BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Dalam penelitian ini, metodologi penelitian yang dilakukan adalah sebagai berikut.

1. Studi literatur

   Dalam proses ini dipelajari teori-teori yang berhubungan dengan aplikasi yang akan dibangun, seperti pembuatan aplikasi dengan berbasis *android*, buku-buku dan jurnal yang membahas tentang algoritma *ROT13* dan *RC6*.

2. Perancangan Aplikasi

   Pada tahap ini, dirancang aplikasi yang dilakukan dengan pembuatan *Flowchart Diagram*, *Data Flow Diagram*, struktur tabel, dan rancangan antarmuka guna memahami dan mendesain alur kerja dari aplikasi yang akan dibangun, agar proses pembangunan aplikasi dapat berjalan sesuai dengan rencana.

3. Pembuatan Program

   Pada tahap ini dilakukan proses pembuatan program yang berdasarkan tujuan dan kegunaan aplikasi. Pembuatan program meliputi tampilan antarmuka dan *coding* program secara keseluruhan.

4. Testing atau Pengujian

   Proses pengujian aplikasi dengan menggunakan *smartphone* dengan sistem operasi *Android* Lenovo Vibe Shot, Sony C3, Samsung Galaxy Mega 6.3. Aplikasi yang telah dibangun akan dilakukan pengujian atau *testing* dengan

menggunakan *black-box testing* untuk menguji enkripsi dan dekripsi (apakah aplikasi dapat melakukan enkripsi dan juga dekripsi terhadap pesan), dan pengujian pengiriman dan penerimaan pesan(apakah aplikasi dapat mengirim dan menerima pesan). Setelah itu akan dibuat kuesioner untuk mengetahui apakah aplikasi berhasil dalam pengamanan teks dalam mengirim ataupun menerima pesan.

5. Evaluasi

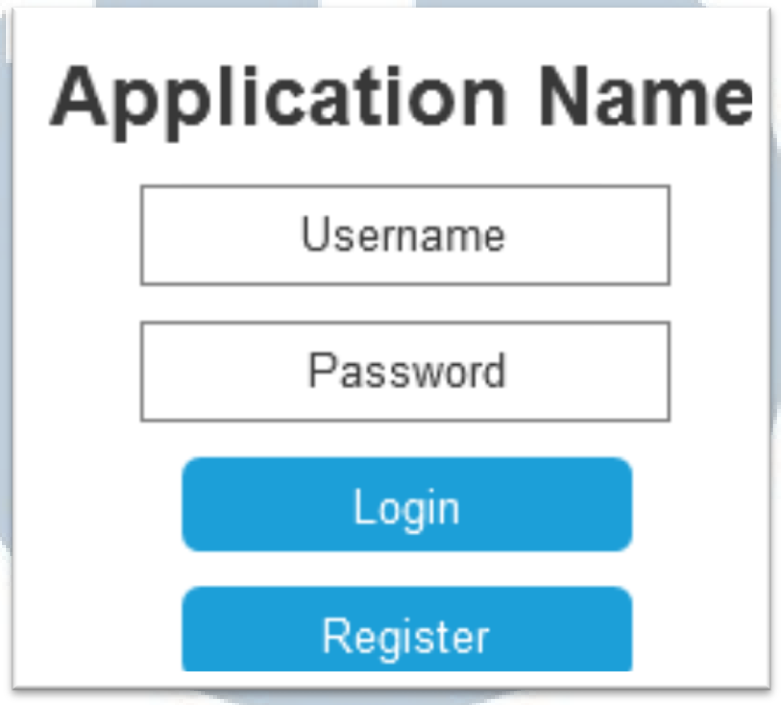
Evaluasi dilakukan berdasarkan hasil dari kuesioner yang telah dibagikan. Hasil evaluasi diperoleh berdasarkan faktor *user satifaction, dan user experience* pengguna menggunakan skala Likert sebagai metode penyimpulan aplikasi yang digunakan untuk mengetahui persentase keberhasilan aplikasi dalam evaluasi penilaian navigasi dan juga apakah aplikasi termasuk *user friendly*. Kemudian, seberapa besar minat orang-orang dalam membutuhkan pengamanan dalam mengirim pesan dan juga apakah benar aplikasi ini dapat bermanfaat dalam pengamanan pesan.

3.2 Perancangan Aplikasi

Perancangan aplikasi yang dibuat dalam penelitian ini adalah rancangan antarmuka aplikasi yang menggambarkan secara kasar tampilan aplikasi yang akan dibangun, *Flowchart diagram* yang menggambarkan alur proses aplikasi, *Data Flow Diagram* yang menggambarkan aliran data dari suatu proses ke proses lain dalam aplikasi dan struktur tabel.

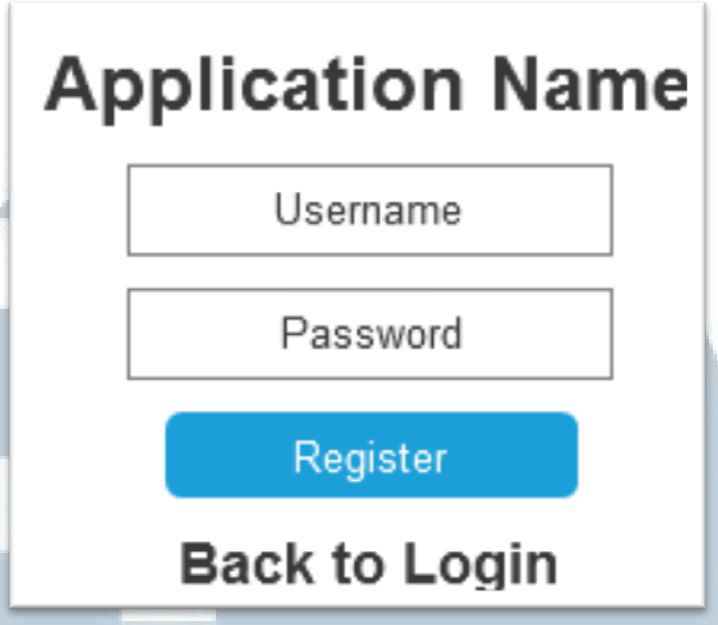
3.2.1 Rancangan Antarmuka

Berikut ini merupakan gambaran umum tentang rancangan dari aplikasi:



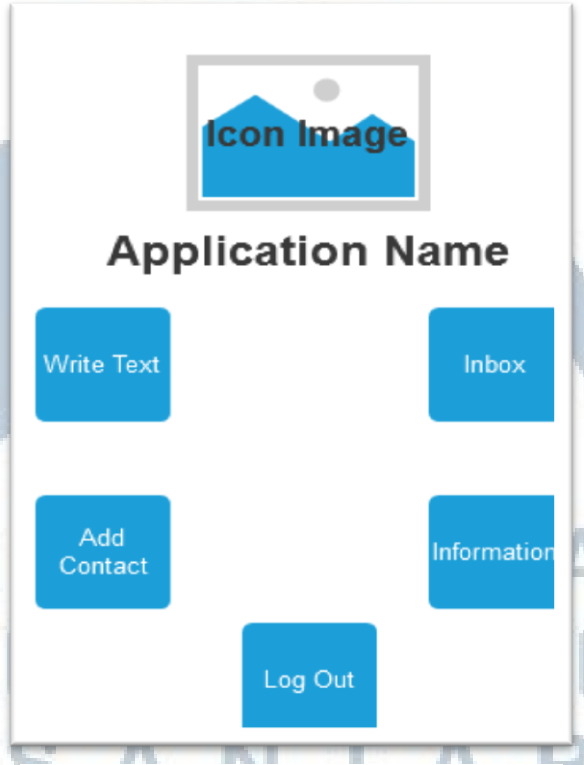
Gambar 3.1 Menu Awal

Pada awal mula aplikasi dijalankan, pengguna akan berada pada menu *login* dan terdapat dua *button* yaitu *Login* dan *Register*, terdapat juga dua buah *Edit Text* yang digunakan untuk menginputkan *username* dan *password*. Jika akun sudah terdaftar maka ketika memilih tombol *login* akan langsung masuk kepada menu utama, sebaliknya ketika masih belum terdaftar maka harus mendaftarkan akun terlebih dahulu dengan memilih tombol *register*.



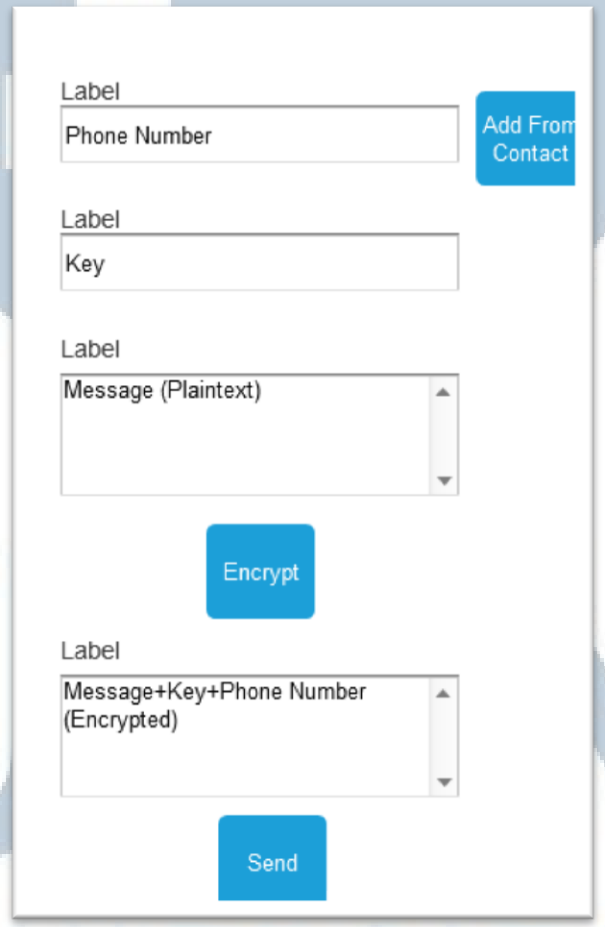
Gambar 3.2 Menu Register

Pada menu *register*, pengguna akan memasukan *username* dan juga *password* yang ingin digunakan, kemudian setelah itu memilih tombol *register* untuk menyimpannya didalam *database user*. *Button link* digunakan untuk kembali ke menu awal yaitu menu *login.*



Gambar 3.3 Menu Utama

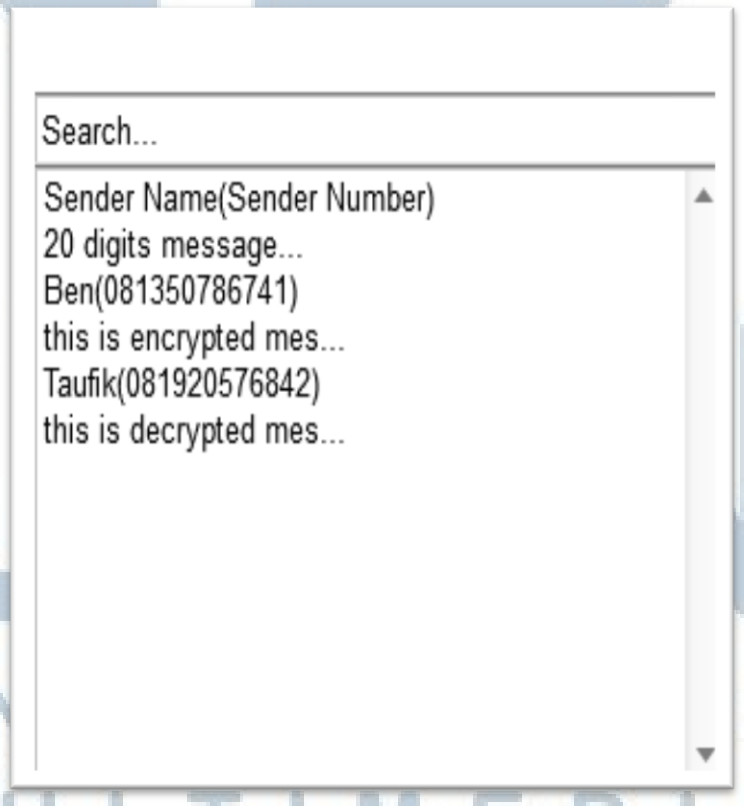
Pada Gambar 3.3 terdapat beberapa pilihan menu seperti ketika pengguna ingin mengirim pesan, melihat *inbox* atau kotak pesan yang masuk, kemudian juga dapat menambahkan nomor telepon baru, menu info aplikasi yang berisi tentang informasi dari aplikasi, dan terakhir yaitu tombol *logout* yang ketika dipilih akan mengeluarkan pengguna dari menu utama dan kembali kepada menu awal yaitu menu *login*.



