

Penerapan Metode Simple Columnar Transposition (SCTR) untuk Aplikasi Enkripsi Teks Berbasis Android

Bramantara Yudha

Teknik Informatika, Universitas Indraprasta PGRI, Jakarta, Indonesia
e-mail: bram.proklamatorz@gmail.com

Submitted Date: February 25th, 2021
Revised Date: June 04th, 2021

Reviewed Date: June 02nd, 2021
Accepted Date: June 15th, 2021

Abstract

The development of communication technology such as the use of mobile phones based on android. Android has the capability of getting good as personal computer and it has a large storage area. Android users use it to store important data. Critical data that is stored to make various ways to membobolnya. The role of cryptography is indispensable for securing the data. Cryptography is the science to encode a message, while to keep the message or document is secure and cannot be read by those who are not entitled to use encryption. The application of encryption uses a symmetric algorithm. On the application of this method used is Simple Columnar Transposition method. Simple Columnar Transposition method by dividing plainteks into blocks with a key length (k) certain then these blocks are arranged in the form of rows and columns. The making of this application using the JAVA programming language and XML. The application is made for a mobile with Android using the Studio. This application was eventually shown on android mobile phone users. The application used to encode text/sentence, the result of encryption in the form of chipertext, and the result of decrypting plaintext form.

Keywords: Android; Encryption; Cryptography; Simple Columnar Transposition (SCTR)

Abstrak

Perkembangan teknologi komunikasi sangat pesat seperti penggunaan telepon genggam yang berbasis android. Android memiliki kemampuan semakin bagus layaknya personal computer dan memiliki tempat penyimpanan yang besar. Pengguna android menggunakannya untuk menyimpan data yang penting. Data penting yang disimpan membuat berbagai cara untuk membobolnya. Peran kriptografi sangat diperlukan untuk mengamankan data tersebut. Kriptografi merupakan suatu teknik menyandikan pesan dan enkripsi adalah cara menjaga dokumen atau pesan menjadi lebih karena tidak bisa terbaca oleh orang yang tidak berhak. Aplikasi enkripsi ini menggunakan algoritma simetris. Aplikasi yang dikembangkan dalam penelitian ini menggunakan metode Simple Columnar Transposition. Cara kerja metode ini membagi plainteks ke dalam bentuk blok-blok kunci (k). Blok-blok selanjutnya disusun menjadi baris dan kolom. Pembuatan aplikasi ini menggunakan pemrograman JAVA dan XML. Pada aplikasi ini dibuat untuk mobile dengan menggunakan Android Studio. Aplikasi dapat berguna bagi pengguna dalam menyembunyikan data secara rahasia khusus bagi pengguna handphone android. Pengguna Aplikasi ini dapat menyembunyikan data dengan menyandikan teks/kalimat kemudian dienkripsi dari data chipertext dirubah dekripsi berupa plaintext

Kata Kunci: Android; Enkripsi; Kriptografi; Simple Columnar Transposition (SCTR).

1. Pendahuluan

Perkembangan teknologi komunikasi sangat pesat, seperti penggunaan telepon genggam yang berbasis android dan IOS. Android memiliki banyak kegunaan seperti mengambil dokumen, video, foto, suara, menyimpan data, dan bentuk data asli. Android

memiliki kemampuan yang semakin membaik dikarenakan kapasitas tempat penyimpanan yang besar (Enterprise, 2015), sehingga membuat banyak pengguna yang menggunakan android dapat menyimpan data yang bersifat rahasia.

Banyaknya masalah yang terjadi di saat pengiriman dan penerimaan data seperti

penyadapan, pencurian, dan pemalsuan data membuat keamanan data menjadi penting (Muttaqin, 2010). Peran kriptografi sangat dibutuhkan untuk menjaga keamanan data tersebut (Efrandi, 2014). Beberapa teknik pengamanan data selain kriptografi yaitu enkripsi, firewall, Secure Socket Layer (SSL), dan lain-lain.

