BAB III

ANALISA DAN DESAIN SISTEM

III.1. Analisis Masalah

SMS merupakan suatu layanan yang memungkinkan pengguna telepon genggam untuk mengirim pesan singkat kepada pengguna telepon genggam lainnya dengan cepat dan hanya memakan biaya yang sedikit. *SMS* memiliki banyak celah yang memungkinkan para pencuri atau perusak informasi untuk mengambilnya. Kelebihan dari *SMS* ini adalah ketika tujuan sedang sibuk, pesan tetap dapat dikirimkan dengan menyimpan pesan tersebut pada SMSC (*Short Message Service Center*) dan akan mengirimkan ketika tujuan sudah tidak sibuk. Namun kelebihan ini juga yang menjadikannya kelemahan, dengan tersimpannya pesan pada SMSC, maka penyerang dapat mendapatkan pesan dengan melakukan penyusupan pada SMSC tersebut.

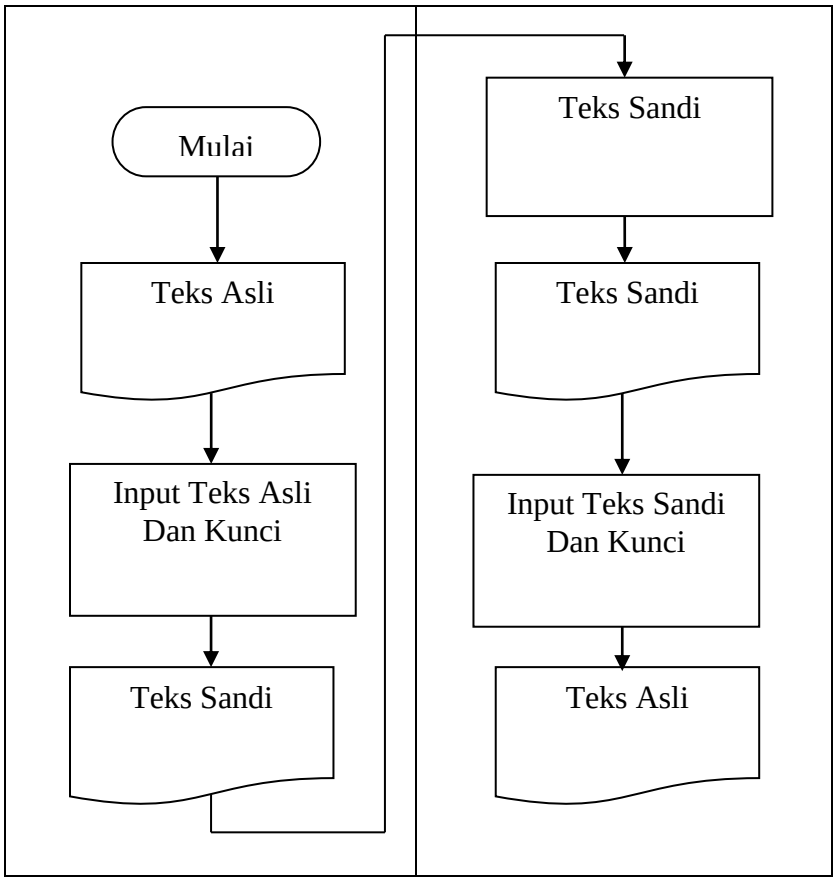
III.1.1. *Input*

Input data pada aplikasi penyandian *SMS* menggunakan metode gronsfeld dan metode vigenere berbasis android adalah berupa teks *SMS*.

III.1.2. Proses

Berdasarkan sistem yang sedang berjalan, tahapan-tahapan proses penyandian teks *SMS* menggunakan metode *gronsfeld cipher* dan *vigenere cipher* adalah sebagai berikut :

Enkripsi	Dekripsi



Gambar III.1. Proses Enkripsi Dan Dekripsi Pesan

III.1.3. Output

Hasil *output* pada aplikasi penyandian *SMS* menggunakan metode gronsfeld dan metode vigenere berbasis android adalah berupa isi teks *SMS* yang tersandikan.

