BAB III

METODE PENELITIAN DAN PERANCANGAN SISTEM

3.1. Metode Penelitian

Penelitian ini adalah tentang bagaimana cara mengimplementasikan AES pada sistem BroadcastMe di iOS. Penelitian yang dilakukan penulis ini tentunya menggunakan tahapan penelitian, yaitu:

1) Studi literatur

   Tahap ini merupakan tahap untuk mencari penelitian dan sumber terkait untuk dipelajari. Penelitian atau sumber terkait dapat berupa media cetak seperti jurnal, buku atau dapat juga berupa media elektronik, seperti artikel dari internet.

2) Perancangan dan pengembangan

   Di tahap ini dilakukan tahap analisis untuk melakukan perancangan sistem dan mengembangkan rancangan tersebut.

3) Tahap uji coba

   Tahap uji coba memastikan bahwa sistem yang dibuat pada penelitian kali ini telah berfungsi dan dapat menjalankan tugasnya dengan baik, yaitu mengenkripsi pesan sehingga pesan menjadi lebih aman.

4) Pengumpulan sampel untuk uji coba

   Di tahap ini dilakukan kuesioner terhadap sejumlah responden untuk meminta pendapat mereka tentang sistem yang dibuat pada penelitian ini.

5) Analisis sampel

Pada tahap analisis sampel dilakukan perhitungan terhadap sampel yang berhasil dikumpulkan pada tahap sebelumnya untuk mencari tahu pendapat responden terhadap sistem yang dibuat.

6) Penulisan laporan

Penulisan laporan merupakan tahap dimana seluruh tahap yang telah disebutkan sebelumnya didokumentasikan dan disusun ke dalam bentuk sebuah laporan.

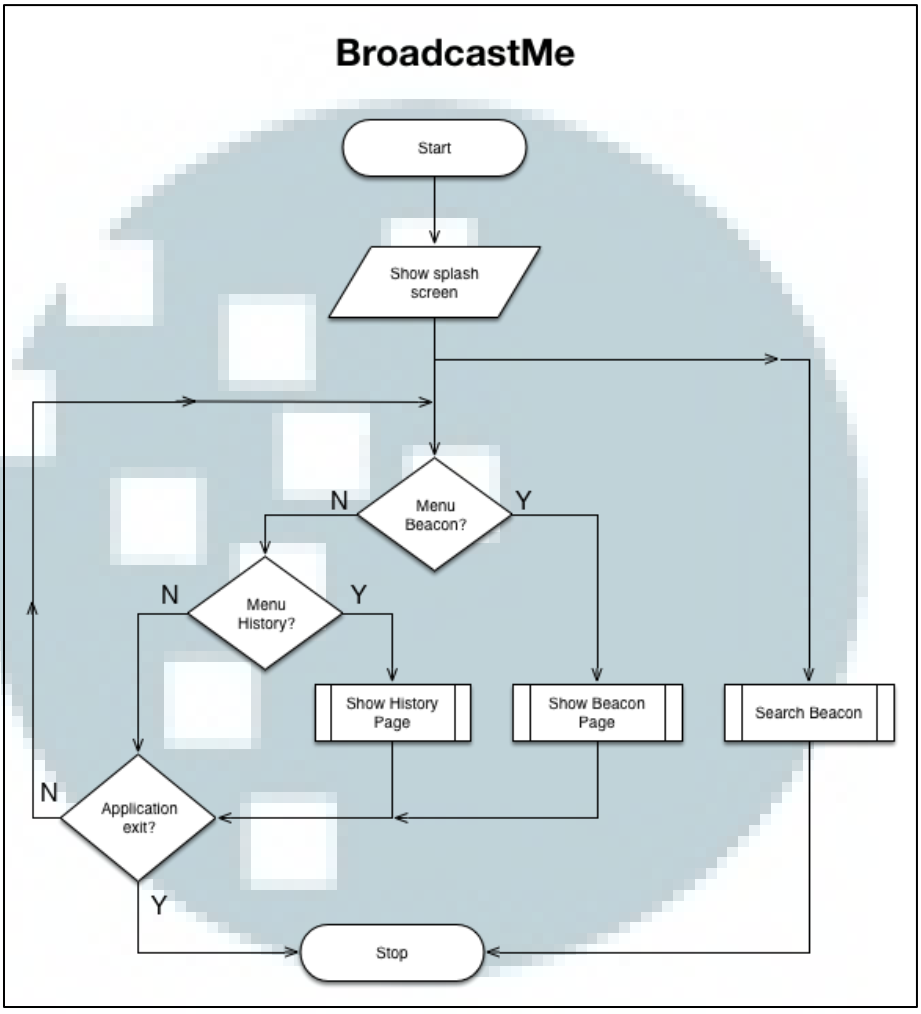
3.2. Perancangan Sistem

3.2.1. BroadcastMe

BroadcastMe adalah sebuah sistem yang memanfaatkan teknologi iBeacon sehingga dapat memberikan informasi berdasarkan lokasi pengguna yang berada dalam jangkauan sinyal iBeacon. Dikarenakan belum terdapat banyak perangkat *beacon* yang beredar, sebuah perangkat iOS dapat disimulasikan sehingga bertindak sebagai pengganti perangkat *beacon*. Fitur untuk melakukan *broadcast* sinyal iBeacon inilah yang dimasukkan ke dalam sistem BroadcastMe.

