

DAFTAR PUSTAKA

- Al Fikri, I. (2016). Aplikasi Navigasi Berbasis Perangkat Bergerak dengan Menggunakan Platform Wikitude untuk Studi Kasus Lingkungan ITS. *Jurnal Teknik ITS*, 5(1), 48–51. <https://doi.org/10.12962/j23373539.v5i1.14511>
- Andriyani, S. (2016). Aplikasi Akademik Online Berbasis Mobile Android. *Jurnal Sains Dan Teknologi Utama, Volume XI, Nomor 1, April 2016*, XI(152), 15–26.
- Basim, Z., & Painem, P. (2020). Implementasi Kriptografi Algoritma RC4 Dan 3DES dan Steganografi Dengan Algoritma EOF Untuk Keamanan Data Berbasis Desktop Pada SMK As-Su'udiyah. *Skatika*, 3(4), 45–52. <http://jom.fti.budiluhur.ac.id/index.php/SKANIKA/article/view/1739>
- Djuwitaningrum, E. R., & Apriyani, M. (2016). Teknik Steganografi Pesan Teks Menggunakan Metode Least Significant Bit dan Algoritma Linear Congruential Generator (Text Message Steganography Using Least Significant Bit Method and Linear Congruential Generator Algorithm). *Juuta*, IV(2), 79–85.
- Hardiansyah, B., Armin, A. P., & Yunanda, A. B. (2019). Rekonstruksi Citra Pada Super Resolusi Menggunakan Interpolasi Bicubic. *INTEGER: Journal of Information Technology*, 4(2), 1–12. <https://doi.org/10.31284/j.integer.2019.v4i2.684>
- Hasugian, A. H., & Kom, M. (2017). Perancangan Perangkat Lunak Pengenkripsian Citra BMP , GIF dan JPG dengan Menggunakan Metode HIL L. *Jurnal Ilmu Komputer Dan Informatika*, 01(November), 1–11.
- Ikromina, F. I., & Ujianto, E. I. H. (2019). Invisible Watermarking Citra Digital Menggunakan Kombinasi Metode Discrete Cosine Transform Dan Discrete Wavelet Transform. *JANAPATI: Jurnal Nasional Pendidikan Teknik Informatika*, 8, 261–271.
- Imam Adli, HarunMukhtar, J. A. A. (2018). Perancangan dan pembuatan visual novel sejarah kh. ahmad dahlan sebagai media pembelajaran berbasis android. *RABIT*

- (*Jurnal Teknologi Dan Sistem Informasi Univrab*), 3(2), 69–82.
- Maiyana, E. (2018). Pemanfaatan Android Dalam Perancangan Aplikasi Kumpulan Doa. *Jurnal Sains Dan Informatika*, 4(1), 54–65. <https://doi.org/10.22216/jsi.v4i1.3409>
- Megantara, R. A., & Rafrastara, F. A. (2019). Super Enkripsi Teks Kriptografi menggunakan Algoritma Hill Cipher dan Transposisi Kolom. *Prosiding SENDI_U 2019*, 85–92.
- Nasution, Y. R., & Furqan, M. (2020). Aplikasi Mobile Media Pembelajaran Dasar Algoritma dan Pemrograman Berbasis Android. *Syntax : Journal of Software Engineering, Computer Science and Information Technology*, 1(1), 45–51. <https://doi.org/10.46576/syntax.v1i1.791>
- Nirmala, E. (2020). Penerapan Steganografi File Gambar Menggunakan Metode Least Significant Bit (LSB) dan Algoritma Kriptografi Advanced Encryption Standard (AES) Berbasis Android. *Jurnal Informatika Universitas Pamulang*, 5(1), 36. <https://doi.org/10.32493/informatika.v5i1.4646>
- Pramadana, T. I., Soro, S., & Siswanto, R. D. (2019). Pengembangan Aplikasi Bangun Datar Sederhana (Bandara) Matematika Berbasis Android Pada Materi Bangun Datar Sederhana di Tingkat SMP. *Prosiding Seminar Nasional Teknoka*, 3(2502), 13. <https://doi.org/10.22236/teknoka.v3i0.2894>
- Rachmadsyah, A., Perdana, A., & T, A. B. S. (2020). *Kombinasi Algoritma Beaufort Cipher Dan Vigenere Cipher Untuk Pengamanan Pesan Teks Berbasis Mobile Application*. 9(September), 12–17.
- Rosidin, Sugiantoro, B., & Prayudi, Y. (2018). Analisis Pendeteksi Kecocokan Objek Pada Citra Digital Dengan Metode Algoritma Sift Dan Histogram Color Rgb. *Cyber Security Dan Forensik Digital*, 1(1), 20–27. <https://doi.org/10.14421/csecurity.2018.1.1.1235>
- Santoso, H., & Fakhriza, M. (2018). Perancangan Aplikasi Keamanan File Audio Format Wav (Waveform) Menggunakan Algoritma Rsa. *ALGORITMA: Jurnal Ilmu Komputer Dan Informatika*, 2(1), 47–54.