III.2. Evaluasi sistem yang berjalan

Berdasarkan analisa terhadap *input*, proses dan *output* pada aplikasi penyandian sms menggunakan metode gronsfeld dan metode vigenere berbasis android yang sedang berjalan, penulis menemukan beberapa kelemahan antara lain sebagai berikut :

1. Diperlukan sebuah keamanan isi *SMS*.
2. Diperlukan sebuah metode untuk penyandian *SMS*.
3. Diperlukan sebuah perangkat lunak untuk menyandikan data *SMS.*

Untuk menangani kelemahan-kelemahan sistem yang ada salah satu solusi yang ditawarkan adalah dengan merancang aplikasi penyandian *SMS* menggunakan metode *gronsfeld* dan metode *vigenere* berbasis *android*.

III.3. Penerapan Metode

Penerapan metode *gronsfeld* yang dipadukan dengan metode *vigenere* terdapat pada contoh kasus berikut :

Algoritma enkripsi *vigenere cipher* yang dipadukan dengan *gronsfeld cipher*: (F. Wiwiek Nurwiyati dan Indra Yatini B, 2013).

$Cv_i = (Pi + Ki) \bmod 256$

$Cg_i = (Cv_i – (Ki + 48) \bmod 256)$

Contoh Proses Enkripsi :

Plaintext : abc

Kunci : wxy

Solusi :

Ascii Plaintext :

a = 97

b = 98

c = 99

Key :

w = 119

x = 120

y = 121

$Cv_1 = (P1 + k1) \bmod 256$

= (97 + 119) mod 256

= 216 mod 256

= 216

Cg_1 = (Cv_1 – (K1 + 48)) mod 256

= (216 – (119+48)) mod 256

= (216 – 167) mod 256

= 49 mod 256

= 49

Cv_2 = (B + k2) mod 256

= (98 + 120) mod 256

= 218 mod 256

= 218

Cg_2 = (Cv_2 – (K2 + 48)) mod 256

= (218 – (120+48)) mod 256

= (218 – 168) mod 256

= 50 mod 256

= 50

Cv_3 = (C + k3) mod 256

= (99 + 121) mod 256

= 220 mod 256

= 220

Cg_3 = (Cv_3 – (K3 + 48)) mod 256

= (220 – (121+48)) mod 256

= (220 – 169) mod 256

= 51 mod 256

= 51

Chipertext : 123

Algoritma dekripsi *vigenere cipher* yang dipadukan dengan *gronsfeld cipher*: (Aznuddin, 2013).

Pg_i = (Ci + (Ki + 48) mod 256)

Pv_i = (Pg_i - Ki) mod 256

Contoh Proses Dekripsi :

Ciphertext : 123

Kunci : wxy

Solusi :

Ascii Ciphertext :

1 = 49

2 = 50

3 = 51

Key :

w = 119

x = 120

y = 121

Pg_1 = (C1 + (K1 + 48) mod 256)

= (1 + (w+48) mod 256)

= (49 + (119+48)) mod 256

= 216 mod 256

= 216

Pv_1 = (Pg_1 – K1) mod 256

= (216 – 119) mod 256

= 97 mod 256

= 97

Pg_2 = (C2 + (K2 + 48) mod 256)

= (2 + (x+48) mod 256)

= (50 + (120+48)) mod 256

= 218 mod 256

= 218

Pv_2 = (Pg_2 – K2) mod 256

= (218 – 120) mod 256

= 98 mod 256

= 98

Pg_3 = (C3 + (K3 + 48) mod 256)

= (3 + (y+48) mod 256)

