DAFTAR SINGKATAN

IP	: *Internet Protocol*
PTT	: *Push-to-talk*
UDP	: *User Datagram Protocol*
QoS	: *Quality of Service*

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dewasa ini teknologi komunikasi sekarang sudah sangat berkembang dengan pesat. Salah satu media yang paling berkembang dengan pesat adalah komunikasi pada *smarphone* dengan media suara. Banyak sekali aplikasi – aplikasi yang disediakan sebagai media komunikasi suara, salah satunya adalah *Push to Talk* (PTT). *Push-to-talk* adalah salah satu komunikasi dua arah berupa komunikasi suara yang bekerja layaknya *walkie-talkie*. Komunikasi pada *push-to-talk* ini bekerja secara *half-duplex* yaitu pengguna tidak dapat mengirim dan menerima informasi secara bersamaan. *push-to-talk* menggunakan jaringan *Internet Protocol* (IP) untuk melakukan pertukaran informasi yang dilakukan[1].

Dalam proses saling melakukan pertukaran informasi, keamanan merupakan suatu hal yang dianggap sangat penting agar informasi yang dianggap rahasia tidak dapat diketahui oleh pihak yang tidak berhak. Layanan jaringan *push-to-talk* yang menggunakan layanan IP sangat rentan terhadap penyadapan. Sehingga untuk mengamankan layanan komunikasi *push-to-talk* diperlukan sebuah sistem pengamanan yang tinggi agar tidak terjadi.

Ada beberapa cara yang dapat digunakan untuk mengamankan layanan *push-to-talk*, salah satunya dengan teknik kriptografi pada paket-paket data ataupun informasi yang kita kirimkan agar tidak dapat diakses oleh orang ketiga atau yang tidak memiliki hak akses. Sehingga hanya pengirim dan penerima yang dapat mengakses data tersebut. Metode enkripsi menggunakan algoritma kriptografi HC-256 merupakan salah satu metode kriptografi yang dapat digunakan untuk mengamankan layanan *push-to-talk*. Pemilihan algoritma HC-256 karena beroperasi sangat cepat dan dianggap memiliki ketahanan yang bagus karena masih belum ditemukan adanya celah yang bisa di tembus dengan sempurna dari algoritma ini[3].

1.2 Perumusan Masalah

Berdasarkan latar belakang yang telah dikemukakan, terdapat beberapa rumusan masalah antara lain.

2. Aplikasi *push-to-talk* yang belum memiliki fitur enkripsi untuk keamanan pada informasi data.
3. Mengimplementasi teknik kriptografi dengan algoritma stream cipher HC-256 pada aplikasi *push-to-talk*. Performansi kecepatan pengiriman, *delay, packet loss, throughput,* kecepatan proses Enkripsi dan Dekripsi , dan *avalanche effect* proses enkripsi pada data suara di *push-to-talk*.

1.3 Tujuan

Tujuan pembuatan aplikasi ini adalah.

1. Membuat dan merancang aplikasi *push-to-talk* pada smartphone yang memiliki keamanan fitur enkripsi dengan Algoritma kriptografi stream cipher.
2. Menganalisis performansi algoritma kriptografi HC-256 yang di implementasikan pada *push-to-talk*.
3. Menganalisis performansi *push-to-talk* yang telah memiliki fitur enkripsi.
4. Menganalisis hasil pengujian algoritma kriptografi HC-256 dengan Salsa20 yang telah di implementasikan pada *push to talk* berdasarkan parameter pengujian.

1.4 Batasan Masalah

1. Aplikasi ini dirancang untuk smartphone dengan berbasis Android.
2. Data yang dienkripsi berupa suara. Sistem yang diterapkan adalah metode/teknologi *push-to-talk*.
3. Tidak melakukan kompresi pada data suara dan algoritma yang digunakan. Tidak membahas tentang pendistribusian dan pembentukan kunci pada algoritma.
4. Performansi yang diukur adalah *avalanche effect*, waktu proses enkripsi dan dekripsi, *packet loss, delay*, dan *throughput*.
5. Pembobolan atau penyerangan tidak dibahas.