Sistem BroadcastMe memiliki dua menu utama, yaitu menu *history* dan menu *beacon*. Menu *history* digunakan untuk menampilkan daftar *beacon* serta pesan enkripsi dan hasil dekripsinya serta sebuah angka *major* yang diterima dari *beacon* tersebut, sedangkan menu *beacon* digunakan untuk memasukkan sebuah pesan yang akan di-*broadcast* menggunakan perangkat pengguna. BroadcastMe memiliki sebuah subproses yang berjalan di *background* saat sistem dijalankan,

yaitu subproses Search Beacon. Pengguna dapat menavigasi antara kedua menu dan sistem berhenti apabila sistem ditutup oleh pengguna.



Gambar 3.1. Diagram alir untuk sistem BroadcastMe

3.2.2. Search Beacon

Apabila BroadcastMe menemukan *beacon* yang ada disekitarnya, maka sistem melakukan pengecekan terhadap *proximity* dari beacon tersebut sehingga *beacon* dengan *proximity* yang sama dari sebelumnya tidak membanjiri sistem dan membuat notifikasi yang sama berturut-turut. Jika *beacon* tidak memiliki *proximity* yang sama dengan *beacon* sebelumnya, maka sistem melakukan *request* ke *server* untuk mengambil informasi terkait *beacon* tersebut. Informasi yang

diambil dari *server* berupa informasi terkait *beacon* tersebut, yaitu nama yang diberikan kepada *beacon* tersebut dan pesan terenkripsi yang di-*broadcast* oleh beacon tersebut. Sistem juga memeriksa apakah *beacon* tersebut mengirimkan pesan sama berulang-ulang sehingga tidak membuat notifikasi yang membanjiri pengguna.

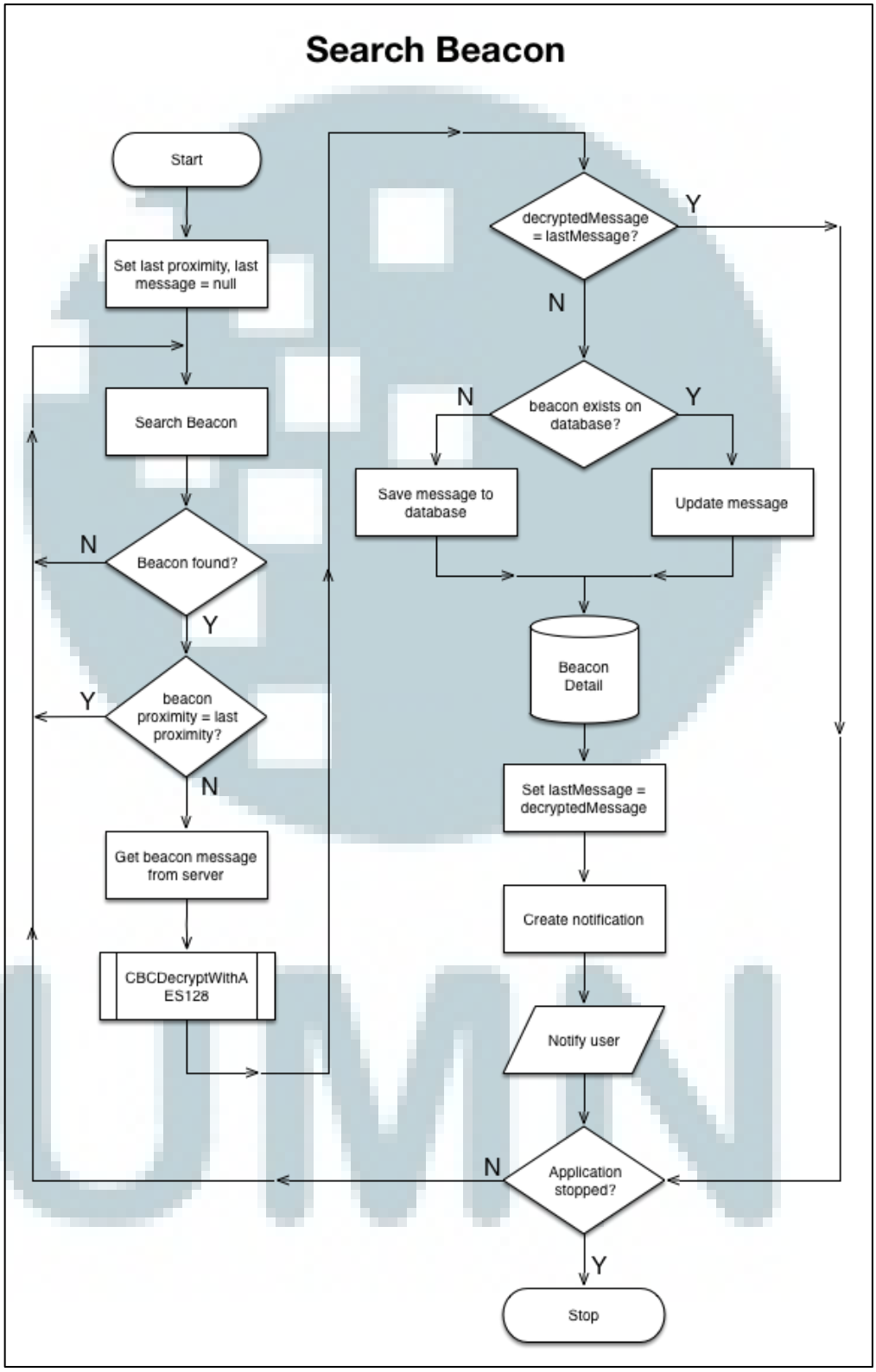
Apabila pesan yang diterima dari *beacon* tersebut merupakan pesan baru, maka sistem melakukan pengecekan terhadap informasi *beacon*, apakah beacon ini sudah terdaftar di *database* atau belum dengan melakukan pencarian terhadap UUID *string* dan nomor *major* milik *beacon*. Apabila *beacon* sudah pernah terdaftar, maka informasi yang ada di tabel Beacon Detail terkait *beacon* tersebut di-*update* sesuai dengan informasi baru yang diterima dari *beacon* tersebut. Tabel Beacon Detail memiliki struktur berupa UUID, nama *beacon*, nomor *major*, pesan terenkripsi dari *beacon* dan hasil dekripsi dari pesan tersebut.