Kriptografi merupakan suatu teknik menyandikan pesan dan enkripsi adalah cara menjaga dokumen atau pesan menjadi lebih karena tidak bisa terbaca oleh orang yang tidak berhak (Mukhtar, 2018). Kriptografi sudah dikenal dan dipakai kurang lebih tahun 1900 sebelum masehi (Sadikin, 2012). Pada kriptografi terdapat beberapa macam algoritma seperti simetris, asimetris, dan hash. Pada penelitian ini algoritma yang saya gunakan adalah simetris dengan metode Simple Columnar Transposition (SCTR). Metode Simple Columnar Transposition ini termasuk dalam beberapa metode kriptografi klasik. Metode kriptografi klasik adalah jenis keamanan data yang pertama kali digunakan, oleh karena itu metode ini adalah konsep dasar dari kriptografi.

Cara kerja metode ini membagi plaintext ke dalam bentuk blok-blok kunci (k). Blok-blok selanjutnya disusun menjadi baris dan kolom. Peneliti membuat penelitian ini untuk mewujudkan implementasi keamanan data menerapkan metode enkripsi Simple Columnar Transposition ke dalam aplikasi yang mudah digunakan.

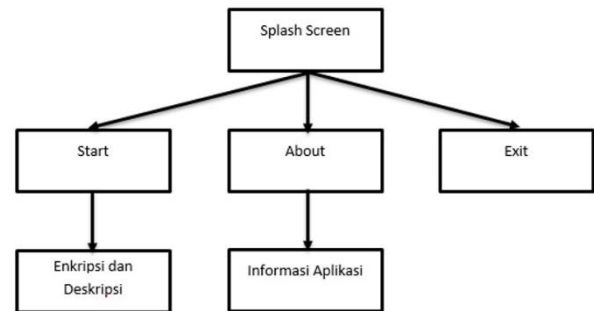
Untuk memastikan bahwa aplikasi yang dikembangkan telah sesuai dengan persyaratan yang telah ditetapkan, maka akan dilakukan pengujian. Pengujian digunakan untuk menemukan galat (cacat) yang disebabkan oleh perbedaan antara hasil yang diharapkan dengan hasil sebenarnya (Pratama, Ristianto, Prayogo, Nasrullah, & Saifudin, 2020). Pengujian merupakan tahapan penting yang harus dilakukan untuk memberikan jaminan terhadap kualitas perangkat lunak yang dikembangkan (Muslimin, et al., 2020).

2. Perancangan dan Penerapan

Pada saat ini ada beberapa cara untuk melakukan enkripsi data, seperti penggunaan metode SCTR ini untuk menyandikan teks. Peneliti melakukan enkripsi teks yang bisa dijalankan menggunakan sistem Android (H, 2012). Pada pembuatan enkripsi ini, kita membutuhkan struktur navigasi untuk

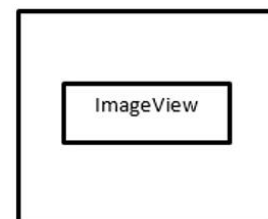
menjelaskan alur aplikasi yang dibuat sehingga lebih terurut.

Pada pembuatan aplikasi akan digunakan bahasa java. Bahasa java dipilih karena merupakan bahasa berorientasi obyek yang banyak digunakan dalam pengembangan aplikasi Android (Sianipar, 2013). Java merupakan bahasa pemrograman multiplatform dan mudah dipahami untuk pengembangan aplikasi (Warno, 2012).



Gambar 1. Struktur Navigasi Aplikasi

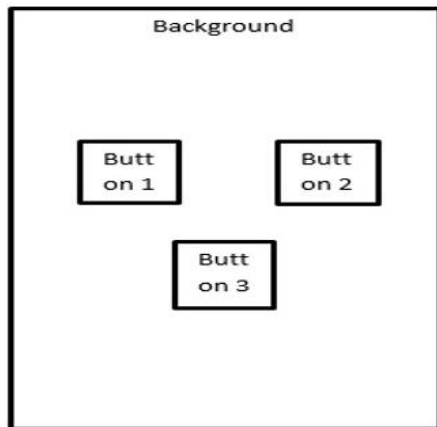
Pada gambar 1 dijelaskan struktur navigasi dari perangkat lunak. Struktur navigasi ini untuk memudahkan pengguna memilih perintah yang digunakan pada aplikasi.