Gambar 3.4 Menu Kirim Pesan

Pada menu kirim pesan, pengguna akan memasukan nomor telepon tujuan, mengisikan kata kunci dengan minimal 8 digit karakter dan maksimal 16 digit, kemudian mengisikan pesan atau teks yang ingin dikirimkan pada kolom *text area*

Message. Setelah itu ketika menekan tombol *encrypt* maka kunci akan di subtitusi menggunakan ROT13, kemudian setelah itu baru dilakukan proses enkripsi menggunakan algoritma RC6. Hasil akhir dari pesan yang sudah dienkripsi tersebut akan muncul pada kolom hasil pesan yang mana pesan tersebut sudah terenkripsi dan berbentuk heksadesimal yang tidak dapat terbaca seperti pesan awal. Setelah itu aplikasi akan melakukan proses enkripsi terhadapa *key* dan juga nomor telepon penerima dan hasil dari enkripsi tersebut akan dijadikan satu bersama pesan yang sudah lebih dulu dienkripsi. Kemudian langkah terakhir yaitu mengirimkan pesan tersebut kepada nomer tujuan melalui tombol *Send*.



Gambar 3.5 Menu Inbox

Pada Gambar 3.5, pengguna akan memilih pesan mana yang ingin didekripsi pada *listview*, yang mana dalam *listview* tersebut ditampilkan nama

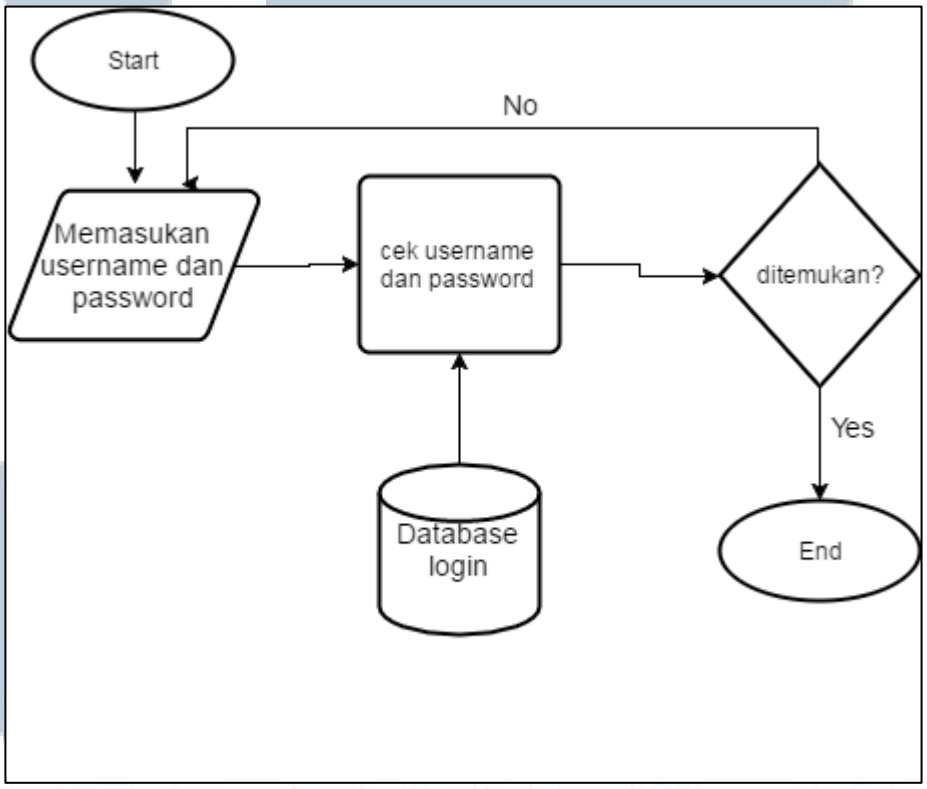
kontak pengirim pesan beserta nomor teleponnya, dan juga 20 digit pertama dari pesan yang ingin didekripsi. Pada menu *inbox* ini juga dapat melakukan pencarian pesan pada *search box* yang telah disediakan.

Gambar 3.6 Menu Baca Pesan

Pada menu baca pesan, pengguna akan mendapatkan nomor telepon pengirim dan pesan berupa teks heksadesimal yang belum terbaca pada *edit text encrypted message*. Langkah pertama sebelum pesan dapat terbaca yaitu aplikasi akan membandingkan apakah nomor telepon yang dikirim sama dengan nomor telepon penerima, jika sama maka kunci atau *key* akan otomatis di *generate* dan ditampilkan pada *edit text key*, jika tidak sama maka pesan tidak akan bisa terbaca.

Setelah melakukan proses verifikasi nomor telepon, maka dan didapatkan hasil yaitu kunci untuk membuka pesan. Kemudian kunci tersebut akan di subtitusi menggunakan ROT13 dan setelah mendapatkan kunci yang asli, maka proses selanjutnya yaitu mendekripsi pesan menggunakan algoritma RC6 dengan menggunakan kunci yang sama, hasil dari dekripsi pesan tersebut akan ditampilkan pada *edit text* pesan (*plaintext*).

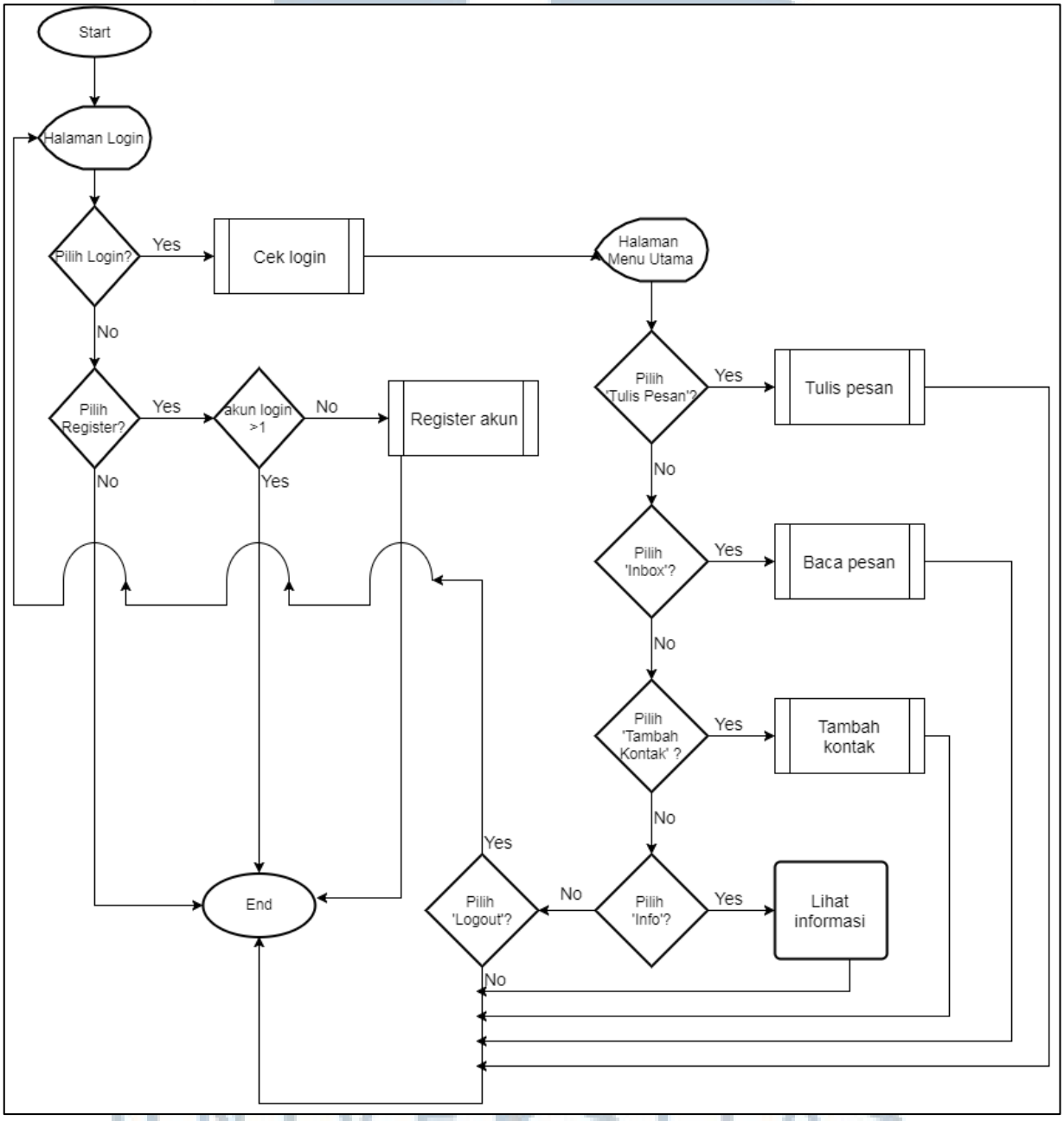
3.2.2 Flowchart



Gambar 3.7 Flowchart login

Pada Gambar 3.5 yaitu pada saat mulai pertama kali aplikasi dijalankan, maka *user* harus *login* dulu untuk bisa masuk ke menu utama. Kemudian akan dilakukan pengecekan ke *database login* dimana ketika *user* dan *password*