Tabel 3.1. Struktur dari tabel Beacon Detail

Nama Field	Tipe Data	Deskripsi
UUID	String	*Universally Unique Identifiers* milik *beacon* yang digunakan untuk membedakan sebuah *beacon* milik suatu sistem dengan *beacon* milik sistem lain
beaconName	String	Nama dari *beacon*
major	Integer	Nomor *major* dari *beacon* yang digunakan untuk membedakan sebuah *beacon* dengan *beacon* lainnya dalam satu sistem
encryptedMessage	String	Pesan terenkripsi yang diterima dari *beacon*
decryptedMessage	String	Pesan hasil dekripsi dari pesan terenkripsi

Apabila *beacon* tersebut belum pernah terdaftar dalam *database*, maka sistem akan menyimpan informasi terkait dari *beacon* tersebut beserta dengan pesan yang dikirimkannya. Terakhir, sistem akan membuat sebuah notifikasi

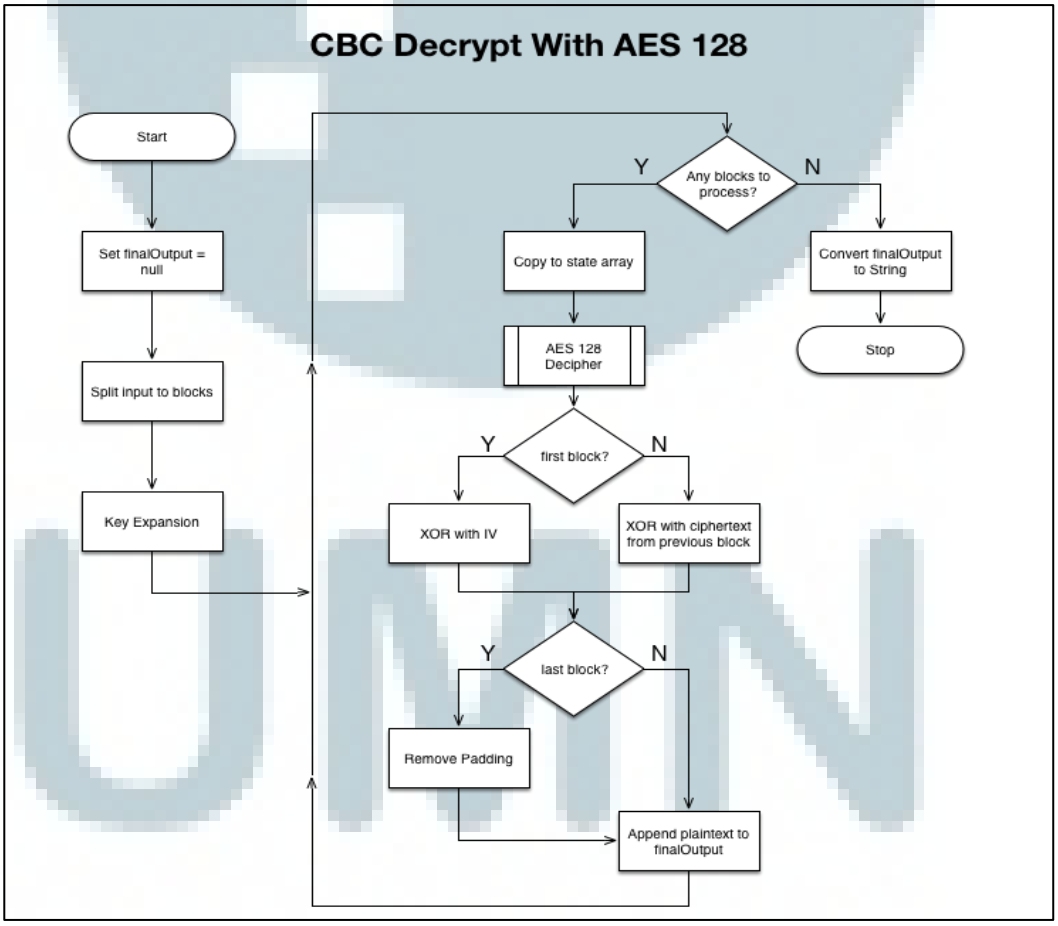
kepada pengguna. Isi dari notifikasi yang dibuat adalah nama dari *beacon* dan hasil dekripsi dari pesan terenkripsi yang diterima dari *beacon*.