<http://jurnal.uinsu.ac.id/index.php/algorithm/article/view/1615>

- Sibarani, N. S. (2018). Analisis Performa Aplikasi Native Android Menggunakan Bahasa Pemrograman Java dan Kotlin. *Researchget, December*.
https://www.researchgate.net/publication/329525878_Analisis_Performa_Aplika_si_Android_Pada_Bahasa_Pemrograman_Java_dan_Kotlin
- Suhardi. (2016). Aplikasi Kriptografi Data Sederhana Dengan Metode Exclusive-or (Xor). *Jurnal Teknovasi, 03(2)*, 23–31.
- Syahputra, R. E. (2019). PERANCANGAN APLIKASI PERBANDINGAN DETEKSI TEPI DALAM CITRA DIGITAL DENGAN METODE EDGE DETECTION LINKING DAN SOBEL. *Informatika, Jurnal Pelita, 18(2301–9425)*, 62–68.
- Syawal, M. F., Fikriansyah, D. C., & Agani, N. (2016). Implementasi Teknik Steganografi Menggunakan Algoritma Vigenere Cipher Dan Metode LSB. *Jurnal TICOM, 4(3)*, 91–99.
- Verawati, & Liksha, P. D. (2018). Aplikasi Akuntansi Pengolahan Data Jasa Service Pada Pt. Budi Berlian Motor Lampung. *Jurnal Sistem Informasi Akuntansi (JUSITA), 1(1)*, 1–14.

LAMPIRAN

1. Listing Program

LSB2bit.java

```
package keamanan.pesan.alg;

import java.util.Vector;
import keamanan.pesan.handler.ProgressHandler;

import android.util.Log;

public class LSB2bit {

    private static int[] binary = { 16, 8, 0 };
    private static byte[] andByte = { (byte) 0xC0, 0x30, 0x0C, 0x03 };
    private static int[] toShift = { 6, 4, 2, 0 };
    public static String END_MESSAGE_COSTANT = "#!@";
    public static String START_MESSAGE_COSTANT = "@!#";

    public static byte[] encodeMessage(int[] oneDPix, int imgCols, int imgRows, String str, ProgressHandler hand) {
        str = START_MESSAGE_COSTANT + str;
        byte[] msg = str.getBytes();
        int channels = 3;
        int shiftIndex = 4;
        //Array.newInstance(Byte.class, imgRows * imgCols * channels);
        byte[] result = new byte[imgRows * imgCols * channels];

        if (hand != null)
            hand.setTotal(imgRows * imgCols * channels);
        int msgIndex = 0;
        int resultIndex = 0;
        boolean msgEnded = false;
        for (int row = 0; row < imgRows; row++) {
            for (int col = 0; col < imgCols; col++) {
                int element = row * imgCols + col;
                byte tmp = 0;
                for (int channelIndex = 0; channelIndex < channels; channelIndex++) {
                    if (!msgEnded) {
                        tmp = (byte) (((oneDPix[element] >> binary[channelIndex
```



```

        builder = builder + str;
        if (builder.length() == START_MESSAGE_COSTANT.length()
            && !START_MESSAGE_COSTANT.equals(builder))
        {
            builder = null;
            break;
        }

        tmp = 0x00;
    }

    if (builder != null)
        builder = builder.substring(START_MESSAGE_COSTANT.length(), builder.length() - END_MESSAGE_COSTANT.length());
    return builder;
}