= (51 + (121+48)) mod 256

= 220 mod 256

= 220

Pv_3 = (Pg_3 – K3) mod 256

= (220 – 121) mod 256

= 99 mod 256

= 99

Plaintext : abc

III.4. Desain Sistem

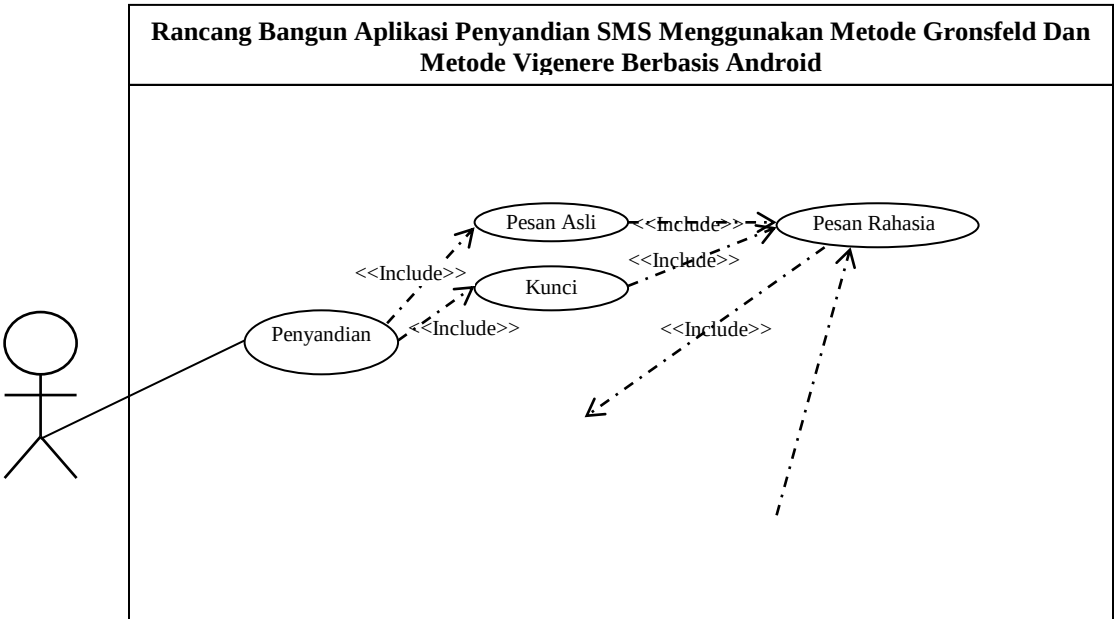
III.4.1. Desain Sistem Secara Global

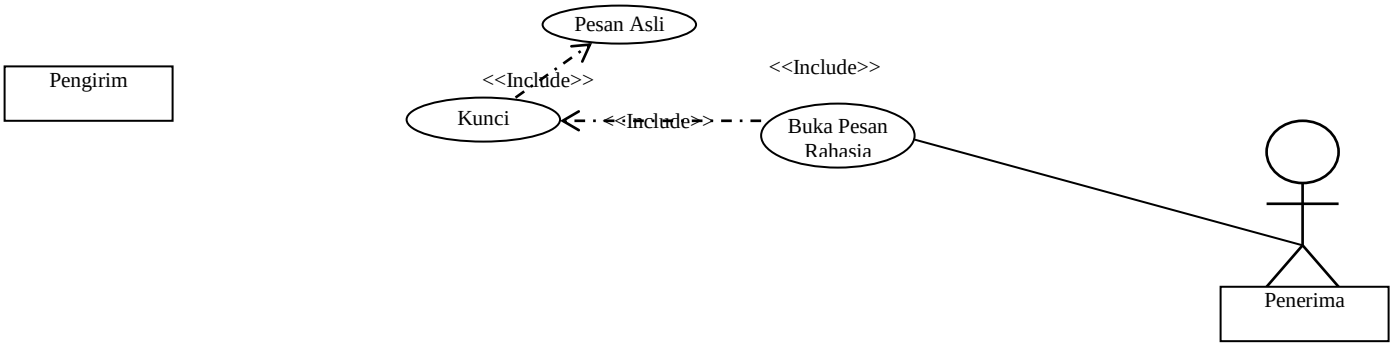
Desain sistem atau perancangan sistem adalah proses pengembangan spesifikasi baru berdasarkan hasil rekomendasi analisis sistem. Dalam tahap perancangan, diharuskan merancang spesifikasi yang dibutuhkan.

Bentuk rancangan sistem yang penulis buat menggunakan beberapa bentuk diagram dari *UML* (*Unified Modeling Language*) yaitu *Use Cace Diagram*, *Class Diagram* dan *Activity Diagram*.

III.4.1.1 *Use Case Diagram*

Perancangan dimulai dari identifikasi aktor dan bagaimana hubungan antara aktor dan use case didalam sistem. Perancangan *Use Case* Diagram dapat dilihat pada gambar III.2.