Gambar 3.2. Diagram alir untuk subproses Search Beacon

3.2.3. CBC Decrypt With AES 128

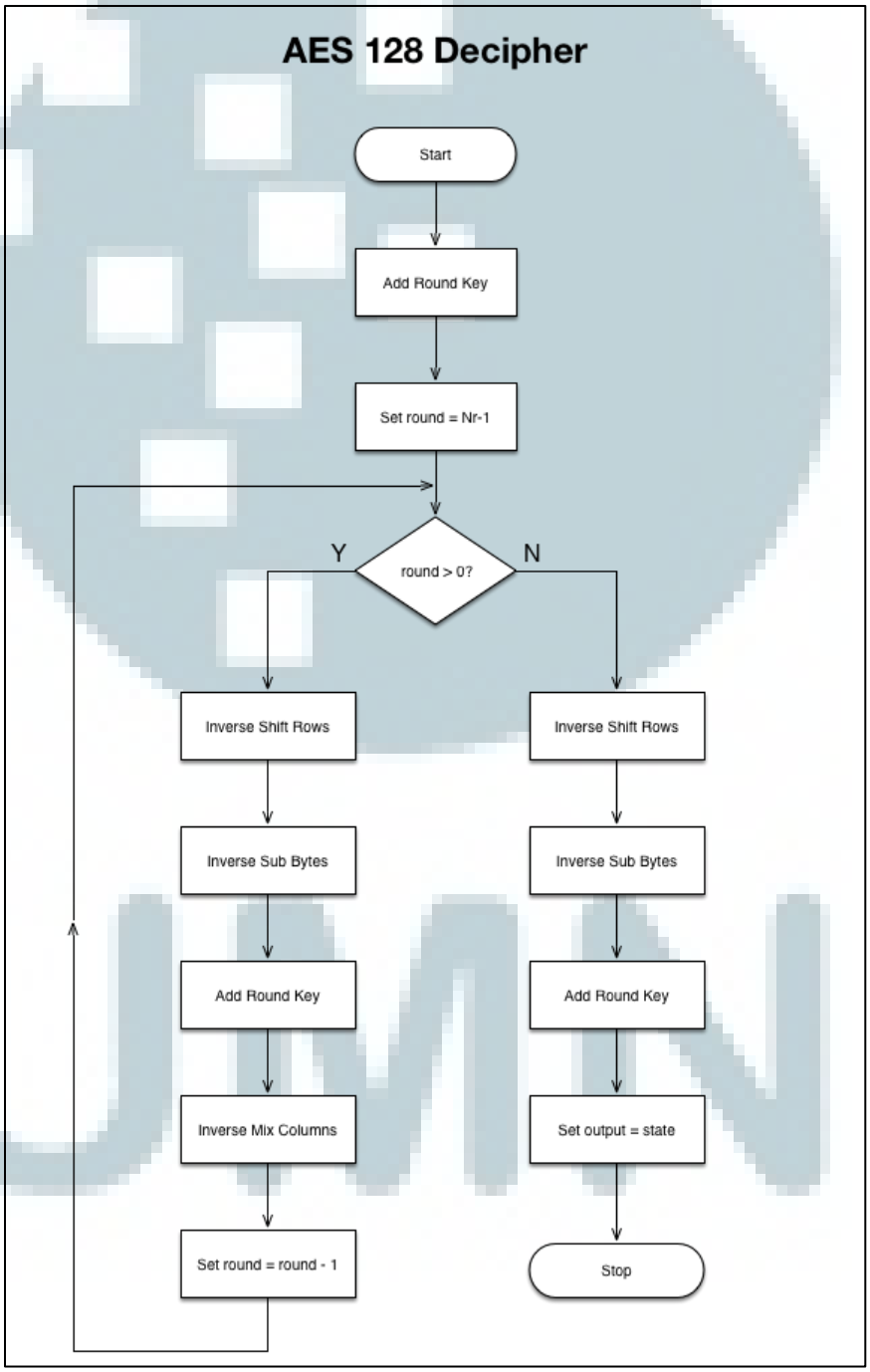
Pada diagram alir sebelumnya (Gambar 3.2) terdapat sebuah subproses dengan nama CBC Decrypt With AES 128. Proses ini adalah sebuah proses dimana pesan yang diterima dari *server* didekripsikan sehingga menjadi sebuah pesan biasa yang dapat dibaca dan dimengerti. Proses pada CBC Decrypt With AES 128 dimulai dengan memecah pesan menjadi beberapa blok dengan panjang masing-masing blok 16 kata, setelah itu proses dilanjutkan dengan melakukan tahap *Key Expansion* versi AES. Selanjutnya untuk setiap blok pesan dilakukan dekripsi, dan hasil dari dekripsi ditambahkan ke hasil akhir yang nantinya ditampilkan dalam bentuk notifikasi.