1.5 Metodologi Penyelesaian Masalah

Metodologi penelitian yang digunakan adalah.

a. Studi literatur, yaitu mempelajari literatur-literatur yang ada sesuai dengan permasalahan yang akan dibahas meliputi, Aplikasi *push-to-talk*, konsep kriptografi, teori algoritma kriptografi HC-256, dan teori mengenai uji performansi menggunakan parameter waktu enkripsi dan dekripsi, avalanche effect dan parameter QoS seperti, *delay*, *throughput*, dan *packet loss*.
b. Perancangan sistem dan perancangan fungsionalitas serta pembuatan desain antarmuka bagi user dengan menggunakan bahasa pemograman java Android Studio dan beberapa pengujian dengan menggunakan bahasa pemograman java Netbeans IDE 7.3.1.
c. Melakukan uji coba aplikasi yang telah dibuat, serta menganalisa masalah-masalah yang muncul, serta melakukan perbaikan terhadap masalah-masalah tersebut.
d. Pembuatan laporan dari hasil penelitian

1.6 Sistematika Penulisan TA

Adapun sistematika penulisan pada Tugas Akhir ini adalah.

BAB I PENDAHULUAN

Berisi tentang latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, metodologi penelitian, dan sistematika penelitian

BAB II DASAR TEORI

Berisi tentang penjelasan mengenai kriptografi, Aplikasi Push-to-talk, algoritma HC-256, dan penjelasan mengenai parameter uji.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Berisi tentang perancangan sistem dan proses analisis yang akan dibangun pada Tugas Akhir ini.

BAB IV IMPLEMENTASI DAN PENGUJIAN SISTEM

Berisi tentang implementasi dan pengujian dari aplikasi yang telah dibuat seperti performansi algoritma terhadap sistem dan analisis hasil penelitian

BAB V KESIMPULAN DAN SARAN

Berisi kesimpulan dari hasil penelitain yang telah dilakukan dan rekomendasi untuk penelitian berikutnya

BAB II
DASAR TEORI

2.1 Kriptografi

Kriptografi adalah ilmu yang mempelajari cara agar suatu informasi tetap aman saat dikirimkan dari pengirim ke penerima tanpa perlu takut informasi bocor atau diketahui oleh orang ketiga. Jadi algoritma kriptografi adalah langkah-langkah yang digunakan untuk menyembunyikan pesan dari orang-orang yang tidak memeliki hak akses terhadap informasi tersebut. Ada pun beberapa prinsip mendasar pada kriptografi, yaitu.

1. *Authentication*, merupakan aspek yang berhubungan dengan identitas, memungkinkan penerima pesan dapat memastikan keaslian pengirim atau sumber pesan.
2. *Confidenlity* (kerahasiaan), merupakan aspek yang bertujuan untuk menjaga agar pesan tidak dapat dibaca oleh pihak-pihak yang tidak berhak.
3. Data integrity (keutuhan data), merupakan aspek yang dapat memastikan bahwa pesan tidak dimodifikasi atau diubah selama pengiriman sehingga penerima dapat menerima pesan yang asli.
4. *Non-repudiation*, yaitu aspek yang bertujuan untuk mencegah pengirim dan penerima yang berkomunikasi melakukan penyangkalan, baik pengirim pesan menyangkal melakukan pengiriman maupun penerima menyangkal telah menerima pesan.

Algoritma kriptografi terdiri dari tiga fungsi dasar, yaitu.

1. Enkripsi, ini merupakan hal yang sangat penting dalam kriptografi. merupakan pengamanan data yang dikirimkan agar terjaga kerahasiaannya. Pesan asli disebut plaintext, yang diubah menjadi kode-kode yang tidak dimengerti.

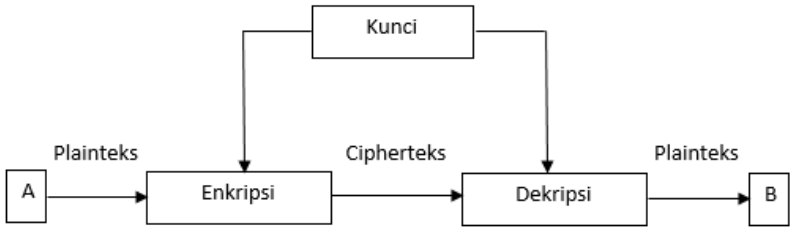
2. Dekripsi, ini merupakan kebalikan dari enkripsi. Pesan yang telah dienkripsi dikembalikan ke bentuk asalnya, disebut dengan dekripsi pesan.

3. Kunci, yang dimaksud di sini adalah kunci yang dipakai untuk melakukan enkripsi dan dekripsi. Kunci terbagi menjadi dua bagian, kunci rahasia (*private key*) dan kunci umum (*public key*).