Gambar III.2. *Use Case* Rancang Bangun Aplikasi Penyandian *SMS* Menggunakan Metode *Gronsfeld* Dan Metode *Vigenere* Berbasis *Android*

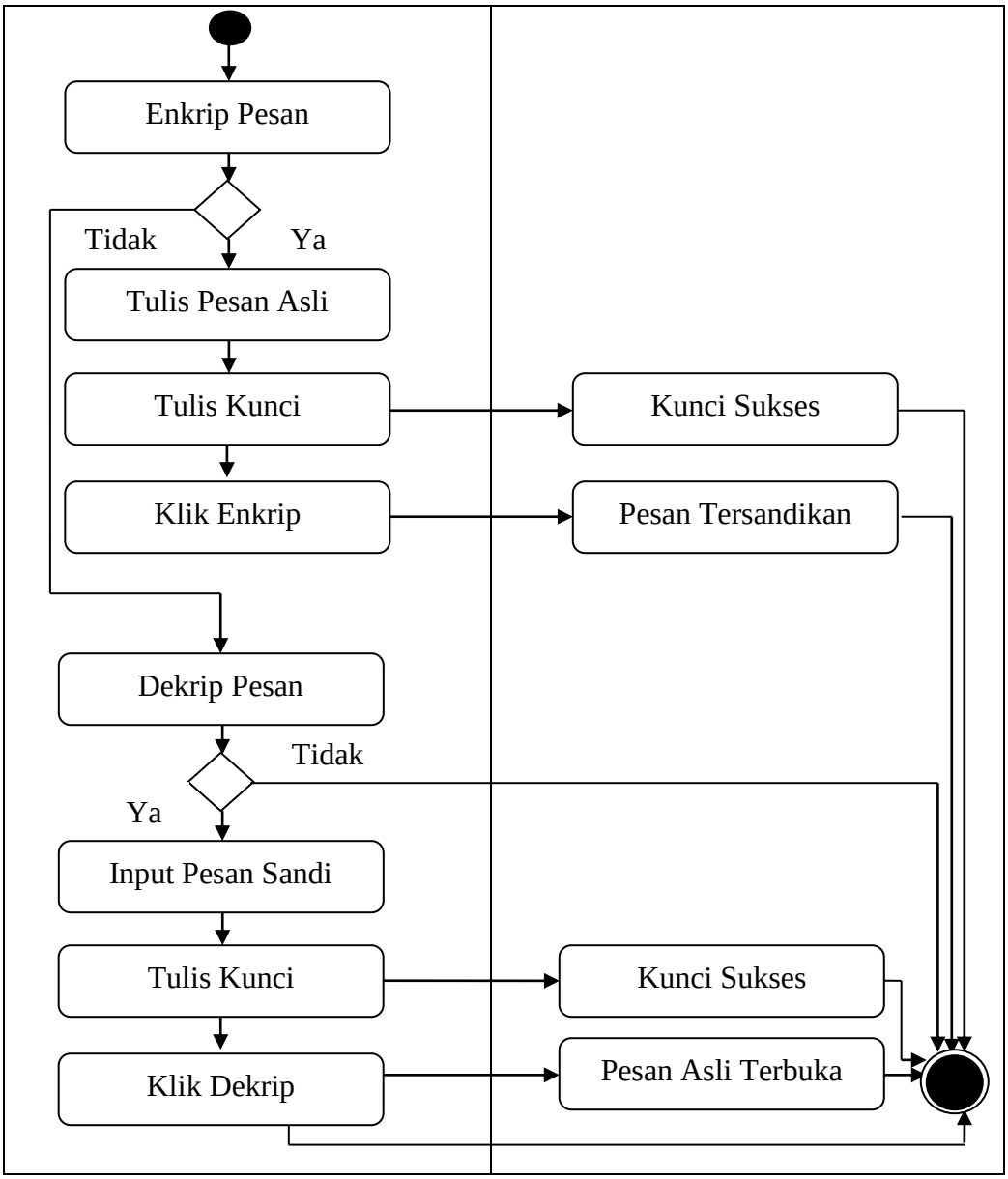
III.4.1.2. *Sequence Diagram*

Rangkaian kegiatan pada setiap terjadi *event* sistem digambarkan pada *sequence* diagram berikut:

1. *Sequence Diagram* Penyandian

Serangkaian kerja melakukan penyandian dapat terlihat seperti pada gambar III.3 berikut :