Gambar 3.3. Diagram alir untuk subproses CBC Decrypt With AES 128

3.2.4. AES 128 Decipher

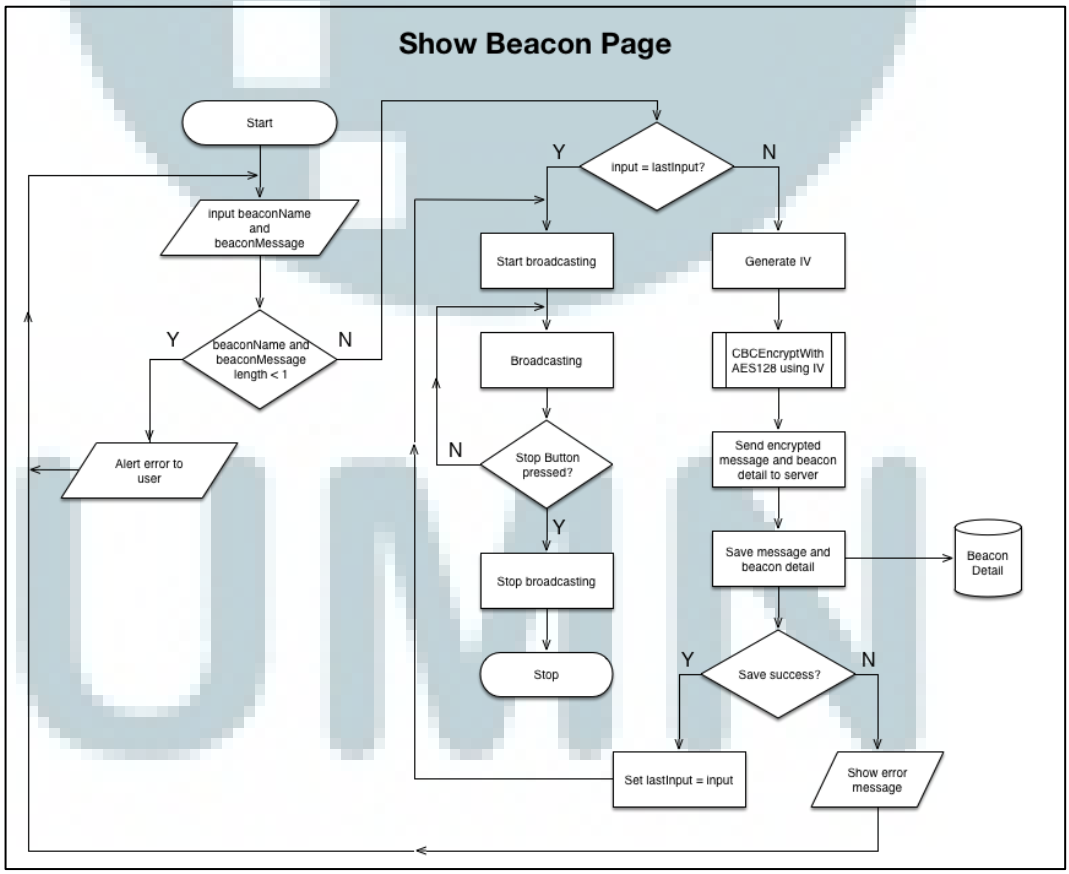
Subproses AES 128 Decipher pada gambar diagram alir sebelumnya adalah proses dimana dekripsi berlangsung. Di dalam proses ini terdapat empat buah proses utama untuk melakukan dekripsi seperti yang telah dijelaskan sebelumnya (lihat *pseudocode* tentang dekripsi di Gambar 2.22).