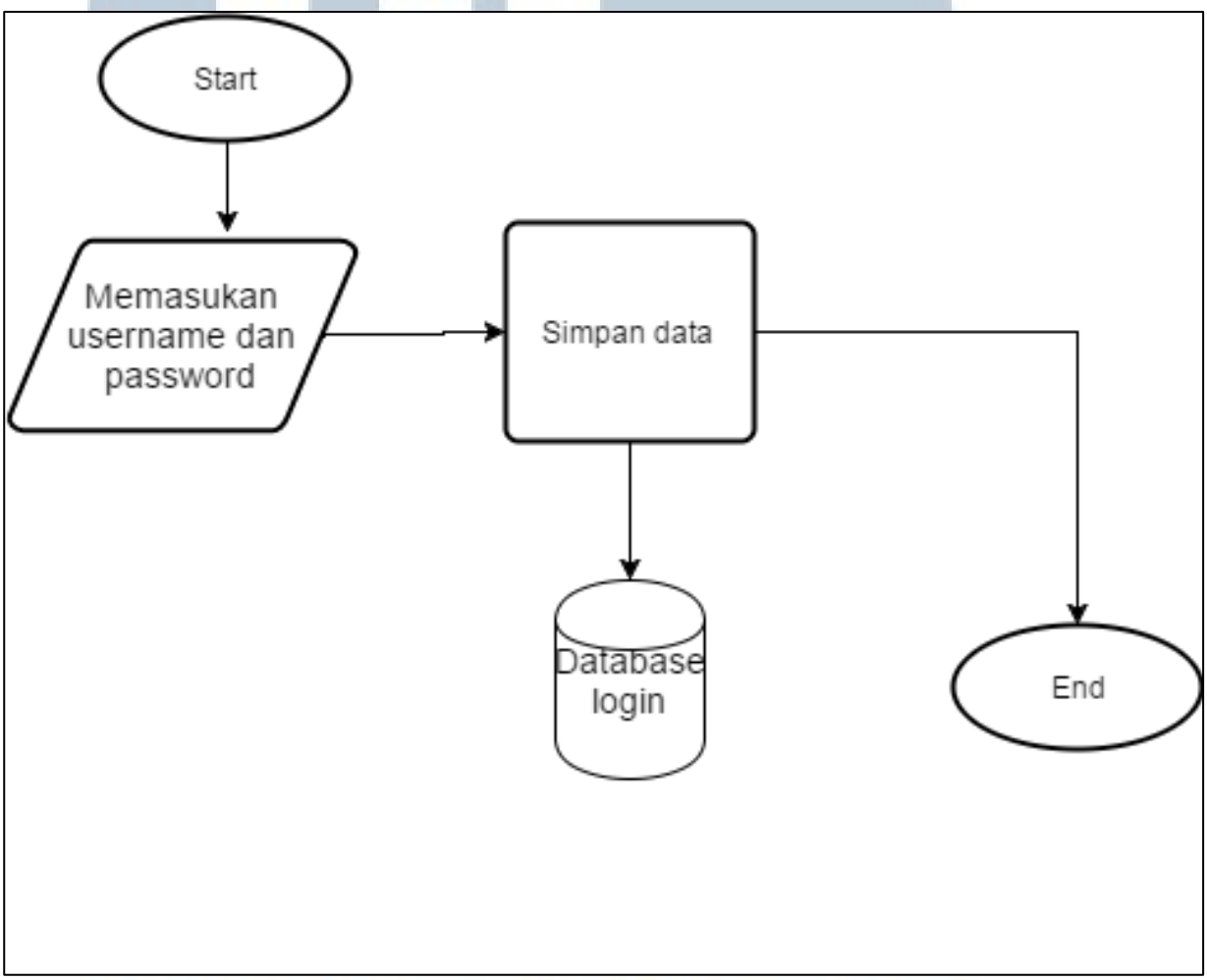
ditemukan, maka *login* akan berhasil dan apabila tidak ditemukan, maka *user* diharuskan memasukan ulang *user* dan *pass* yang terdaftar dalam *database login*.



Gambar 3.8 Flowchart Menu Utama

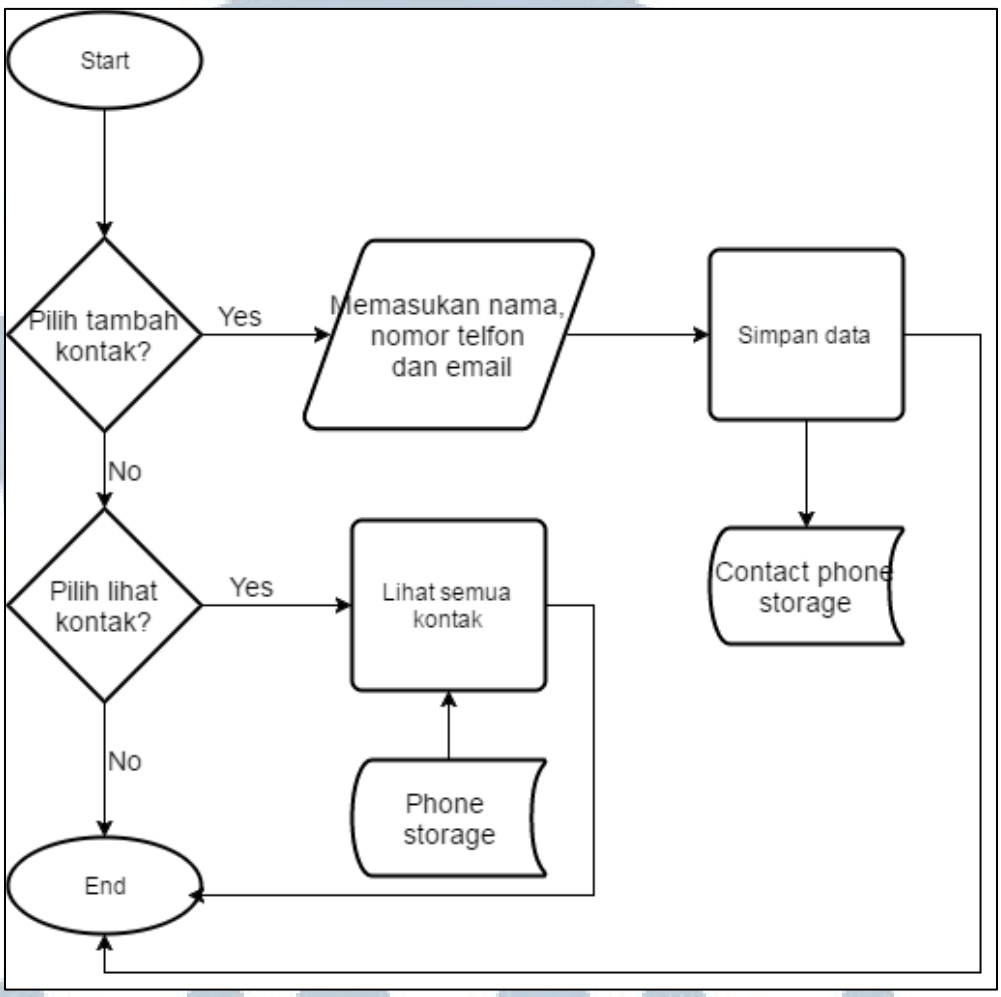
Pada Gambar dijelaskan secara rinci jalannya keseluruhan aplikasi. Mulai dari *login*, jika tidak terdapat *user* maka terlebih dahulu harus *register* dan memilih tombol *register*. Pada menu *register* dilakukan pembuatan akun untuk

login, jika sudah ada akun yang dibuat sebelumnya maka *user* tidak akan dapat membuat akun baru, jika masih kosong maka akun akan tersimpan dan dapat digunakan untuk *login* ke menu utama. Setelah masuk ke halaman utama, *user* dapat menulis pesan dengan memilih tombol tulis pesan, dapat juga membaca *inbox*, menambahkan kontak baru, dan juga melihat informasi tentang aplikasi.



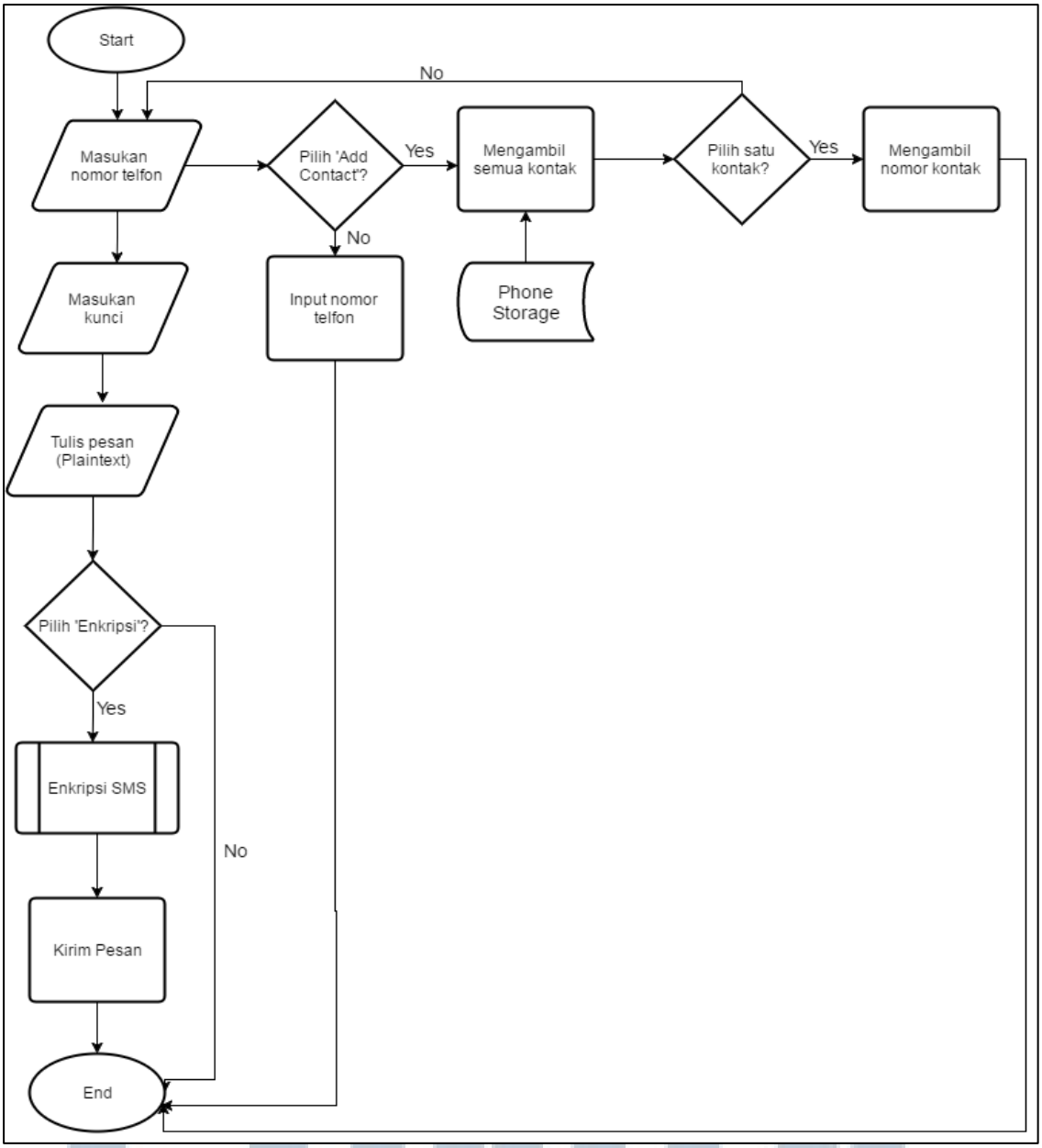
Gambar 3.9 Flowchart Register

Pada Gambar 3.9 *user* proses yang terjadi pada *register* yaitu ketika inputan berupa *username* dan *password* dimasukan, maka akan disimpan ke dalam *database login*.