User	Sistem



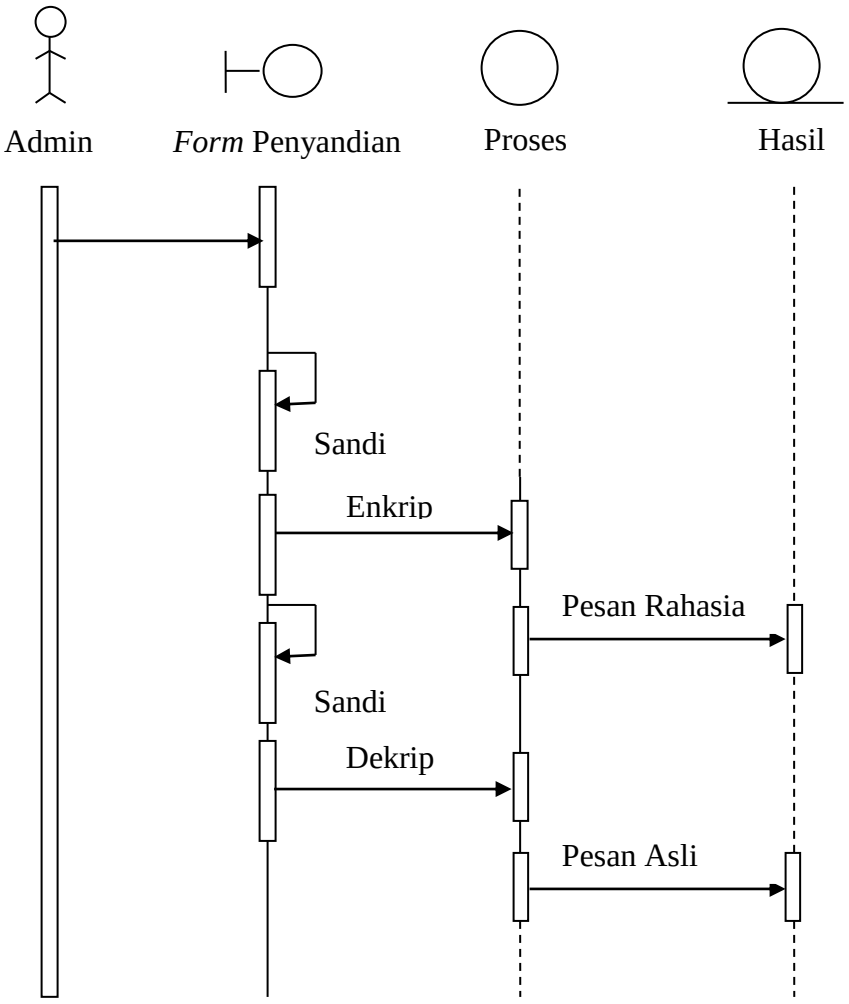
Gambar III.3. *Sequence Diagram* Penyandian

III.4.1.3. *Activity Diagram* Penyandian

Pada proses ini kita akan membuat alur dari system yang dirancang yaitu *activity diagram*. Berikut adalah *activity diagram* sistem yang dirancang.

1. *Activity* Diagram Penyandian

Aktivitas yang dilakukan untuk melakukan penyandian dapat dilihat seperti pada gambar III.4 berikut :



Gambar III.4. *Activity Diagram* Penyandian

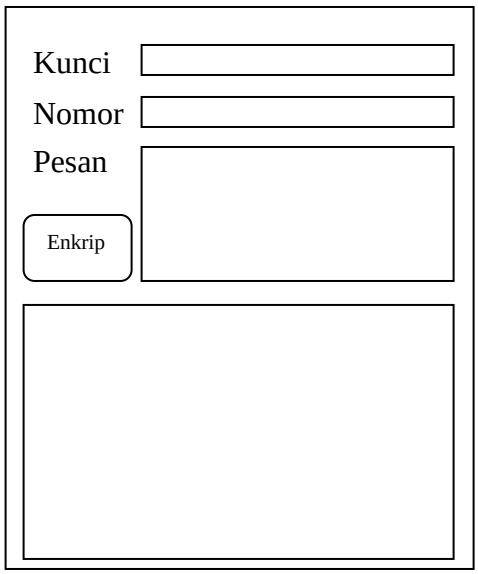
III.4.2. Desain Sistem secara Detail

Perancangan *User Interfase* merupakan masukan yang penulis rancang guna lebih memudahkan dalam *entry* data. *Entry* data yang dirancang akan lebih mudah dan cepat dan meminimalisir kesalahan penulisan dan memudahkan perubahan.