Gambar 3.4. Diagram alir untuk subproses AES 128 Decipher

3.2.5. Show Beacon Page

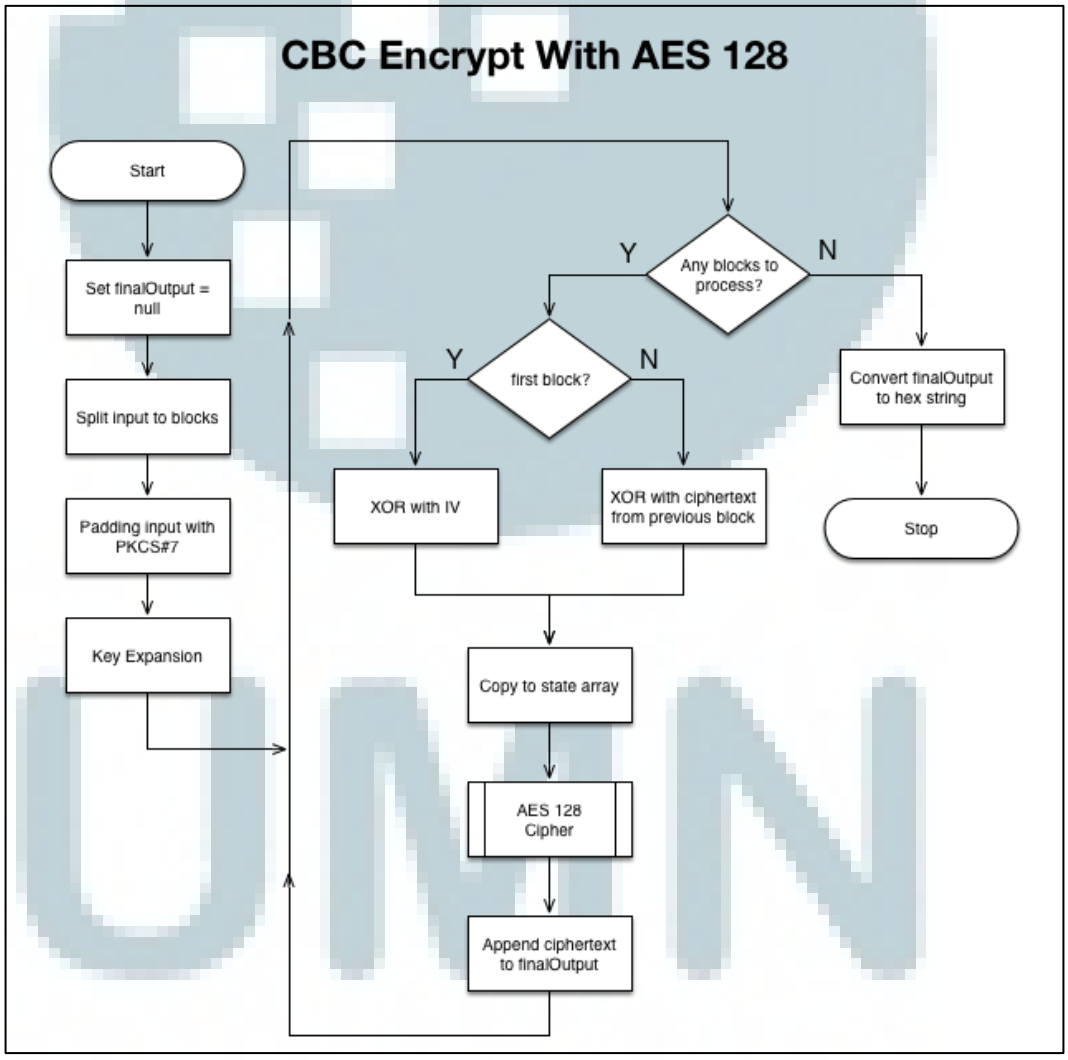
Subproses ini menampilkan sebuah halaman dimana pengguna dapat melakukan *broadcast*. Sebelum melakukan *broadcast*, pengguna diwajibkan untuk mengisi nama dari *beacon* serta pesan yang ingin di-*broadcast*. Setelah itu terdapat sebuah tombol untuk melakukan *broadcast*. Jika pesan yang dimasukkan pengguna sama dengan pesan sebelumnya, maka sistem akan langsung melakukan *broadcast*, tetapi jika tidak maka sistem akan mengenkripsi pesan tersebut sebelum dikirimkan ke *server* untuk disimpan ke dalam tabel Beacon Detail (struktur tabel sama dengan Tabel 3.1). Apabila penyimpanan informasi *beacon* berhasil maka sistem memulai *broadcast*, apabila tidak sistem akan menampilkan pesan *error*. Berikut adalah diagram alir dari subproses Show Beacon Page.