Gambar 3.10 Flowchart Tambah Kontak

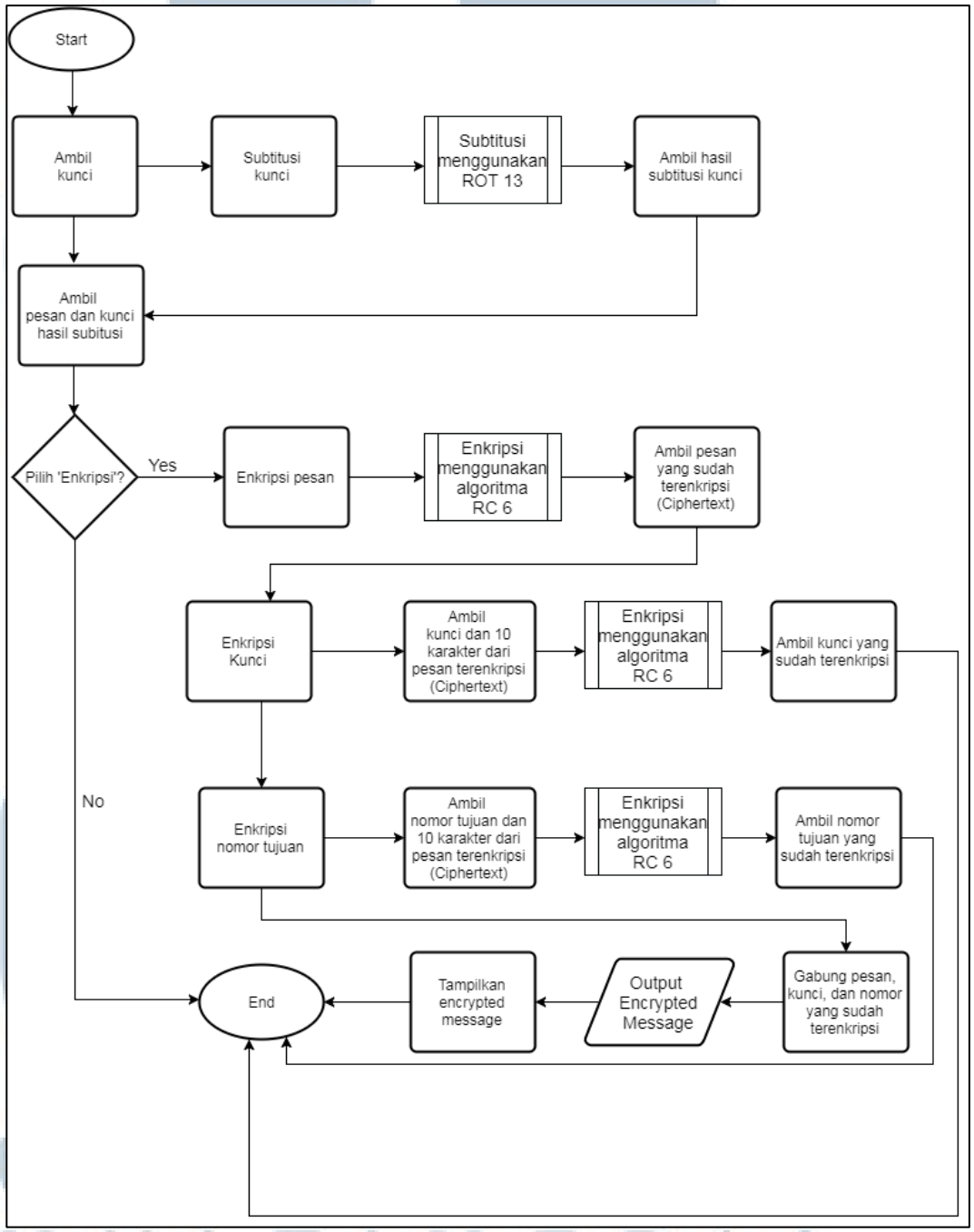
Pada Gambar 3.10 *flowchart* tambah kontak berfungsi untuk menyimpan kontak baru seseorang, yaitu dengan memasukan masing-masing nama, nomor telepon dan alamat *email* dan sistem akan menyimpan data tersebut langsung pada tempat penyimpan *handphone*. Jika tidak ingin menambah kontak baru, disini juga dapat melihat kontak yang sudah disimpan dengan memilih lihat kontak. *List* kontak juga akan langsung diambil dari kontak yang sudah tersimpan pada *handphone*.



Gambar 3.11 Flowchart Tulis Pesan

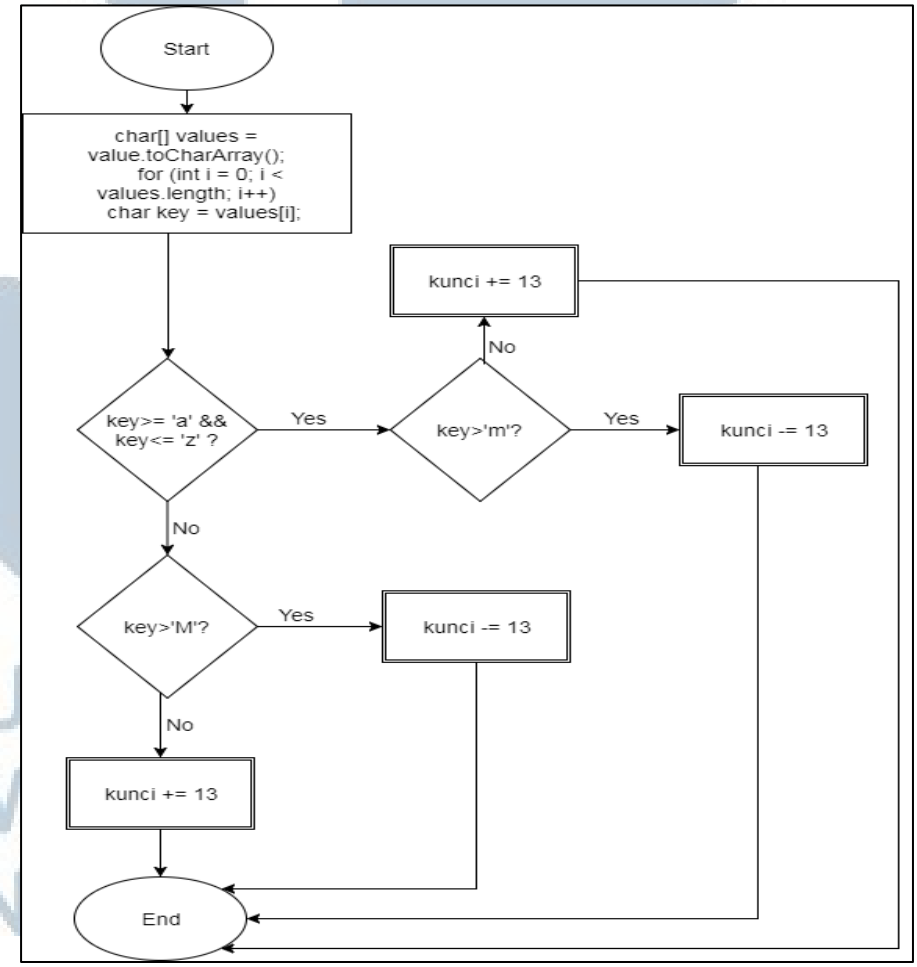
Pada Gambar 3.11 dijelaskan tentang gambaran umum ketika ingin menulis pesan, yaitu pertama-tama harus memasukan nomor telepon tujuan, bisa dengan menginputkan manual atau langsung mengambil lewat kontak yang sudah tersimpan pada penyimpanan *handphone*, kemudian memasukan *key* yang ingin digunakan dan juga pesan yang ingin dikirim itu sendiri. Pesan tidak akan bisa

dikirim jika tidak dienkripsi terlebih dahulu, oleh karena itu *user* harus memilih tombol enkripsi dan setelah pesan terenkripsi barulah pesan dapat dikirim.



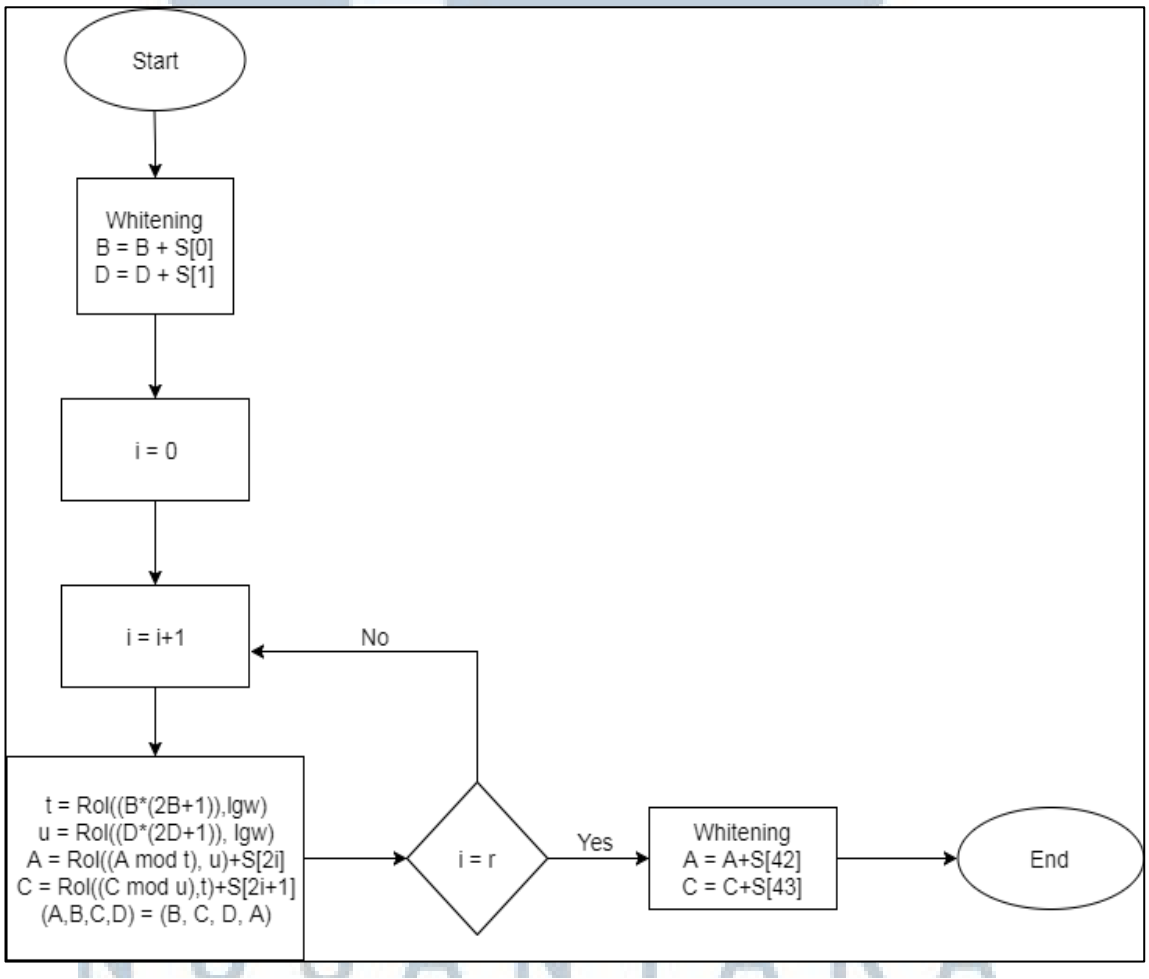
Gambar 3.12 Flowchart Enkripsi Pesan

Pada Gambar 3.12 dijelaskan bagaimana sebuah pesan dapat dienkripsi. Pertama-tama kunci atau *key* akan dirotasi menggunakan ROT13 yaitu dengan cara subtitusi kemudian hasil dari subtitusi kunci itu digunakan untuk mengenkripsi pesan dengan menggunakan algoritma enkripsi RC6. Kemudian setelah itu hasil dari pesan yang sudah terenkripsi (*ciphertext*) itu kemudian akan diambil 10 karakter pertamanya yang kemudian digunakan untuk mengenkripsi nomor tujuan dan juga kunci atau *key* yang kemudian akan digabungkan menjadi satu dan dikirimkan bersamaan. Jadi hasil akhir pesan yang dikirimkan terdiri dari gabungan pesan terenkripsi (*ciphertext*), nomor tujuan terenkripsi (*ciphertext*), dan juga *key* atau kunci (*ciphertext*).