```

```

public static int[] byteArrayToIntArray(byte[] b) {
    Log.v("Size byte array", b.length+ "");
}

```

```

int size=b.length / 3;
Log.v("Size Int array",size+ "");
System.runFinalization();
System.gc();
Log.v("FreeMemory", Runtime.getRuntime().freeMemory()+ "");
int[] result = new int[size];
int off = 0;
int index = 0;
while (off < b.length) {
    result[index++] = byteArrayToInt(b, off);
    off = off + 3;
}

return result;
}

```

```

public static int byteArrayToInt(byte[] b) {
    return byteArrayToInt(b, 0);
}

```

```

public static int byteArrayToInt(byte[] b, int offset) {
    int value = 0x00000000;
    for (int i = 0; i < 3; i++) {
        int shift = (3 - 1 - i) * 8;
        value |= (b[i + offset] & 0x000000FF) << shift;
    }
}

```

```

        value = value & 0x00FFFFFF;
        return value;
    }
}

```

```

public static byte[] convertArray(int[] array) {
    byte[] newarray = new byte[array.length * 3];

    for (int i = 0; i < array.length; i++) {
        /*
         * newarray[i * 3] = (byte) ((array[i] & 0xFF); newarray[i * 3 + 1]
         * = (byte)((array[i] >> 8)& 0xFF); newarray[i * 3 + 2] =
         * (byte)((array[i] >> 16)& 0xFF);
         */
        newarray[i * 3] = (byte) ((array[i] >> 16) & 0xFF);
        newarray[i * 3 + 1] = (byte) ((array[i] >> 8) & 0xFF);
        newarray[i * 3 + 2] = (byte) ((array[i] & 0xFF);
    }
    return newarray;
}

```

MainActivity.java

```

package keamanan.pesan;

import android.Manifest;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Bundle;
import android.support.v4.app.ActivityCompat;
import android.support.v4.content.ContextCompat;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends Activity {
    //private Context context;

```

```

public static final int PICK_ENCODE = 1;
public static final int PICK_IMAGE = 2;
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    //context = this;
    checkAndRequestPermissions();
    initClickListener();
}

private void initClickListener()
{
    ((Button) findViewById(R.id.buttonEncode))
        .setOnClickListener(new OnClickListener() {
            public void onClick(View v)
            {
                MainActivity.this.startActivity(new Intent(
                    MainActivity.this, EncodeActivity.class));
            }
        });
}

Button buttonDecode = (Button)
findViewById(R.id.btnDecode);
buttonDecode.setOnClickListener(new Button.OnClickListener()
{
    public void onClick(View v)
    {
        Intent photoPickerIntent =
new Intent(Intent.ACTION_PICK);
        photoPickerIntent.setType(
"image/*.png");
        startActivityForResult(photoPickerIntent, PICK_IMAGE);
    }
});

((Button) findViewById(R.id.buttonTentang))
    .setOnClickListener(new OnClickListener() {
        public void onClick(View v)
        {
            MainActivity.this.startActivity(new Intent(
                MainActivity.this, TentangActivity.class));
        }
    });
}

```



```

    }
    e.printStackTrace();
    }
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);

    switch (requestCode) {
        case (PICK_IMAGE) :
            if (resultCode == RESULT_OK) {
                Uri photoUri = intent.getData();
                if (photoUri != null) {
                    try {
                        Intent intent1 = new Intent();
                        intent1.setData(photoUri);
                        intent1.setComponent(new ComponentName(DecodeActivity.class.getPackage().getName(), DecodeActivity.class.getCanonicalName()));
                        startActivity(intent1);
                    } catch (Exception e) {
                        break;
                    }
                }
            }
            default:
                break;
    }
}

private boolean checkAndRequestPermissions() {
    int writeExternalStoragePermission = ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE_EXTERNAL_STORAGE);
    int readExternalStoragePermission = ContextCompat.checkSelfPermission(this, Manifest.permission.READ_EXTERNAL_STORAGE);
    List<String> listPermissionsNeeded = new ArrayList<>();
    if (writeExternalStoragePermission != PackageManager.PERMISSION_GRANTED) {
        listPermissionsNeeded.add(Manifest.permission.WRITE_EXTERNAL_STORAGE);
    }
}