2.1.1 Jenis-jenis Kriptografi

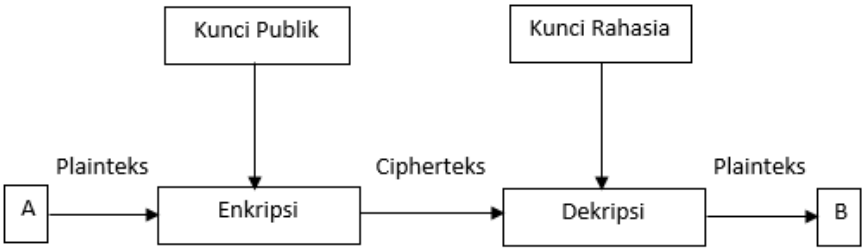
Berdasarkan kunci yang digunakan, algoritma kriptografi dapat dibagi menjadi 3 yaitu.

1. Kriptografi Simetri, menggunakan kunci yang sama untuk melakukan proses enkripsi dan dekripsi.



Gambar 2.1 Kriptografi Simetris [4]

2. Kriptografi Asimetri, menggunakan kunci yang berbeda (kunci publik dan kunci rahasia) untuk melakukan proses enkripsi ataupun proses dekripsi.

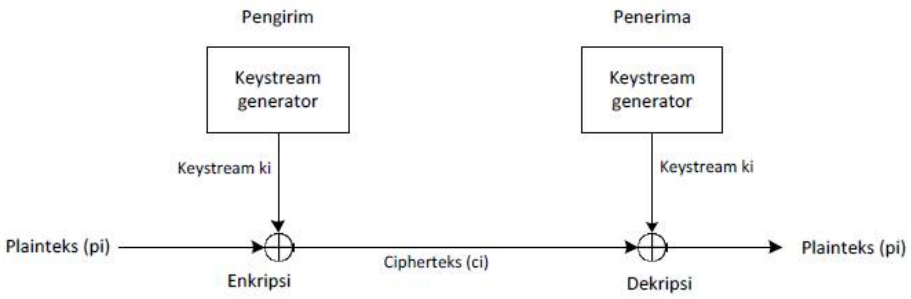


Gambar 2.2 Kriptografi Asimetris [4]

3. Fungsi Hash, fungsi satu arah (*one-way function*) yang merupakan suatu fungsi matematika yang mengambil masukan panjang variabel dan mengubahnya ke dalam urutan biner dengan panjang yang tetap.

Algoritma kriptografi Simetri dapat dibagi menjadi 2 jenis, yaitu.

1. *Cipher* Block (*block cipher*), Algoritma kriptografi ini beroprasi dalam bntuk block bit, bit-bit yang ingin di enkripsi di rangkai menjadi blok-blok bit panjangnya sudah ditentukan sebelumnya.
2. *Cipher* Stream (*Stream Cipher*), algoritma kriptograf ini beroprasi dalam bentuk bit tunggal, yang artinya proses enkripsi dilakukan 1 bit per 1 bit.



Gambar 2.3 Konsep cipher aliran (stream cipher) [5]

2.1.2 Algoritma Kriptografi HC-256

Algoritma kriptografi HC-256 adalah algoritma kriptografi aliran. Algoritma kriptografi ini di buat untuk melakukan proses enkripsi yang besar dalam suatu perangkat lunak dengan kecepatan yang tinggi dan memiliki keamanan yang tangguh. Kriptogafi HC-256 membangkitkan bit kunci (*keystream*) dari 256 bit kunci rahasia dan 256 bit *Initialitation Vector*. Kriptografi HC-256 memiliki 2 tabel rahasia, yang masing-masing mempunya 1024 elemen, dengan masing-masing elemen berukuran 32 bit. Setiap langkah akan memperbarui elemen tabel menggunakan *non-linear* fungsi umpan balik. Semua elemen-elemen kedua tabel diperbarui setiap 1024 langkah dan masing-masing menghasilkan 32 bit output *keystream*.[1]

A. Operasi, Variable dan Fungsi

Algoritma kriptografi HC-256 memiliki operator-operator sebagai berikut

a. + : x + y berarti x + y mod 232, dimana $0 \leq x \leq 232$.

b. – : x – y berarti x – y mod 1024.

c. ^ : x ^ y berarti XOR.

d. || : x || y berarti konkatenansi.

e. >> : x >> n berarti x digeser ke kanan sejauh n.

f. << : x << n berarti x digeser ke kiri sejauh n.

g. >>> : x >>> n berarti ((x >> n) ^ (x << (32 - n))) dimana $0 \leq n < 32$, $0 \leq x < 232$.