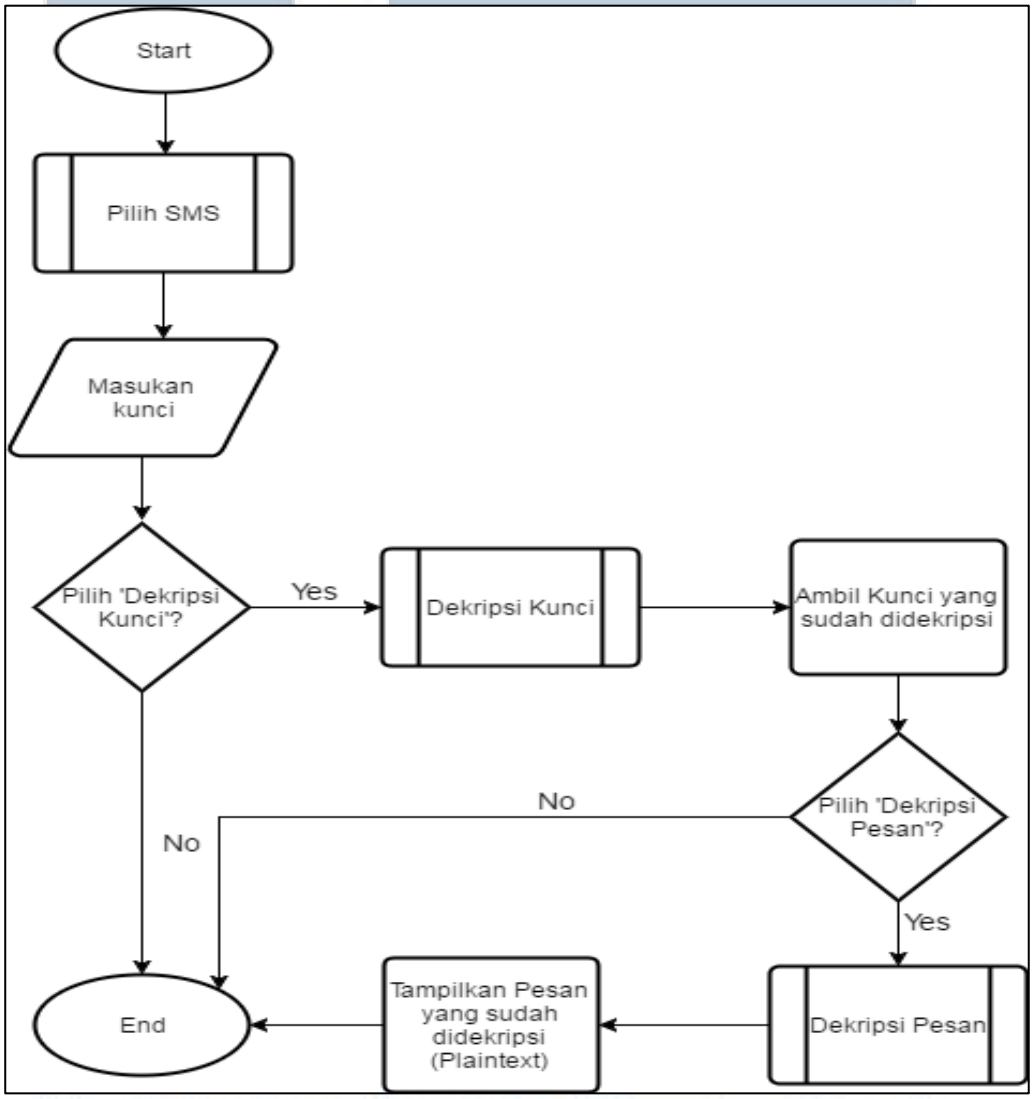
Gambar 3.13 Flowchart ROT13

Pada Gambar 3.13 merupakan proses bagaimana ROT13 bekerja yaitu pertama-tama *key* atau kunci dijadikan char yang akan ditampung pada sebuah *array* dan dilakukan perulangan sebanyak panjang kunci. Kemudian char *key* akan dipisahkan bagian mana yang *lowercase* atau *uppercase*, jika char *key* tersebut masuk ke dalam *lowercase* maka akan dipilih lagi jika *key* > 'm', maka *key*-=13, dan jika sebaliknya *key* < 'm' maka *key* akan menjadi *key*+=13. Sebagai contoh jika *char key* yang didapat adalah 'a' maka akan masuk dalam kondisi *key* < 'm', maka otomatis *char* 'a' tersebut akan ditambah 13 dan menjadi 'n'. Sama halnya jika *key* berupa *uppercase* maka akan gantinya akan berupa *uppercase* juga.



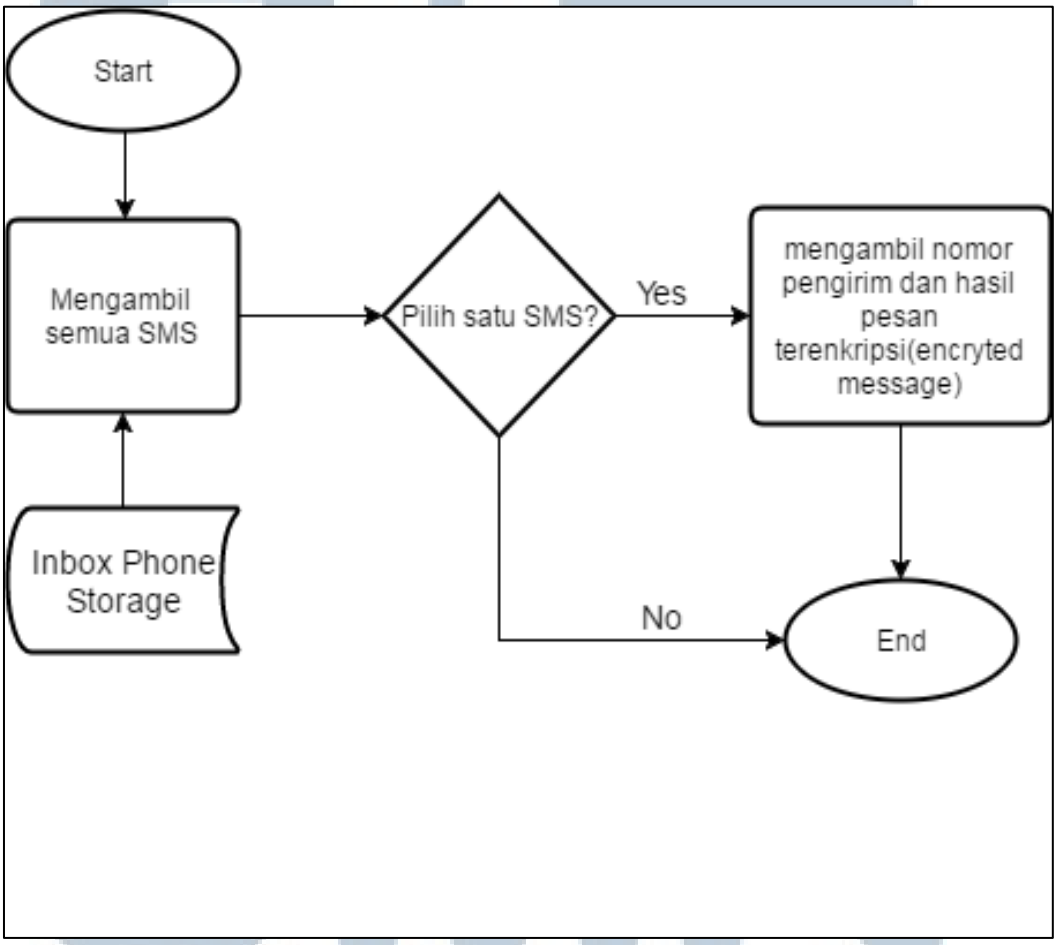
Gambar 3.14 Flowchart Enkripsi RC6

Pada Gambar 3.14 proses *whitening* awal, nilai B akan dijumlahkan dengan S[0], dan nilai D dijumlahkan dengan S[i]. Pada masing-masing iterasi pada RC6 menggunakan dua buah sub kunci. Sub kunci pada iterasi yang pertama menggunakan S[2] dan S[3], sedangkan iterasi-iterasi berikutnya menggunakan sub-sub kunci lanjutannya. Setelah iterasi ke-20 selesai, dilakukan proses *whitening* akhir dimana nilai A dijumlahkan dengan S[42], dan nilai C dijumlahkan dengan S[43].



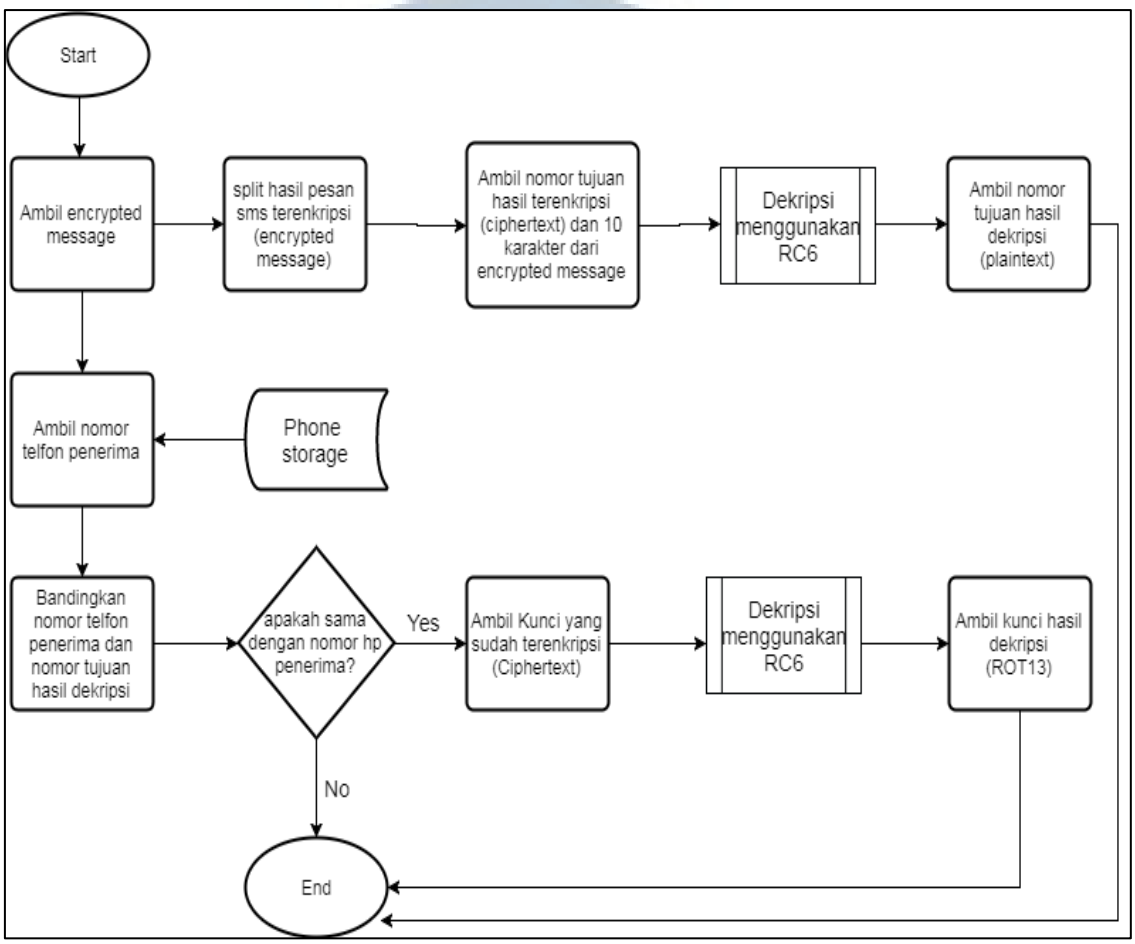
Gambar 3.15 Flowchart Baca Pesan

Pada Gambar 3.15 dijelaskan bagaimana alur ketika ingin membaca pesan yang telah diterima. Gambaran umumnya yaitu *user* memilih pesan mana yang ingin dibaca kemudian sebelum mendekripsi pesan, kunci atau *key* harus didekripsi terlebih dahulu. Setelah kunci telah didekripsi dan didapatkan hasilnya kemudian pesan pun dapat didekripsi menggunakan kunci tadi.