Gambar 2. Halaman Splash

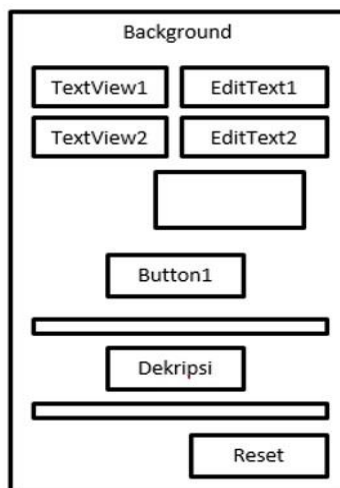
Halaman splash screen pada gambar 2 merupakan halaman awal yang digunakan untuk menampilkan informasi sesaat pada user. Splash Screen juga berfungsi sebagai pemanis tampilan aplikasi. Pada halaman ini terdapat ImageView dan Progress Bar dengan durasi tampilan beberapa detik. Rancangan halaman splash screen seperti pada gambar 2.

Halaman menu ditampilkan setelah proses splashscreen sebelumnya. Pada halaman ini terdapat tiga buah tombol yaitu berupa tombol mulai, tombol tentang dan tombol keluar. Rancangan halaman menu seperti gambar 3.



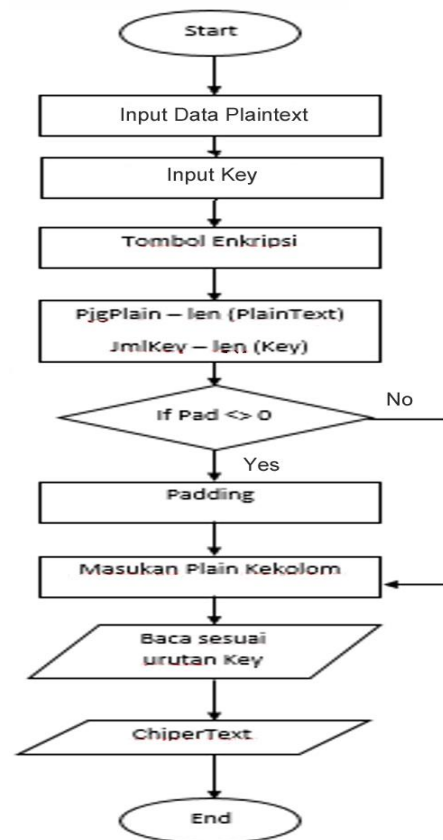
Gambar 3. Halaman Menu

Pada halaman enkripsi dan dekripsi ditampilkan setelah pengguna mengklik button mulai pada menu utama. Halaman ini berisi tentang enkripsi dan dekripsi, terdapat karakter yang akan dienkripsi atau didekripsi, key yang dipakai pada proses enkripsi dan dekripsi, tampilan matriks, beserta reset dari field yang ada, dan terdapat tampilan hasil dari kalimat yang telah terenkripsi dan terdekripsi. Berikut adalah tampilan dari halaman Enkripsi dan Dekripsi seperti pada rancangan tampilan pada gambar 4.



Gambar 4. Halaman Menu

Flowchart Diagram digunakan sebagai penggambaran dalam bentuk grafik menjelaskan tahapan yang terurut dari prosedur dari suatu program (Wulandari, 2017). Kita bisa membuat flowchart diagram ini menggunakan aplikasi Star UML.



Gambar 5. Flowchart Aplikasi

Pada Gambar 5 merupakan gambar Flowchart untuk membantu analis dan programmer menyelesaikan masalah kedalam bagian yang lebih kecil dan menolong dalam menganalisis alternatif pada proses operasi. Pada hal berikut ini kita akan membuat aplikasinya dengan menggunakan Android Studio. Pembuatan ini peneliti tidak memasukkan semua kode sumbernya, dan hanya menuliskan secara detail pada bagian pembuatan halaman enkripsi dan deskripsi. Halaman splashscreen dibuat dengan komponen ImageView yang berada ditengah halaman. Potongan program desain layout pada splashscreen.xml sebagai berikut

