BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Pembahasan

Aplikasi dengan judul "Implementasi Algoritma Kriptografi Dengan Metode Vigenere Cipher Untuk Pengamanan Sms Pada Android" dibangun dengan menggunakan bahasa pemrograman java. Berdasarkan analisis dan perancangan sistem yang telah dibuat pada Bab sebelumnya, maka untuk Bab ini, akan diimplemantasikan ke dalam sebuah sistem yang dapat dioperasikan dalam keadaan yang sebenarnya.

Pada bab ini akan dibahas mengenai implementasi dan pembahasan secara menyeluruh dengan menyertakan tampilan aplikasi dan potongan kode program.

4.1.1. Potongan Skript Class Vigenere Cipher

Class ini berisi implementasi algoritma enkripsi metode *Vigenere,* berikut ini cuplikan kode program :

```
public class Vigenere_Chiper {

      public String enkrip(String PlainText, String Kunci){
      String hasil = "";

      for(int i = 0, j = 0; i < PlainText.length(); i++){
            char p = PlainText.charAt(i);
            char k = Kunci.charAt(j);
```

```
            //Rumus
            // C = (P + K) % 255
            int chiper = ((int) p + (int) k) % 255;
            hasil += (char) chiper;
            j = ++j % Kunci.length();
      }

      return hasil;
}

      public String dekrip(String dekrip, String Kunci){
      String hasilDeskript = "";

      for(int i = 0, j = 0; i < dekrip.length(); i++){
            char p = dekrip.charAt(i);
            char k = Kunci.charAt(j);

            //Rumus
            // C = (P + K) % 255
            int chiper = ((int) p - (int) k) % 255;
            hasilDeskript += (char) chiper;
            j = ++j % Kunci.length();
      }

      return hasilDeskript;
}

}
```

Potongan program diatas adalah class vigenere cipher dimana terdapat method enkripsi dan method dekripsi.

4.1.2 Class Baca SMS

Dalam class Baca SMS ini terdapat beberapa method diantaranya adalah :

Method

```
      private void dekrip() {
            try{
            Vigenere_Chiper vigenere_Chiper = new
Vigenere_Chiper();
            Skunci=kunci.getText().toString();
            Spesan=pesan.getText().toString();

                  if(Skunci.length()>0) {
                  //--Vigenere Chiper
                  //menDekripsi pesan - mengubah hasil bentuk
                  Enkripsi ke Deskripsi
                  enkript_chiper = vigenere_Chiper.dekrip(Spesan,
                  Skunci);
                  //cetak hasil enkrip dalam bentuk heksadesimal
                  hasil2.setText(enkript_chiper);

                  byte[] b = Spesan.getBytes();
                  String s = new String(b, "UTF-8");
                  System.out.println("Deskript : "+Spesan+" :
"+s);
                  }
                  else {
                  Toast.makeText(getBaseContext(),"kunci tidak
                  boleh kosong", Toast.LENGTH_SHORT).show();
                        }
                  }

            }
```

Potongan script diatas merupakan class baca sms, terdapat sebuah method dengan nama dekrip. Method dekrip ini digunakan untuk mendekripsi pesan masuk.

4.1.3 Class Tulis Pesan

Dalam class Tulis Pesan ini terdapat beberapa method diantaranya adalah :

```
//operasi load kontak
private void kontak() {
    Intent intent = new Intent(Intent.ACTION_PICK,
    ContactsContract.Contacts.CONTENT_URI);
    startActivityForResult(intent, RQS_PICK_CONTACT);
  }
```

```
//operasi enkrip
  private void enkrip() {
  Vigenere_Chiper vigenere_Chiper = new
  Vigenere_Chiper();
  Skunci=kunci.getText().toString();
  Spesan=pesan.getText().toString();
```