Gambar 3.16 Flowchart Pilih SMS

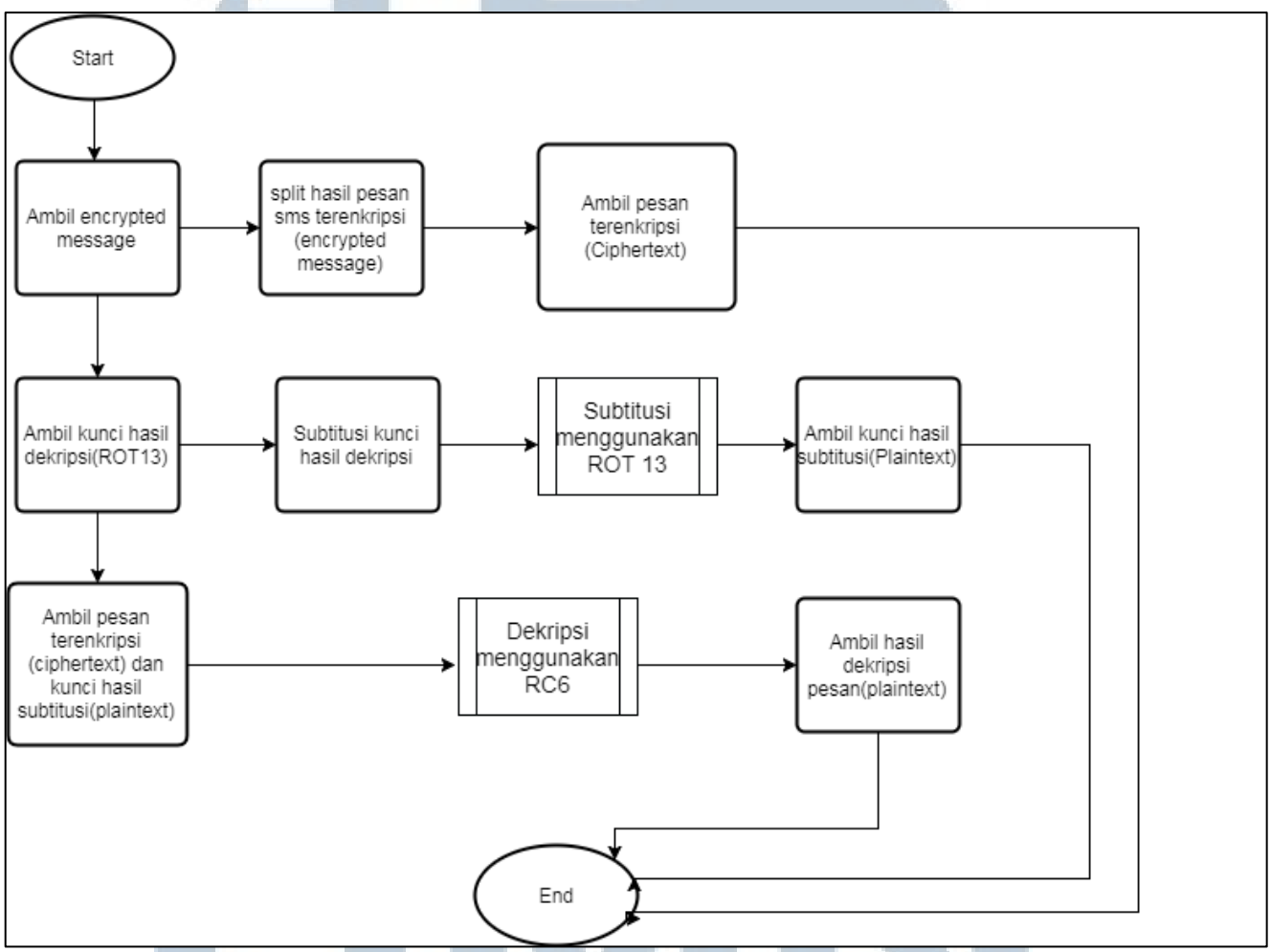
Pada Gambar 3.16 dijelaskan lebih dalam lagi bagaimana proses pemilihan pesan yang ingin dibaca. Pertama-tama sistem akan mengambil pesan yang ada pada *inbox handphone*, kemudian setelah memilih salah satu dari pesan itu kemudian setelah dipilih maka akan diambil nomor pengirim dan juga hasil pesan terenkripsinya.



Gambar 3.17 Flowchart Dekripsi Kunci

Pada Gambar 3.17 merupakan proses lanjutan tentang bagaimana proses dekripsi kunci bekerja. Pertama-tama ambil hasil gabungan pesan enkripsi tersebut kemudian di *split* menggunakan fungsi .subString. Kemudian langkah pertama dekripsi nomor yang tadi sudah dikirim bersamaan pesan menggunakan 10 digit pertama dari pesan terenkripsi, kemudian ketika nomor telepon sudah didekripsi maka hasil dari dekripsi itu akan dibandingkan dengan nomor telepon penerima. Jika tidak cocok maka proses tidak bisa dilanjutkan dan pesan tidak dapat didekripsi. Jika nomor telepon tadi ternyata sama berarti penerima memang benar adalah nomor yang dituju, kemudian langkah selanjutnya pun mendekripsi

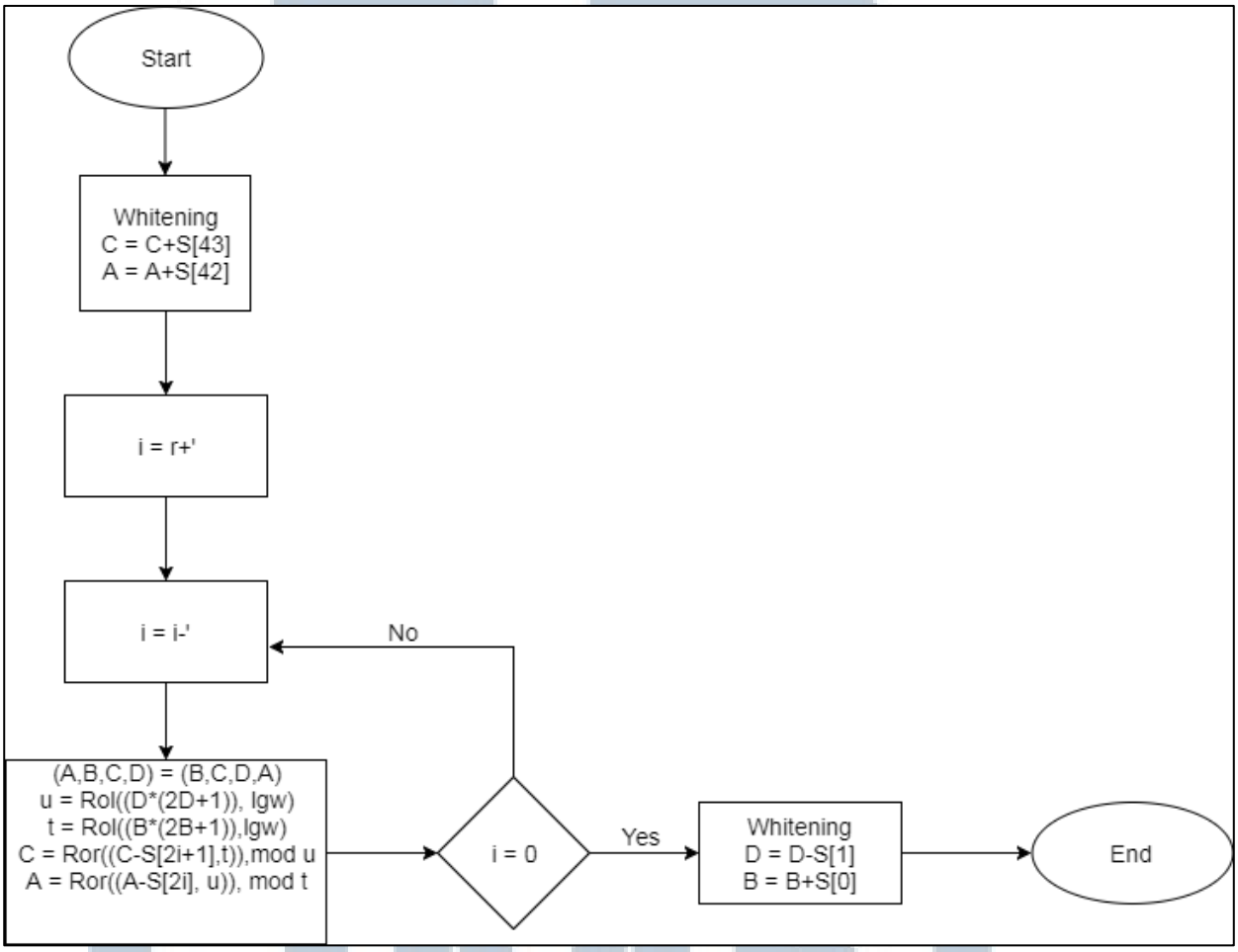
key atau kunci yang tadi juga dikirim dengan menggunakan 10 digit pertama dari pesan terenkripsi. Setelah itu kunci hasil dekripsi pun berhasil didapatkan.



Gambar 3. 18 Flowchart Dekripsi Pesan

Pada Gambar 3.18 dijelaskan proses terakhir untuk mendekripsi pesan yaitu dengan cara kembali memisahkan pesan sehingga hanya isi pesan yang diambil. Kemudian setelah itu kembali melakukan subtitusi kunci yang tadi telah didekripsi menggunakan ROT13 agar *key* atau kunci sama seperti yang pertama kali di-input. Kemudian setelah itu pesan *ciphertext* tadi didekripsi dengan *key* atau kunci yang sudah asli (*plaintext*) menggunakan RC6 dan kemudian pesan

pun berhasil didekripsi dan hasil pesan langsung akan ditampilkan pada kolom *Plain Message*.