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:gravity="center"
android:background="#9e9a9a"
android:orientation="vertical"
android:weightSum="1">
```

```
<ImageView
    android:layout_gravity="center"
    android:layout_margin="70dp"
    android:layout_width="283dp"
    android:layout_height="254dp"

    android:src="@drawable/privateicon" />
```

Penggunaan sourcode untuk program java pada splash screen Activity.java sebagai berikut

```
package com.example.venom.enkripsi;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
```

Import pada source code java berfungsi memakai *method-method* dari *class/library* lain, *method* selanjutnya dipanggil dan digunakan oleh *class*.

Pada halaman utama untuk membuat *button-buttonnya* dibutuhkan *source code*. Berikut adalah *source code* (.XML).

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.example.venom.enkripsi.MainActivity"
    android:background="#ffffff"
    android:gravity="center">
```

Pada pembuatan halaman enkripsi dibutuhkan *button textview* dan *edittext*, berikut adalah tampilan

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android
```

```
.com/tools"
```

```
tools:context="com.venom.enkripsi2.
AplikasiActivity"
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:background="#ffffff">
<LinearLayout
```

```
android:layout_width="match_parent"
```

```
android:layout_height="match_parent"
```

```
android:orientation="vertical"
    android:weightSum="1">
```

Memberikan background pada halaman enkripsi dan dekripsi dengan source android:background. Pada source diatas jenis lebar dan tinggi layout menggunakan match_parent agar tempat masukan atau keluaran yang disediakan tidak berubah sesuai panjang dan lebar kata yang dimasukkan.

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:weightSum="5">
```

Memperjelas kegunaan *edittext* digunakan *imageview*. Source yang digunakan android:src.

```
<ImageView
    android:layout_width="112dp"
    android:layout_height="match_parent"
    android:layout_marginTop="20dp"
    android:id="@+id/imageView"
    android:src="@drawable/pt2"/>
```

Tempat untuk memasukkan karakter (*plaintext*) yang akan dienkripsi menggunakan *edittext*.

```
<EditText
    android:id="@+id/input1"
    android:layout_marginRight="20dp"
    android:layout_marginTop="20dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    />
```

Memperjelas kegunaan *edittext* digunakan *imageview*. *Source* yang digunakan `android:src`

```
<ImageView  
android:layout_width="113dp"  
android:layout_height="match_parent"  
android:id="@+id/imageView2"  
android:layout_marginTop="15dp"  
android:src="@drawable/key"  
android:layout_weight="0.72" />
```

Tempat untuk memasukan karakter (*kunci*) yang akan di enkripsi menggunakan *edittext* dan aplikasi ini saya batasi 6 karakter untuk pengisian kunci.

```
<EditText  
android:id="@+id/input2"  
android:layout_marginRight="20dp"  
android:layout_marginTop="15dp"  
android:layout_width="250dp"  
android:maxLength="6"  
android:layout_height="wrap_content"  
/>  
</LinearLayout>
```

Saat memasukkan *plaintext* dan *key*, untuk mendapatkan hasil enkripsi tekan tombol enkripsi. Pada *button* digunakan *background* untuk memberikan inisial bahwa *button* tersebut ialah *button* enkripsi. Memberikan nama pada *button* sesuai kegunaan pada *button* dengan `source android: text`. Enkripsi akan masuk pada field *edittext* secara otomatis setelah *button* diklik.

```
<ImageButton  
android:layout_width="100dp"  
android:layout_height="45dp"  
android:id="@+id/en"  
android:layout_gravity="center_hori  
zontal"  
android:layout_marginTop="30dp"  
android:src="@drawable/el2" />
```

```
<EditText  
android:editable="false"  
android:id="@+id/input3"  
android:layout_width="match_parent"  
android:background="#ffffff"  
android:layout_height="wrap_content"  
android:layout_margin="20dp"/>
```