Pada potongan program diatas digunakan untuk melakukan enkripsi pesan dengan cara memasukkan kunci pada pesan melalui objek vigenere cipher pada Skunci dan Spesan.

```
//pengecekan panjang kunci
  if (Skunci.length()==0){
        Toast.makeText(getBaseContext(),"Kunci belum
        terisi", Toast.LENGTH_SHORT).show();
        }

        else if (Skunci.length()>0 && Spesan.length()>0)
        {
        //--Vigenere Chiper
        //mengenkripsi pesan - mengubah hasil enkripsi
        ke bentuk ASCII 256 bit
        enkript_chiper = vigenere_Chiper.enkrip(Spesan,
        Skunci);
        //cetak hasil enkrip dalam bentuk heksadesimal
        hasil.setText(enkript_chiper);


        }
        else {
        Toast.makeText(getBaseContext(),"Pesan Tidak
        Boleh Kosong!!", Toast.LENGTH_SHORT).show();
        }
  }
```

Potongan program diatas adalah untuk mengecek panjang kunci, yaitu dengan cara jika kunci panjangnya sama dengan 0 maka tampil pesan kunci belum terisi, kemudian jika kunci lebih panjang dari 0 dan pesan lebih panjang dari 0 maka dilakukan proses enkripsi pada pesan, selanjutnya pesan akan ditampilkan.

```
//operasi kirim pesan
private void send() {
no=NoTujuan.getText().toString();
pesanEnkrip=hasil.getText().toString();

if(no.length()>0 && pesanEnkrip.length()>0) {
      sendSMS(no,pesanEnkrip);
            pindah();
}
else if (no.length()>0 && hasil.length()==0) {
      Toast.makeText(getBaseContext(),"Pesan Kosong /
      Belum Terenkripsi", Toast.LENGTH_SHORT).show();
}
else {
      Toast.makeText(getBaseContext(),"Nomor Tujuan
      Belum Terisi", Toast.LENGTH_SHORT).show();
}
}
```

Potongan program diatas adalah proses operasi kirim pesan, dengan cara memasukkan nomor tujuan, pesan enkripsi. Jika panjang nomor dan pesan lebih dari 0 maka dilakukan pengiriman pesan.

```
  //ambil nomor dari kontak hp
@SuppressWarnings("deprecation")
  @Override
protected void onActivityResult(int requestCode, int
resultCode, Intent data) {
  super.onActivityResult(requestCode, resultCode, data);
  if (requestCode == RQS_PICK_CONTACT &&
  resultCode==Activity.RESULT_OK) {
      Uri contactData = data.getData();
      ContentResolver cr = getContentResolver();
      Cursor cur = managedQuery(contactData, null,
      null, null, null);
      if (cur.getCount() > 0) {
            while (cur.moveToNext()) {
            //ambil id contact
            String id = cur.getString(cur
      .getColumnIndex(ContactsContract.Contacts._ID));
            //cek jumlah nomor pada contact yg dipilih
            if (Integer
      .parseInt(cur.getString(cur
      .getColumnIndex(ContactsContract.Contacts.HAS_PH
      ONE_NUMBER))) > 0) {
      Cursor pCur = cr
            //query ke SQLite Contact
      .query(ContactsContract.CommonDataKinds.Phone.CO
      NT ENT_URI,ull,
```

```
ContactsContract.CommonDataKinds.Phone.CONTACT_ID
      + " = ?", new String[] { id },null);
       while (pCur.moveToNext()) {
 //ambil nomor berdasarkan id yang dipilih
 String nomorHp = pCur.getString(pCur
 .getColumnIndex(ContactsContract.CommonDataKinds.Phon
 e.DATA));
 NoTujuan.setText(nomorHp);
 }
 pCur.close();


             }
      }
  }
```

Potongan program diatas adalah operasi untuk mengambil nomor dari kontak ponsel.