Pada algoritma HC-256, terdapat 2 buat S-box (tabel) rahasia, yaitu P dan Q. Masing-masing berisi 1024 elemen dengan 32 bit per elemen, Notasi-notasi yang digunakan dalam pemrosesan S-box yaitu.

a. *P* : tabel dengan 1024 32-bit elemen. Tiap elemen digambarkan dengan *P[i]* dengan $0 \leq i \leq 1024$.

b. *Q* : tabel dengan 1024 32-bit elemen. Tiap elemen digambarkan dengan *Q[i]* dengan $0 \leq i \leq 1024$.

c. *K* : kunci 256 bit yang digunakan HC-256.

d. *IV* : Initialisation Vector 256 bit dalam HC256.

e. *s* : keystream total yang dihasilkan HC256. 32 bit keystream yang dihasilkan pada langkah ke *i* disebut *si*.

256-bit kunci array K[0, . . . , 3] dan 256-bit inisialisasi vector IV [0, . . . , 3] digunakan, di mana setiap masukan dari array yang merupakan elemen dari 32-bit. s ditandai sebagai *keystream* yang dibuat oleh langkah ke-i, i = 0, 1, 2, . . .

Dalam Algoritma HC-256 digunakan beberapa fungsi yaitu.

a. $f1(x) = (x >>> 7) \wedge (x >>> 18) \wedge (x >> 3)$

b. $f2(x) = (x >>> 17) \wedge (x >>> 19) \wedge (x >> 10)$

c. $g1(x; y) = ((x >>> 10) \wedge (y >>> 23)) + Q\,[(x \wedge y) \bmod 1024]$

d. $g2(x; y) = ((x >>> 10) \wedge (y >>> 23)) + P\,[(x \wedge y) \bmod 1024]$

e. $h1(x) = Q\,[x0] + Q\,[256 + x1] + Q\,[512 + x2] + Q\,[768 + x3]$

f. $h2(x) = P\,[x0] + P\,[256 + x1] + P\,[512 + x2] + P\,[768 + x3]$

dimana $x = x3||x2||x1||x0$. x adalah karakter 32 bit dengan $x0,x1,x2$ dan $x3$ masing-masing 4 bit. [1]

B. Inisialisasi Proses

Dalam proses inisialisasinya, yang pertama kali dilakukan adalah memasukkan key dan IV kedalam S-box (tabel) P dan Q, dan melakukan pembangkitan *keystream* sebanyak 4096 langkah.

1. Tentukan K = K0 ‖ K1 ‖ K2 ‖ K3 dan IV = IV0 ‖ IV1 ‖ IV2 ‖ IV3, dimana setiap Ki dan IVi terdiri dari 32-bit angka. K dan IV ini kemudian dimasukkan kedalam array Wi(0≤i≤2559) sebagai berikut [1].

$$wi\begin{cases} Ki & 0 \le i \le 7 \\ IV_{i-8} & 8 \le i \le 15 \\ f2(W_{i-2}) + W_{i-7} + f1(W_{i-15}) + W_{i-16} + i & 16 \le i \le 2559 \end{cases}$$

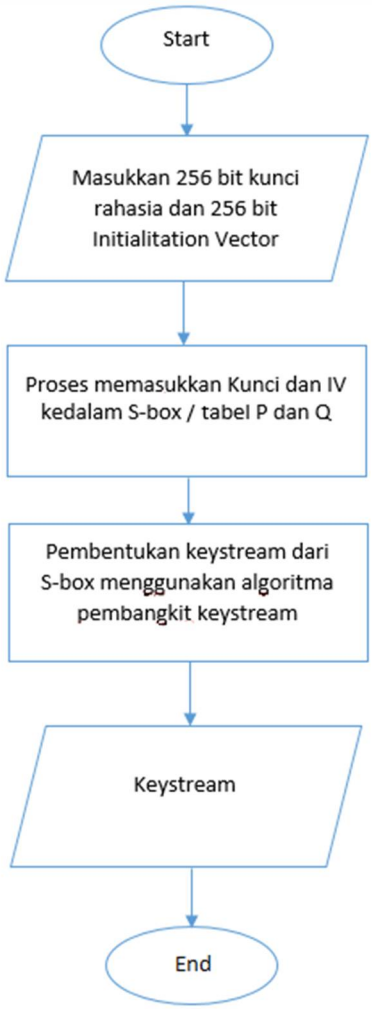
2. Memperbarui table P dan Q ke dalam array W.
   P[i] = Wi+512 untuk 0 ≤ i ≤ 1023 (8)
   Q[i] = Wi+1536 untuk 0 ≤ i ≤ 1023
3. Membangkitkan *keystream* sebanyak 4096 kali dengan algoritma pembangkitan *keystream*.