Saat memasukan *plaintext* dan *key*, untuk mendapatkan hasil dekripsi tekan tombol dekripsi. Pada *button* digunakan *background* untuk memberikan inisial bahwa *button* tersebut

ialah *button* dekripsi. Pemberian nama pada *button* sesuai kegunaan pada *button* dengan `source android: text`. Dekripsi akan masuk pada field *edittext* secara otomatis setelah *button* diklik maka hasil.

```
<ImageButton  
android:layout_width="wrap_content"  
android:layout_height="wrap_conten"  
android:id="@+id/de"  
android:layout_gravity="center_hori  
zontal"  
android:src="@drawable/dl2"  
android:background="#ffffff"  
android:adjustViewBounds="true" />
```

```
<EditText  
android:editable="false"  
android:id="@+id/input4"  
android:layout_width="match_parent"  
android:textColor="#000000"  
android:background="#ffffff"  
android:layout_height="wrap_conten"  
android:layout_margin="20dp"/>
```

Pembuatan *layout* halaman enkripsi pada XML, selanjutnya membuat pada *AplikasiActivity.java* dengan `source` sebagai berikut:

```
import java.util.Arrays;  
public class AplikasiActivity  
extends AppCompatActivity {  
private Button reset, en, de;  
private EditText input,kunci,enkrip,  
dekrip;  
private TextView cara;  
private String encryptedMessage;  
private String enkripMatriks;
```

Pada program ini peneliti menggunakan bantuan array sehingga pada awal `source` dituliskan `import java.util.Arrays`. Pada `source` di atas berfungsi untuk mendeklarasikan *tools* yang sudah digunakan pada XML, seperti *textview*, *edittext* dan *button*.

```
@Override  
protected void onCreate(Bundle  
savedInstanceState) {  
super.onCreate(savedInstanceState);  
setContentView(R.layout.activity_ap  
likasi);  
reset = (Button)  
findViewById(R.id.reset);  
en = (Button)  
findViewById(R.id.en);
```

```
de = (Button)
findViewById(R.id.de);
input = (EditText)
findViewById(R.id.input1);
cara = (TextView)
findViewById(R.id.proses);
kunci = (EditText)
findViewById(R.id.input2);
enkrip = (EditText)
findViewById(R.id.input3);
dekrip = (EditText)
findViewById(R.id.input4);
```

Penjabaran *tools* di atas berdasarkan *id* masing - masing, selanjutnya ialah membuat *tools* sesuai dengan fungsi masing - masing.

```
reset.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
input.setText("");
kunci.setText("");
enkrip.setText("");
dekrip.setText("");
}
});
```

Pada *button reset* untuk mengosongkan seluruh *field* yang telah terisi, maka digunakan *source* seperti di atas.

```
en.setOnClickListener(new
View.OnClickListener() {
@Override
public void onClick(View v) {
if
(input.getText().toString().trim().
equals("")) ||
kunci.getText().toString().tr
im().equals("")) {
Toast.makeText(AplikasiActivity.thi
s, "Anda harus Memasukan
Data Terlebih Dahulu",
Toast.LENGTH_SHORT).show();
}
else {
final char
abjad[] = {' ', '0', '1', '2', '3',
'4', '5', '6', '7', '8', '9',
'a', 'b', 'c', 'd', 'e', 'f', 'g',
'h', 'i', 'j', 'k', 'l', 'm', 'n',
'o', 'p', 'q', 'r', 's', 't', 'u',
'v', 'w', 'x', 'y', 'z', 'A',
'B', 'C', 'D', 'E', 'F', 'G', 'H',
'I', 'J', 'K', 'L', 'M', 'N',
'O', 'P', 'Q', 'R', 'S', 'T', 'U',
'V', 'W', 'X', 'Y', 'Z'};
```

Pada *source* di atas, setelah menekan tombol *button enkripsi* maka akan muncul hasil enkripsi pada *edittext* yang telah disediakan. Pada *source code* diatas dijelaskan bahwa jika pesan atau kunci masih kosong maka tidak bisa memproses hasil. *Source code* di atas terdapat urutan angka dan huruf yang dapat dipakai saat memasukan pesan ataupun kunci.

```
String sentence =
input.getText().toString().replace(
" ", "*");
String key =
kunci.getText().toString();

char keyChar[] = key.toCharArray();
char sentenceChar[] =
sentence.toCharArray();
int numSentence = sentence.length();
int kolom = key.length();
int baris = numSentence / kolom;
```