4.2. Implementasi

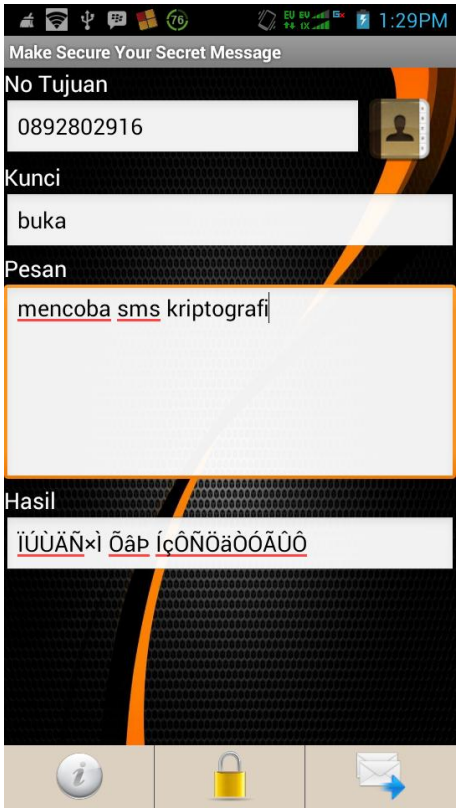
Pada saat aplikasi enkripsi SMS ini dijalankan, maka akan muncul tampilan awal berupa menu utama. Dalam menu utama ini terdapat 3 buah menu yaitu : menu tulis pesan, menu inbox, dan menu about.

4.2.1. Form Menu Utama

Gambar 4.1 Menu Utama

Gambar 4.1 diatas merupakan tampilan menu utama dari aplikasi enkripsi SMS dimana terdapat tiga menu utama yaitu menu Tulis Pesan, Inbox, dan About. Untuk memulai aplikasi ini maka user diminta untuk memilih salah satu dari tiga menu pilihan diatas.

4.2.2. Pengujian Menu Tulis Pesan



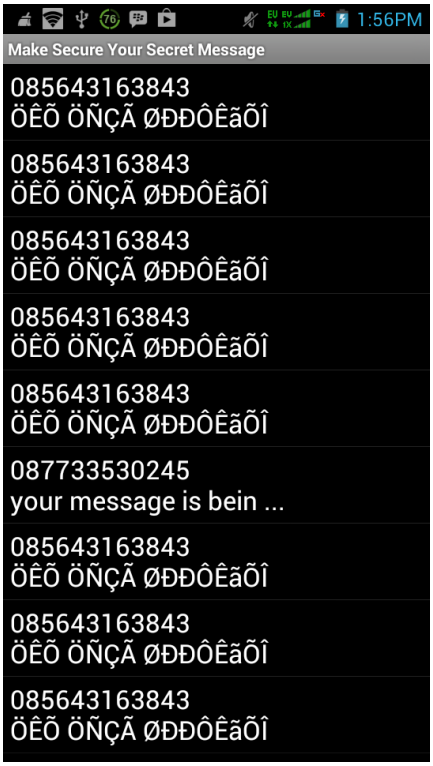
Gambar 4.2 Pengujian Menu Tulis Pesan

Gambar 4.2 diatas merupakan tampilan ketika menu tulis pesan dibuka, tombol phonebook disamping isian nomor itu merukpakan tombol untuk mengambil nomor yang sudah tersimpan pada system telepon pengguna. Pengguna kemudian memasukkan kunci lalu mengetikkan isi pesan dan kemudian menekan tombol enkripsi berbentuk kunci maka hasilnya akan muncul di kolom hasil, setelah itu menekan tombol kirim untuk mengirimkan pesan. Pada gambar diatas proses enkripsi

dari pesan teks "mencoba sms kriptografi " dengan kunci "aku" menghasilkan teks " ÏÚÙÄÑ×Ì Õâþ ÍçÔÑÖäÒÓÃÛÔ"

yang kemudian akan dikirim ke nomor yang dituju.

