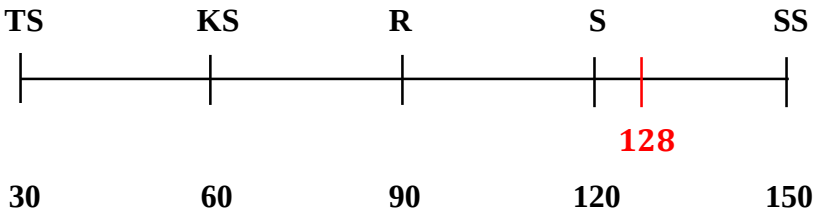
Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 37). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 135. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (4) adalah (135 : 150) x 100% = 90% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 135 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.18 Hasil Pengolahan Data Kuesioner Pertanyaan (5)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Responden)
SS = Sangat Setuju	5	12	60
S = Setuju	4	14	56
R = Ragu-ragu	3	4	12
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	128

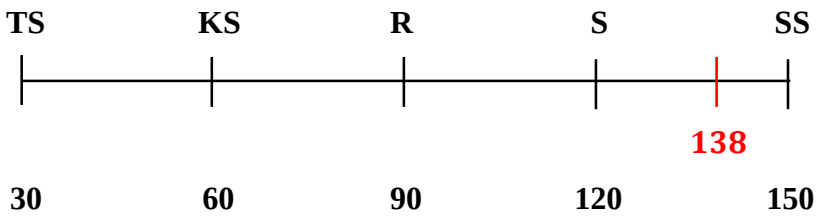
Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 128. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (5) adalah (128 : 150) x 100% = 85.3% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 128 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.19 Hasil Pengolahan Data Kuesioner Pertanyaan (6)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Respoinden)
SS = Sangat Setuju	5	20	100
S = Setuju	4	8	32
R = Ragu-ragu	3	2	6
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	138

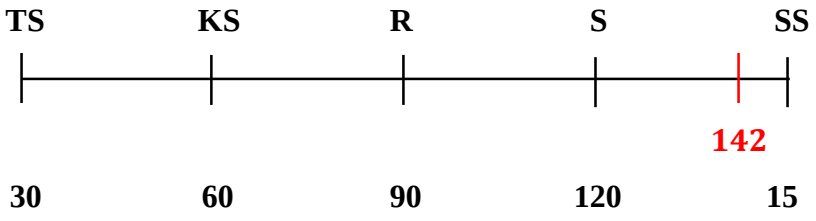
Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 138. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (4) adalah (138 : 150) x 100% = 92% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 150 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



Tabel 4.20 Hasil Pengolahan Data Kuesioner Pertanyaan (7)

Jawaban	Skor	Jumlah Responden	Jawaban Skor (Skor x Jumlah Respoinden)
SS = Sangat Setuju	5	22	110
S = Setuju	4	8	32
R = Ragu-ragu	3	0	0
KS = Kurang Setuju	2	0	0
TS = Tidak Setuju	1	0	0
		Total	142

Jika semua responden (30 orang responden) menjawab SS, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 150 (diperoleh dari 5 x 30). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 142. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor (7) adalah (142 : 150) x 100% = 94,7% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 30 responden, maka rata-rata 142 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut:



4.2.2.4 Kesimpulan Pengujian Beta

Berdasarkan jawaban hasil kuesioner yang diberikan kepada para responden, maka dapat disimpulkan bahwa :

1. Aplikasi keamanan SMS ini dapat menjaga isi pesan singkat yang dikirimkan.
2. Aplikasi keamanan SMS menjaga kerahasiaan pesan singkat.
3. Aplikasi keamanan SMS dengan baik mengekripsi dan mendekripsi pesan singkat.
4. Aplikasi keamanan SMS memiliki tingkat keamanan yang baik.
5. Aplikasi keamanan SMS sesuai dengan kebutuhan pengguna.