Perancangan tampilan yang dirancang adalah sebagai berikut :

1. Perancangan *Form* Enkrip *SMS*

Perancangan *Form* Enkrip *SMS* berfungsi untuk menyandikan isi teks *sms*. Adapun rancangan *form* Enkrip *SMS* dapat dilihat pada gambar III.5. sebagai berikut :



Gambar III.5. Rancangan *Form* Enkrip *SMS*

Keterangan :

a. Edit text Kunci : Untuk *input* kunci penyandian *SMS*

b. Edit text Nomor : Untuk nomor *hp* penerima *SMS*

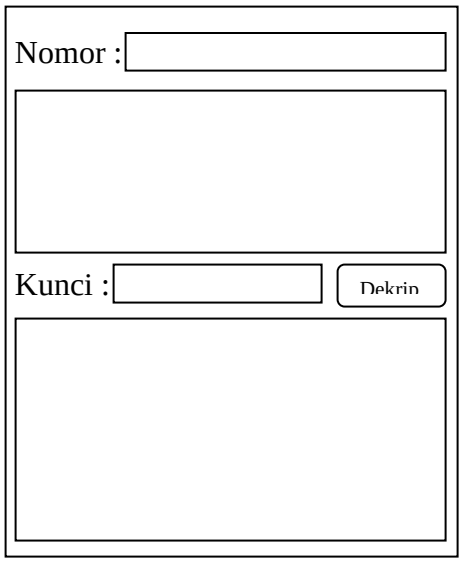
c. Edit text Pesan : Untuk *input text SMS* yang akan disandikan

d. Button Enkrip : Untuk memproses *input text SMS* menjadi *text SMS* tersandi

e. Text view Enkrip : Menampilkan hasil enkripsi *SMS*

2. Perancangan *Form* Dekrip *SMS*

Perancangan *Form* Dekrip *SMS* berfungsi untuk membuka pesan sandi isi teks *sms*. Adapun rancangan *form* Dekrip *SMS* dapat dilihat pada gambar III.6. sebagai berikut :



Gambar III.6. Rancangan *Form* Dekrip *SMS*

Keterangan :

a. Text view Nomor : Nomor *hp* pengirim *SMS*

b. Text view *Sms* : Isi *SMS* pengirim yang tersandikan

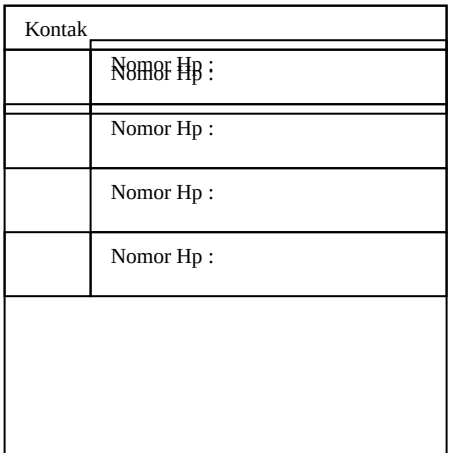
c. Edit text Kunci : Untuk *input* kunci pembuka *SMS* yang tersandikan

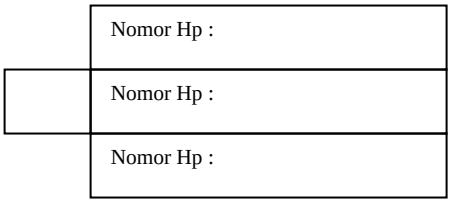
d. Button Enkrip : Untuk memproses isi *SMS* yang diterima menjadi *text* asli

e. Text view Dekrip : Menampilkan hasil dekripsi *SMS*

3. Perancangan *Form* Kontak

Perancangan *Form* Kontak berfungsi untuk membuka pesan sandi isi teks *sms*. Adapun rancangan *form* Kontak dapat dilihat pada gambar III.7. sebagai berikut :





Gambar III.7. Rancangan *Form* Kontak

Keterangan :

Text view Nomor Hp : Untuk Menampilkan isi kontak pada perangkat *android.*