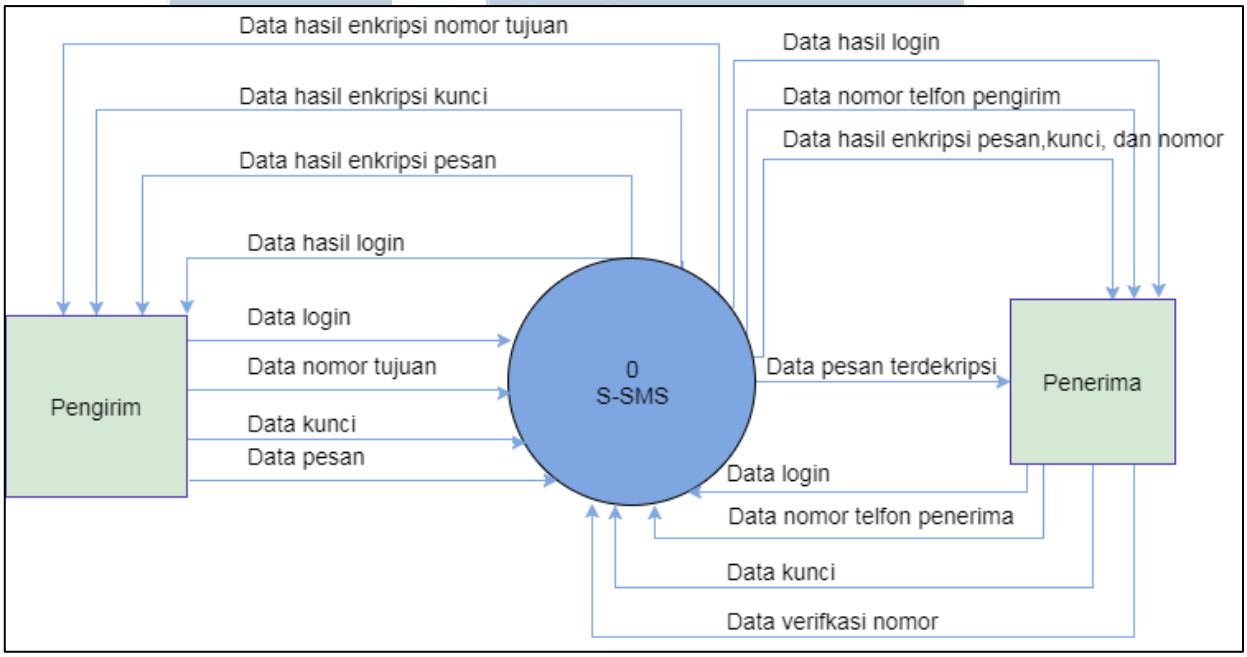


Gambar 3.19 Flowchart Dekripsi RC6

Pada Gambar 3.19 proses dekripsi *ciphertext* pada algoritma RC6 merupakan pembalikan dari proses enkripsi. Pada proses awal, bila proses enkripsi menggunakan operasi penjumlahan, maka pada proses dekripsi menggunakan operasi pengurangan. Sub kunci yang digunakan pada proses akhir setelah iterasi terakhir diterapkan sebelum iterasi pertama, begitu juga sebaliknya sub kunci yang diterapkan pada proses sebelum iterasi pertama digunakan pada iterasi terakhir. Akibatnya, untuk melakukan dekripsi, hal yang harus dilakukan

semata-mata hanyalah menerapkan algoritma yang sama dengan enkripsi, dengan tiap iterasi menggunakan sub kunci yang sama dengan yang digunakan pada saat enkripsi, hanya saja urutan sub kunci yang digunakan terbalik.

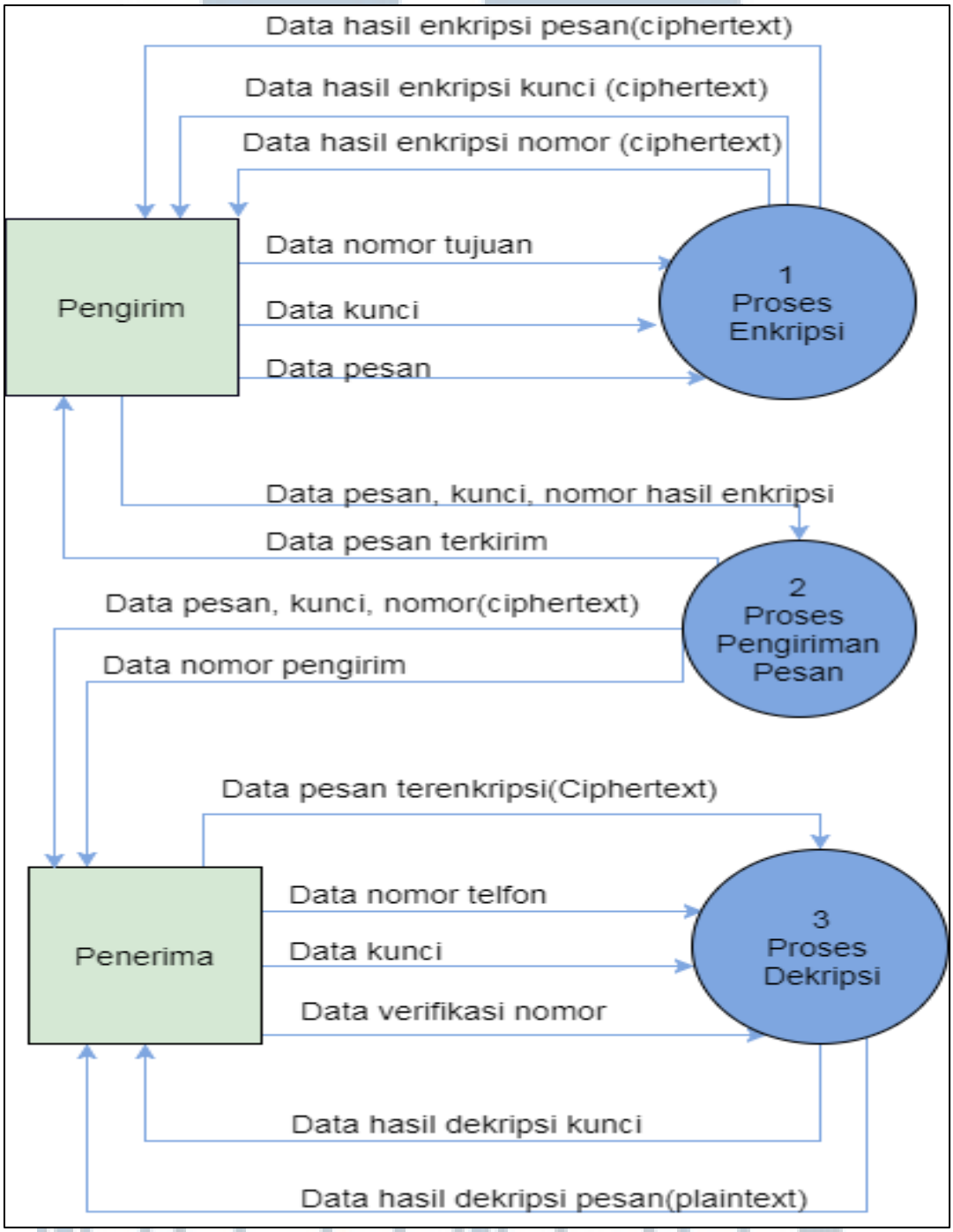
3.2.3 Data Flow Diagram



Gambar 3.20 *Context Diagram*

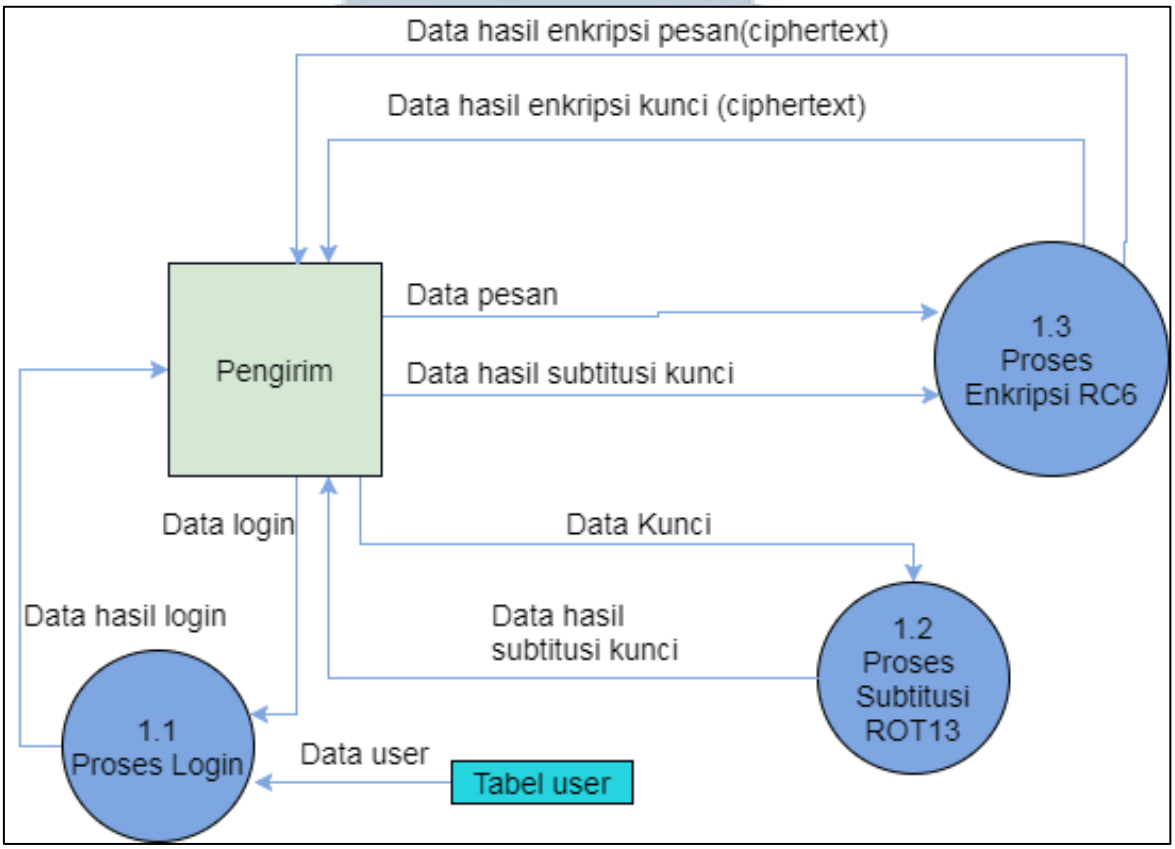
Pada Gambar 3.20 merupakan *context diagram* yang mempunyai dua entitas yang menggambarkan pengguna sistem, yaitu pengirim dan penerima. Entitas pengirim dapat memberikan data *login*, nomor tujuan, kunci, dan juga isi pesan itu sendiri kepada sistem, kemudian sistem akan mengembalikan data hasil *login*. Setelah itu sistem dapat mengembalikan pesan, kunci, dan nomor tujuan yang telah terenkripsi. Pada entitas penerima, entitas ini dapat memberikan data *login*, data nomor telepon penerima dan data kunci kepada sistem, sistem pun akan memberikan nomor telepon pengirim dan juga pesan berupa *ciphertext* yang mana nanti akan bisa dibuka oleh kunci yang dikirim oleh pengirim. Setelah pesan

itu didekripsi maka sistem akan bisa memberikan pesan yang bisa terbaca kepada penerima.