C. Pembangkitan Kunci

Setelah initialisasi dilakukan, *keystream* siap untuk dibangkitkan atau dibuat. Di setiap langkah, satu elemen tabel diperbarui dan satu keluaran 32-bit dibuat. Setiap S-box digunakan untuk menghasilkan hanya 1024 keluaran, kemudian S-box diperbarui dalam 1024 langkah-langkah berikutnya.

```
procedure keystreamGen
{
i = 0;
repeat until enough keystream bits are
```

```
generated
{
j = i mod 1024;
if (i mod 2048) < 1024{
P[j] = P[j]+P[j-10] + g1(P[j-3],P[j-1023]);
si = h1(P[j-12]) ^ P[j];
}
else{
Q[j] = Q[j]+Q[j-10] + g2(Q[j-3],Q[j-1023] );
si = h2(Q[j - 12] ) ^ Q[j];
}
end-if
i = i + 1;
}
end-repeat
}
```



Gambar 2.4 *Flowchat* pembentukan *keystream*

2.2 Push to Talk

Push-to-talk adalah salah satu komunikasi dua arah berupa komunikasi suara yang bekerja layaknya *walkie-talkie*. Komunikasi pada *push-to-talk* ini bekerja secara *half-duplex* yaitu pengguna tidak dapat mengirim dan menerima informasi secara bersamaan. Push to Talk banyak digunakan dalam praktek militer, polisi, keamanan dan komunikasi pada radio darurat.[2]

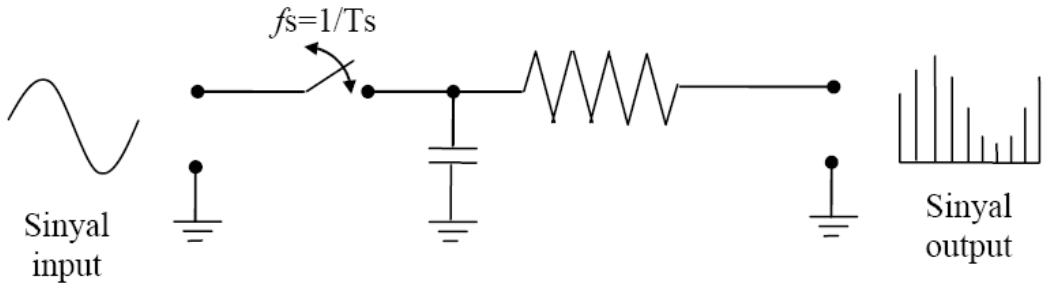
Sekarang ini sudah banyak aplikasi push-to-talk yang digunakan dan diimplementasikan pada smartphone. Aplikasi push-to talk pada smartphone menggunakan jaringan IP sebagain jaringannya. Salah satu protokol yang biasa digunakan untuk untuk mengirimkan data suara adalah UDP. User Datagram

Protocol merupakan salah satu protocol pada *layer transport* pada TCP/IP yang mengimplementasikan protocol data stream yang *connectionless*.[3]

2.3 Dasar Audio Processing

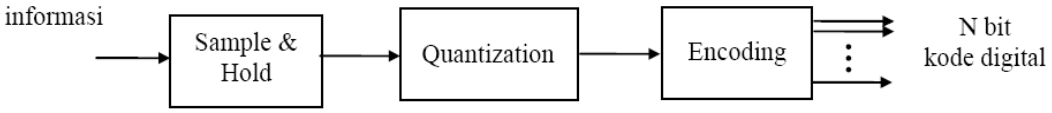
Pada prinsipnya sinyal audio sederhana dinyatakan dalam sinyal sinusoidal, Proses konversi audio analog ke audio digital prinsipnya sama dengan proses ADC (Analog to Digital Converter) pada sinyal analog. Untuk mengubah sinyal analog menjadi digital dilakukan proses *sampling*, *quantization* dan *encoding*.[11]

Sampling adalah proses pengambilan nilai pada titik-titik diskrit sepanjang variable waktu dari sinyal kontinu sehingga didiapkan sinyal waktu diskrit. [5]



Gambar 2.5 Rangkaian Sampling

Rangkaian sampling diatas merupakan proses dari *Analog* to *Digital Conversion* (ADC).