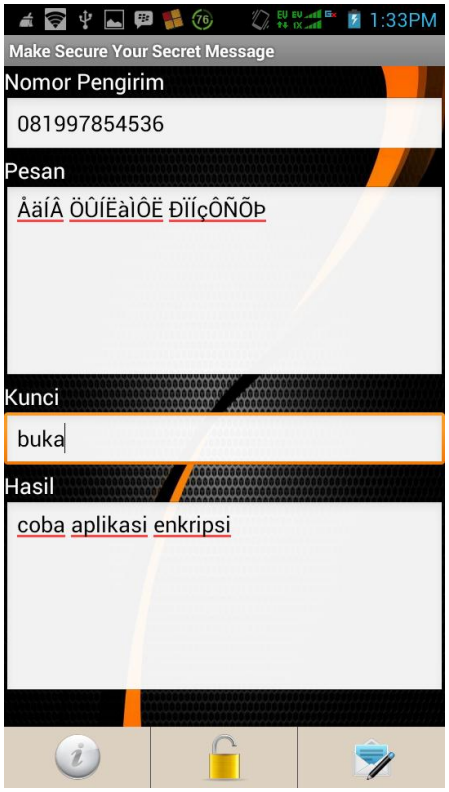
4.2.3. Tampilan menu Inbox



Gambar 4.3 Tampilan Inbox SMS

Gambar 4.3 diatas merupakan tampilan ketika menu inbox dibuka, akan muncul list pesan masuk, danuntuk membacanya klik pada list pesan tersebut.

4.2.4. Pengujian Baca Pesan

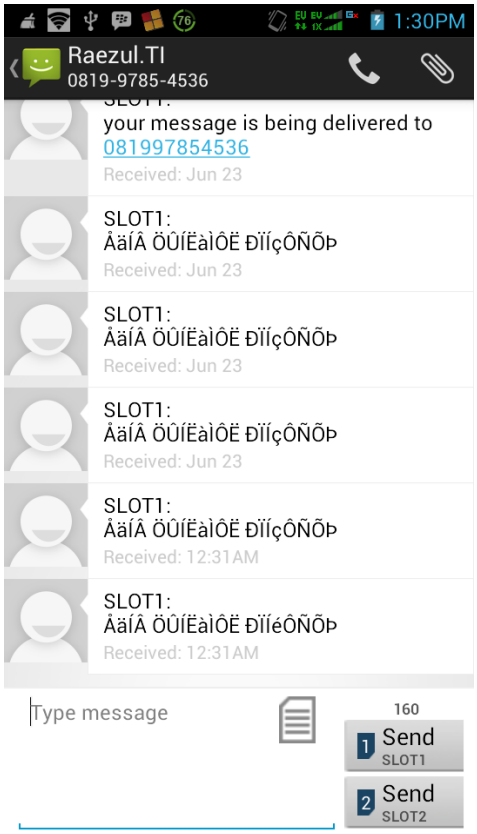


Gambar 4.4 Tampilan Baca Pesan Pada Aplikasi Enkripsi

Gambar 4.4 diatas merupakan tampilan ketika kita memilih pesan dari list pesan untuk membacanya. Pada saat kita membuka pesan ini kita memerlikan kunci yang sesuai saat proses enkripsi agar dapat membacanya. Pada pesan diatas dengen teks ” ÅäíÂ ÖÛÍËàÌÔË ÐÏÍçÔÑÕÞ “ kita masukkan kunci yang sesuai yaitu kata “buka” setelah ditekan tombol dekrip maka pesan tersandi tersebut dapat dibaca yaitu kata

“ coba aplikasi enkripsi “.

4.2.5. Tampilan Pesan Ketika Dibuka Dengan SMS Standard



Gambar 4.5. Tampilan Baca Pesan Pada SMS standard

Gambar 4.5 diatas merupakan tampilan ketika kita pesan dibuka dengan SMS *standard*, pesan tersebut menjadi tidak memiliki makna seperti terlihat pada gambar diatas.