Gambar 3.5. Diagram alir untuk subproses Show Beacon Page

3.2.6. CBC Encrypt With AES 128

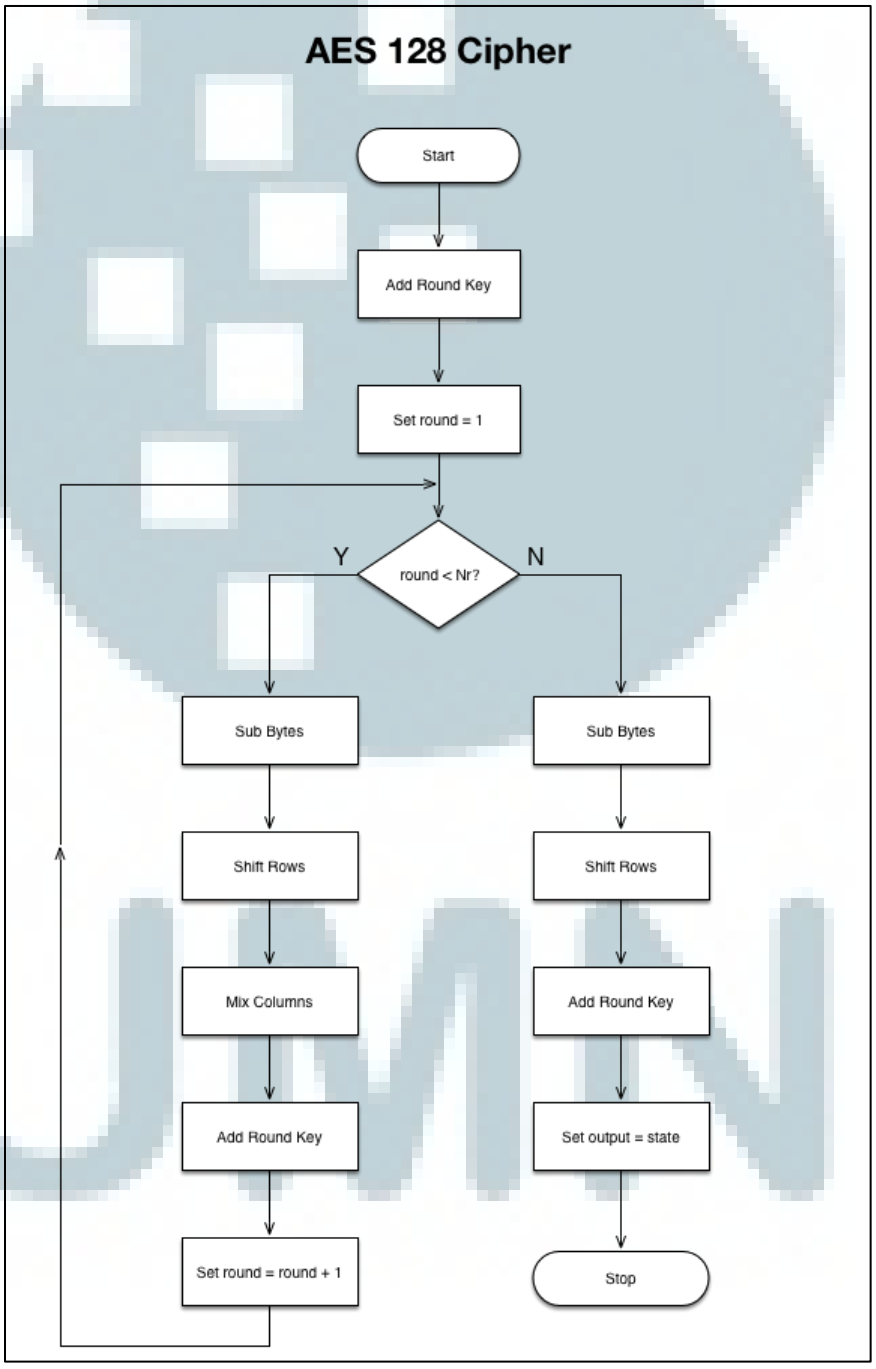
Di diagram alir Show Beacon Page, terdapat sebuah subproses dengan nama CBC Encrypt With AES 128. Subproses ini adalah proses dimana enkripsi menggunakan CBC (*Cipher Block Chaining*) terjadi. Proses enkripsi dimulai dengan memecah pesan ke dalam beberapa blok dengan panjang 16 kata. Pesan ini lalu dimasukkan ke dalam proses *padding* dengan metode PKCS#7. Setelah *padding*, sistem akan melakukan tahap *Key Expansion*. Setiap blok tadi akan dimasukkan ke dalam subproses enkripsi, yaitu AES 128 Cipher.