Pada *source* di atas dijelaskan bahwa untuk memasukan spasi, maka keluarannya akan berubah menjadi tanda “ * “, dengan bantuan *replace* dan dijelaskan masukan pesan dan kunci akan di rubah menjadi sebuah array, pesan maupun kunci akan dihitung panjang karakternya dan untuk membuat baris maka digunakan rumus panjang pesan dibagi panjang kunci.

```
if (numSentence % kolom > 0) baris++;
enkripMatriks = "";
char matriks[][] = new
char[baris + 1][kolom];
for (int i = 0; i < kolom; i++)
matriks[0][i] = keyChar[i];
for (int i = 1; i < baris;
i++) {
for (int j = 0; j <
kolom; j++) {
if (((i - 1) * kolom) +
j - 1 > numSentence) {
matriks[i][j] =
sentenceChar[((i - 1) * kolom)
+ j];
}
}
}
```

Jika hasil modulo dari penjumlahan baris lebih besar dari 0 maka baris akan bertambah 1, setelah itu maka akan mulai terbentuk matrik dari baris dan kolom.

```
for (int i = 1; i < baris + 1; i++)  
{  
for (int j = 0; j < kolom; j++) {  
if (((i - 1) * kolom) + j <  
numSentence) {  
matriks[i][j] =  
sentenceChar[((i - 1) * kolom) + j];  
enkripMatriks += matriks[i][j];  
} else {  
continue;  
}  
if (j == kolom-1) {  
enkripMatriks += "\n";  
}  
else {  
enkripMatriks += " ";  
}  
}  
}  
Log.d("DEBUG_", "Pasang matriks : "  
+enkripMatriks);  
cara.setText(enkripMatriks);
```

Pada source di atas untuk *i* sama dengan 1 dan *i* lebih kecil dari baris ditambah 1 maka *i* akan bertambah 1, dan matrik pun akan terbentuk kembali.

Pada source tersebut terdapat *continue* yang berfungsi untuk melanjutkan program yang berfungsi untuk mengatur pembuatan matriks yang ada pada aplikasi dan debug pada source diatas dimaksudkan untuk mempermudah melihat hasil pembuatan matriks.

```
boolean udah[] = new  
boolean[kolom];  
for (int i = 0; i <  
kolom; i++) udah[i] = false;  
encryptedMessage = "";
```

Pada source diatas terdapat *new boolean* untuk membuat object baru, berfungsi untuk meminimalisir huruf yang sama dalam satu baris.

```
for (int i = 0; i < 63; i++) {  
for (int j = 0; j < kolom; j++) {  
if (udah[j]) continue;  
else {  
if (matriks[0][j] ==  
abjad[i]) {  
udah[j] = true;
```

```
for (int k = 1; k <  
baris + 1; k++) {  
encryptedMessage +=  
matriks[k][j];  
}  
} else {  
continue;  
}  
}  
}
```

Pada source diatas dapat dilihat bahwa *i* dibatasi hanya 63, karena karakter yang saya masukan hanya 63. Pada source di atas merupakan tahapan merupakan proses percabangan untuk objek baru yang pada source sebelumnya telah dijelaskan.

```
enkrip.setText(encryptedMessage);  
}  
});
```

Pada source diatas berfungsi untuk memasukan hasil enkripsi ke dalam edit text yang telah disediakan.

```
de.setOnClickListener(new  
View.OnClickListener() {  
@Override  
public void onClick(View v) {  
if(enkrip.getText().toString().trim  
().equals("")) {  
Toast.makeText(AplikasiActivity.thi  
s, "Anda harus Enkrip terlebih  
dahulu", Toast.LENGTH_SHORT).show();  
}  
else  
dekrip.setText(input.getText().toSt  
ring());  
}  
});
```

Pada source di atas, setelah menekan tombol button dekripsi maka akan muncul hasil dekripsi pada edittext yang telah disediakan. Pada source code di atas dijelaskan bahwa jika pesan atau kunci masih kosong maka tidak bisa memproses hasil.