Gambar 2.6 Block Diagram Rangkaian ADC

Quantization adalah proses memetakan nilai sinyal kontinu menjadi diskrit sehingga didapatkan nilai sinyal diskrit, lalu dilakukan proses encoding dimana proses tersebut mengubah nilai-nilai sinyal menjadi bilangan biner. [5]

Nilai bit dari kode digital tersebut akan dijadikan sebagai *plaintext* yang akan melalui proses enkripsi dan dekripsi untuk mengamankan data suara pada komunikasi push to talk

2.4 Kriptografi pada Data Suara

Suara merupakan sesuatu yang setiap hari kita dengar dan gunakan dan suara biasa dipakai untuk berkomunikasi. Sekarang ini suara sudah dapat diubah menjadi bit data dan digunakan dalam berbagai hal seperti di simpan dalam bentuk file atau sebagi alat komunikasi dengan internet. Data suara ini sering kita gunakan untuk berbagi informasi dengan orang lain, tetapi kita hanya ingin orang-orang tertentu yang mendapatkan informasi ini, di sinilah kriptografi berperan penting.

Dengan menggunakan kriptografi, kita membuat informasi menjadi rahasia dan hanya orang tertentu yang dapat mengaksesnya. Hal ini juga dapat di diterapkan pada data suara. Seperti pada komunikasi suara dengan internet, kita bisa mengubah atau mengacak bit suara yang akan dikirim. Begitu juga dengan file, hanya aplikasi tertentu yang telah dimplementasi kriptografi yang dapat mengeluarkan informasi suara yang ada pada file.

2.5 Uji Performansi

Adapun parameter yang akan diuji dalam penelitian ini adalah

1. *Avalanche Effect*

   Merupakan Salah satu cara untuk menguji ketahanan kriptografi adalah dengan mengukur nilai avalanche effect kriptografi tersebut. Pengukuran avalanche effect meliputi pengaruh dari perubahan satu bit plain text, kunci enkripsi, dan cipher text [5].

2. Waktu proses enkripsi dan dekripsi

   Waktu proses dilakukan untuk melihat perbandingan waktu proses saat enkripsi suara dan dekripsi suara menggunakan algoritma kriptografi HC-256.

3. *Delay*

   Lamanya proses pengiriman suara, mulai dari data suara terenkripsi dikirim sampai data diterima.

4. *Packet Loss*

   Paket data yang hilang pada saat pengiriman data karena gangguan tertentu dalam batas waktu tertentu [5].

5. *Throughtput*

Throughtput adalah jumlah bit yang berhasil diterima melalui sebuah media komunikasi [5]

BAB III
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Kebutuhan

3.1.1 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut.

a. Sistem operasi Windows 8.1 Pro
b. Sistem Operasi Android 5.0 (*Lollipop*)
c. Android Studio
d. *Neatbeans IDE 7.3.1*
e. *Java Development Kit 8*
f. *Java Runtime Environment 8*

3.1.2 Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam penelitian Tugas Akhir ini adalah sebagai berikut.

3.1 Laptop Processor Intel Core i5, RAM 8GB
3.2 Android Dual-core 2.0 GHz, RAM 2GB

3.2 Deskripsi Sistem

Aplikasi yang digunakan adalah aplikasi berbasis java pada android. Aplikasi ini akan mengirimkan suara dari pengirim ke penerima saat tombol push ditekan. Dasarnya aplikasi yang akan digunakan ini bekerja seperti walkie-talkie. Aplikasi ini menggunakan jangingan IP yang di koneksikan dengan *Access Point* atau *wireless* sebagai jalur pengiriman data dan menggunakan protokol UDP untuk pengiriman data. Data suara akan di *broadcast* ke penerima.

Pada aplikasi ini akan dilakukan implementasi algoritma kriptografi HC-256. Aplikasi ini akan memiliki fitur enkripsi dan dekripsi di dalam systemnya. Sesaat sebelum paket data suara dikirim, data tersebut akan dienkripsi lalu dikirim.