Gambar 3.6. Diagram alir untuk subproses CBC Encrypt With AES 128

3.2.7. AES 128 Cipher

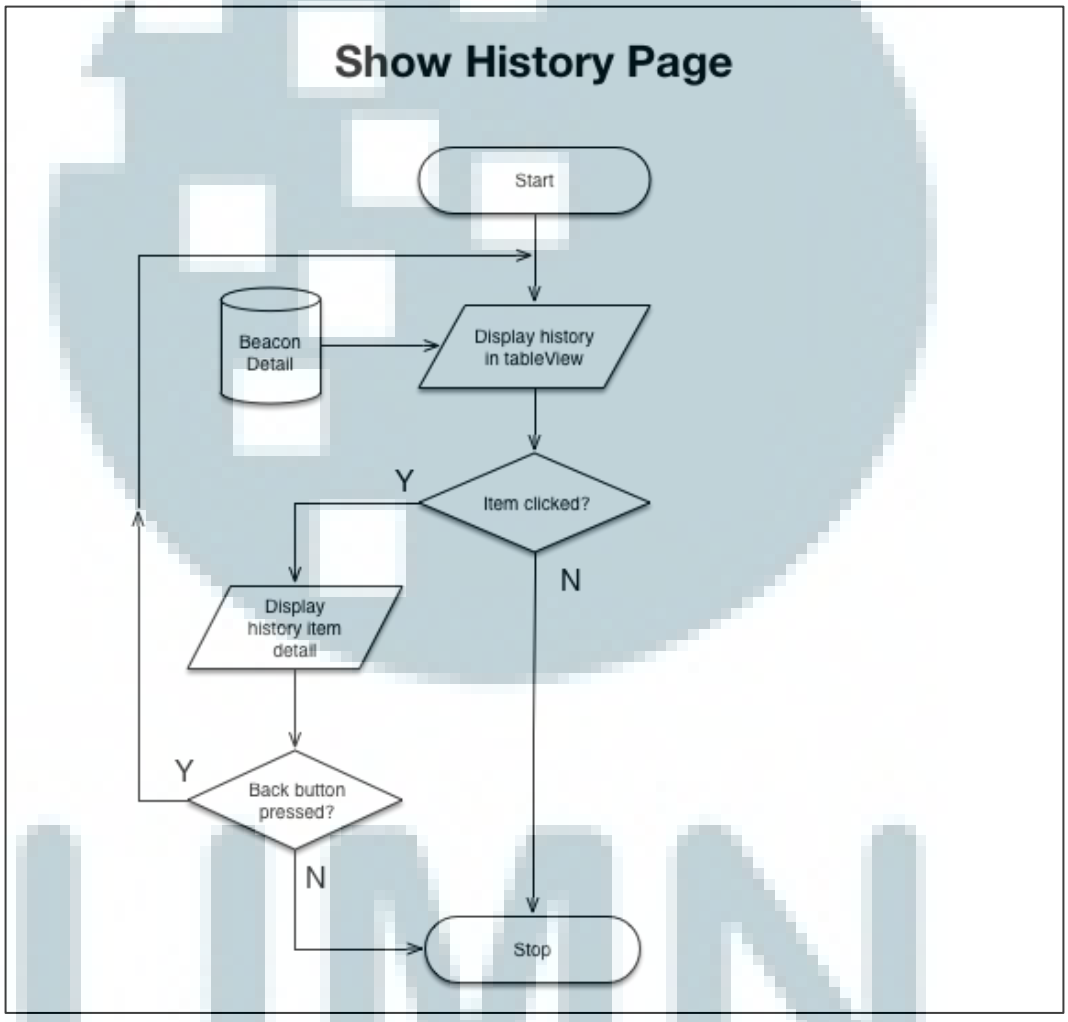
Subproses AES 128 Cipher yang ada di gambar sebelumnya adalah proses dimana terjadinya enkripsi AES. Proses enkripsi terdiri dari empat buah tahap utama, yaitu *Add Round Key*, *Sub Bytes*, *Shift Rows*, dan *Mix Columns* (lihat penjelasan enkripsi di Gambar 2.4).



Gambar 3.7. Diagram alir untuk subproses AES 128 Cipher

3.2.8. Show History Page

Subproses Show History Page digunakan untuk menampilkan daftar pesan yang pernah diterima pada perangkat pengguna. Daftar *history* diambil dari *database* yang berisikan informasi *beacon detail*. Pengguna dapat mengklik *item* yang ada pada tampilan untuk melihat detail *history* lebih lanjut. Pengguna juga diberikan navigasi untuk kembali ke daftar *history*.



Gambar 3.8. Diagram alir untuk subproses Show History Page