3. Hasil Perancangan

Kita bisa melihat hasil tampilan yang telah peneliti buat seperti gambar dibawah ini.



Gambar 6. Tampilan Hasil

4. Kesimpulan

Penelitian ini berhasil menghasilkan aplikasi enkripsi menggunakan algoritma simetris dengan metode Simple Columnar Transposition pada sistem operasi android. Aplikasi ini dapat memberikan pengetahuan tentang cara kerja metode enkripsi simetris. Aplikasi ini juga telah berjalan dan berproses dengan baik pada device.

Hasil uji coba pada android digunakan 3 tipe android yang pertama menggunakan Asus Zenfone 2, Asus Zenfone 4 dan Samsung Duos, untuk ketiga tipe android tersebut aplikasi enkripsi berjalan sesuai dengan yang diharapkan, untuk splashscreen, menu utama dan halaman enkripsi juga sesuai dengan metode yang digunakan. Berikut ini adalah hasil dari penelitian berupa aplikasi penyandian data:

- Penelitian ini berhasil menghasilkan aplikasi enkripsi menggunakan algoritma simetris dengan metode Simple Columnar Transposition pada sistem operasi android.
- Aplikasi ini dapat memberikan pengetahuan tentang cara kerja metode enkripsi simetris. Aplikasi ini juga telah berjalan dan berproses dengan baik pada device.
- Hasil uji coba pada android digunakan 3 tipe android yang pertama menggunakan Asus Zenfone 2, Asus Zenfone 4 dan Samsung Duos, untuk ketiga tipe android tersebut aplikasi enkripsi berjalan sesuai dengan yang diharapkan, untuk splashscreen, menu utama dan halaman enkripsi juga sesuai dengan metode yang digunakan.

Referensi

- Efrandi, A. &. (2014). Aplikasi Kriptografi Pesan Menggunakan Algoritma Vigenere Cipher. *Jurnal Media Infotama*, 120-128, ISSN 1858 – 2680, Vol.2.
- Enterprise, J. (2015). *Mengenal Dasar-Dasar Pemrograman Android*. Jakarta: PT Elex Media Komputindo.
- H, N. S. (2012). *Pemrograman Aplikasi Mobile Smartphone dan Table PC Berbasis Android*. Bandung: Informatika.
- Mukhtar, H. (2018). *Kriptografi Untuk Keamanan Data*. Sleman: CV Budi Utama.
- Muslimin, D. B., Kusmanto, D., Amilia, K. F., Ariffin, M. S., Mardiana, S., & Yulianti, Y. (2020). Pengujian Black Box pada Aplikasi Sistem Informasi Akademik Menggunakan Teknik Equivalence Partitioning. *Jurnal Informatika Universitas Pamulang*, 5(1), 19-25. doi:10.32493/informatika.v5i1.3778
- Muttaqin, Z. (2010). *Pembuatan Aplikasi Enkripsi Menggunakan Metode Advance Encryption Standard dan Rivest Shamir Adleman*. Jakarta: UIN Syarif Hidayatullah.
- Pratama, B. P., Ristianto, I. B., Prayogo, I. A., Nasrullah, & Saifudin, A. (2020). Pengujian Perangkat Lunak Sistem Informasi Penilaian Mahasiswa dengan Teknik Boundary Value Analysis Menggunakan Metode Black Box Testing. *Journal Of Artificial Intelligence And Innovative Applications*, 32-36.
- Sadikin, R. (2012). *Kriptografi Untuk Keamanan Jaringan*. Yogyakarta: CV ANDI OFFSET.
- Sianipar. (2013). *Teori dan Implementasi Java*. Bandung: Informatika.
- Warno. (2012). Pembelajaran Pemrograman Bahasa Java dan Arti Keyword. *Jurnal, Universitas Indraprasta PGRI, Jakarta Selatan*, 65-70.
- Wulandari, T. (2017, Maret 04). *Simbol – Simbol Flowchart Standard yang Dikeluarkan ANSI*. Retrieved from <http://www.academia.edu/http://www.academia.edu/4686454/Simbol>