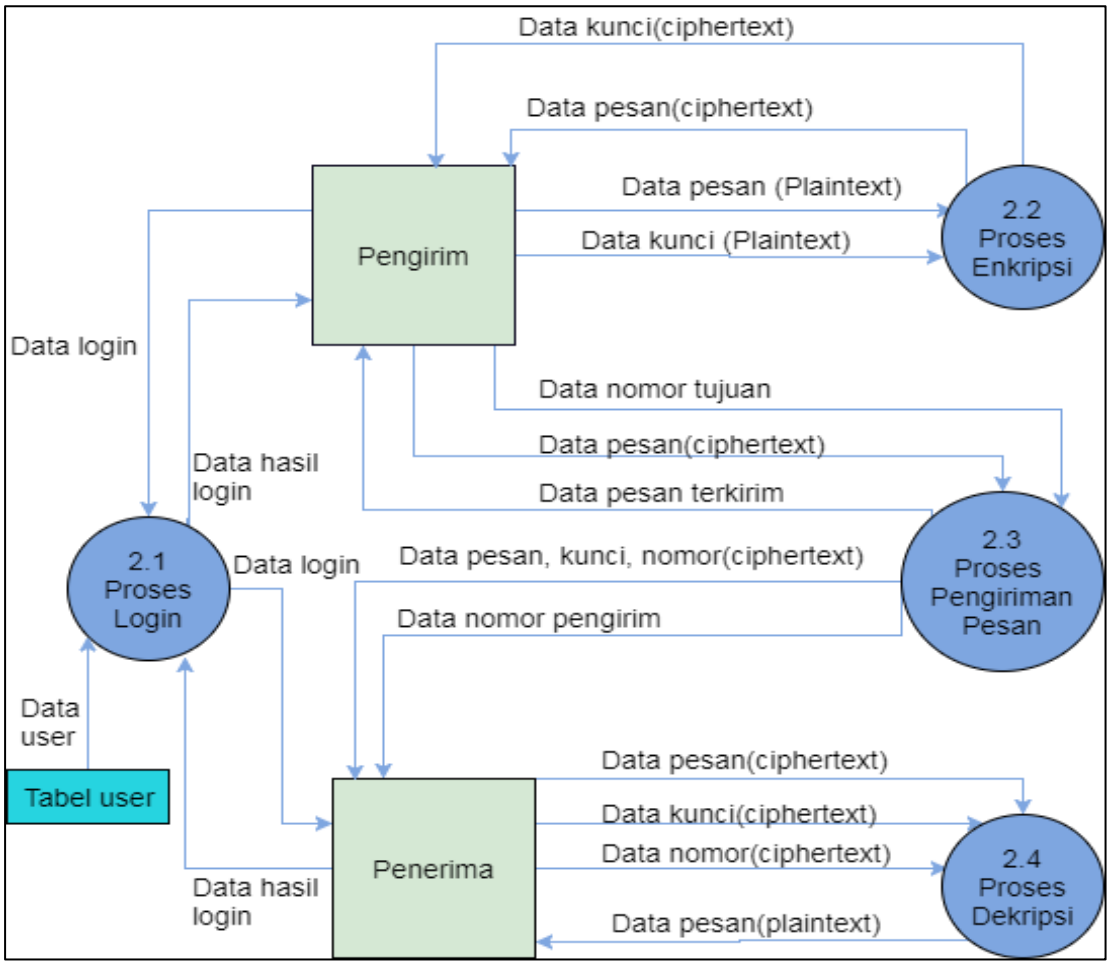
Gambar 3.21 Data Flow Diagram Level 1

Pada Gambar 3.21 dijelaskan bahwa *Data Flow Diagram Level* 1 memiliki tiga buah proses, yaitu proses enkripsi, proses pengiriman pesan dan proses dekripsi. Proses enkripsi merupakan proses yang berinteraksi langsung kepada pengirim, pada proses ini pengirim akan *input* nomor tujuan, kunci atau *key* yang ingin digunakan dan kemudian isi pesan yang ingin dikirim, setelah itu proses enkripsi akan menerima dan mengenkripsi pesan, kunci, dan nomor tersebut menjadi *ciphertext* dan setelah itu akan mengirimkannya kembali kepada pengirim. Proses kedua yang terjadi yaitu saat proses pengiriman pesan. Pesan yang tadi sudah di enkripsi oleh proses enkripsi akan dikirim kepada penerima melalui proses pengiriman pesan, kemudian proses pengiriman pesan akan memberitahu kepada pengirim bahwa pesan telah terkirim. Proses ketiga yang langsung berinteraksi kepada entitas penerima yaitu proses dekripsi, pada proses dekripsi ini, entitas penerima akan diberikan nomor pengirim dan juga hasil pesan terenkripsi, kemudian entitas penerima akan melakukan verifikasi nomor telepon oleh proses dekripsi. Jika nomor telepon yang dikirim oleh pengirim sama dengan nomor telepon yang tersimpan pada hp penerima, maka kunci akan didekripsi oleh proses dekripsi dan dapat digunakan untuk membuka pesan yang telah dikirim oleh pengirim.



Gambar 3.22 *Data Flow Diagram Level* 2 Proses Enkripsi

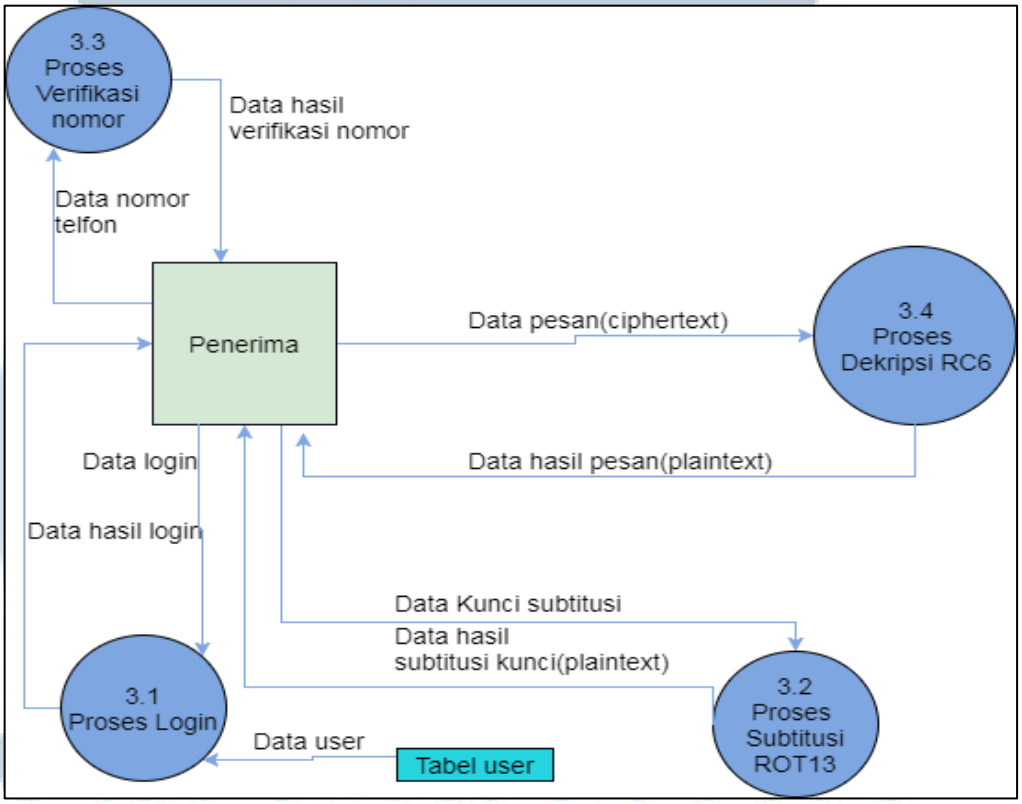
Gambar 3.22 merupakan *data flow diagram level* 2 yang terdiri dari tiga proses, yaitu proses enkripsi RC6, proses *login*, dan juga proses subtitusi ROT13. Pengirim pertama-tama akan memasukan nomor tujuan, kunci, dan juga isi pesan yang kemudian akan diproses oleh proses enkripsi RC6. Sebelum melakukan enkripsi pesan, pengirim terlebih dahulu memberikan kunci untuk diproses oleh proses subtitusi ROT13 agar kunci atau *key* tidak berbentuk statik atau apa adanya. Kemudian setelah selesai melakukan subtitusi, proses subtitusi akan memberikan data hasil subtitusi kunci kepada pengirim, kunci inilah yang nanti akan digunakan untuk mengenkripsi pesan. Tahap akhir proses enkripsi RC6 akan mengembalikan hasil pesan berupa *ciphertext*.



Gambar 3.23 *Data Flow Diagram Level* 2 Proses Pengiriman Pesan

Gambar 3.23 merupakan *data flow diagram level* 2 proses pengirim pesan. yang terdiri dari empat proses, yaitu proses *login*, proses enkripsi, proses pengiriman pesan, dan proses dekripsi. Pada proses pengiriman pesan ini

juga terdapat dua entitas yaitu pengirim dan penerima. Pengirim akan memberikan data pesan dan juga data kunci kepada proses enkripsi untuk dilakukan proses enkripsi, kemudian nomor tujuan dan juga pesan yang sudah terenkripsi tadi diberika kepada proses pengiriman, dan proses pengiriman pesan akan memberi tahu bahwa pesan telah terkirim. Di sisi penerima akan menerima data berupa nomor pengirim dan juga pesan yang diberikan oleh proses pengiriman pesan. Selanjutnya penerima pun akan memberikan data tersebut kepada proses dekripsi untuk diproses. Setelah selesai maka proses dekripsi akan mengembalikan data pesan berupa *plaintext* atau pesan yang dapat terbaca.



Gambar 3.24 *Data Flow Diagram Level* 2 Proses Dekripsi Pesan

Gambar 3.24 *merupakan data flow diagram level* 2 proses dekripsi yang terdiri dari empat proses, yaitu proses verifikasi nomor, proses dekripsi RC6, proses login, dan juga proses subtitusi ROT13. Sebelum penerima dapat membaca atau mendekripsi pesan yang diterima, penerima harus memverifikasi nomor telepon yang sudah dikirimkan oleh pengirim, apakah sama dengan nomor telepon yang tersimpan didalam *handphone* penerima, jika sama maka kunci otomatis akan didekripsi oleh proses dekripsi RC6 dan akan mengembalikan hasil kunci yang masih tersubtitusi. Kemudian disinilah peran dari proses subtitusi ROT12 lagi untuk mengembalikan bentuk kunci kembali ke bentuk asal (*plaintext*) dan memberikan hasil subtitusi kunci tadi kepada penerima. Setelah itu penerima akan memberikan kunci yang sudah berupa *plaintext* tadi untuk diproses oleh proses dekripsi RC6. Setelah proses dekripsi dilakukan dan berhasil, proses dekripisi RC6 akan memberikan pesan yang sudah didekripsi kepada penerima berupa pesan yang bisa dibaca (*plaintext*).

3.2.4 Struktur Tabel

A. Tabel users

Primary key : _id

Tabel 3.1 Struktur Tabel User

Atribut	Tipe	Ukuran	Keterangan
_id	Integer	10	Kode unik *user*
email	Text	-	*Username* untuk *login*
password	Text	-	*Password* untuk *login*