```

```

        if (readExternalStoragePermissi
on != PackageManager.PERMISSION
N_GRANTED) {
            listPermissionsNeeded.add(M
anifest.permission.READ_EXTERN
AL_STORAGE);
        }
        if (!listPermissionsNeeded.isEm
pty()) {
            ActivityCompat.requestPermi
ssions(this, listPermissionsNeeded.to
Array(new String[listPermissionsNee
ded.size()],1);
            return false;
        }
        return true;
    }
}

```

EncodeActivity.java

```

package keamanan.pesan;

import keamanan.pesan.alg.LSB2bit;
import keamanan.pesan.dialog.Mobi
ProgressBar;
import keamanan.pesan.handler.Prog
ressHandler;

import java.io.BufferedReader;
import java.io.File;

```

```

import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.IOException;
import java.io.OutputStream;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterfa
ce;
import android.content.Intent;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFacto
ry;
import android.graphics.Color;
import android.graphics.Bitmap.Conf
ig;
import android.net.Uri;
import android.os.Bundle;
import android.os.Environment;
import android.os.Handler;
import android.provider.MediaStore;
import android.support.v7.app.AppC
ompatActivity;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ImageView;
import android.widget.LinearLayout;

```

```

import android.widget.TextView;
import android.widget.Toast;

public class EncodeActivity extends
AppCompatActivity {

    private Context context;
    private final Handler handler = ne
w Handler();
    private Bitmap sourceBitmap;
    public static final int PICK_IMAG
E = 1;
    public static final int PICK_TEXT
= 2;
    private String absoluteFilePathSou
rce ;
    private MobiProgressBar progress
Bar;
    String FileName;
    String FileExtension;
    File sdcard;
    ImageView imgView;
    LinearLayout linearLayout;

    /** Called when the activity is first
created. */
    @Override
    public void onCreate(Bundle save
dInstanceState) {
        super.onCreate(savedInstanceSt
ate);

        setContentView(R.layout.activit
y_encode);

        context = this;
        sdcard = Environment.getExtern
alStorageDirectory();

        imgView = findViewById(R.id.
imgView);
        linearLayout = findViewById(R
.id.linearLayout);
        linearLayout.setVisibility(View.
GONE);

        TextView buttonSelectImage =
findViewById(R.id.btnPilihGambar);
        buttonSelectImage.setOnClickL
istener(new Button.OnClickListener(
) {

            public void onClick(View v)
{
                Intent photoPickerIntent =
new Intent(Intent.ACTION_PICK);
                photoPickerIntent.setType(
"image/*.png");
                startActivityForResult(phot
oPickerIntent, PICK_IMAGE);
            }
        });
    }
}

```

```

        TextView buttonSelectText = findViewById(R.id.btnPilihText);
        buttonSelectText.setOnClickListener(new Button.OnClickListener()
        {
            public void onClick(View v)
            {
                Intent textPickerIntent = new Intent(Intent.ACTION_GET_CONTENT);
                textPickerIntent.setType("*/*");
                startActivityForResult(textPickerIntent, PICK_TEXT);
            }
        });
        Button buttonEnkripsi = findViewById(R.id.btnEnkripsi);
        buttonEnkripsi.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v)
            {
                EditText etPesan = findViewById(R.id.editPesan);
                EditText etKunci = findViewById(R.id.editKunci);

                String pesan = etPesan.getText().toString().toLowerCase();
                String kunci = etKunci.getText().toString().toLowerCase();

                if (etPesan.length() == 0) {
                    Toast.makeText(getApplicationContext(), "Pilih File Text yang memiliki pesan!", Toast.LENGTH_SHORT).show();
                } else if (etKunci.length() == 0) {
                    Toast.makeText(getApplicationContext(), "Input kunci!", Toast.LENGTH_SHORT).show();
                } else {
                    String bPesan = encrypt(pesan, kunci);
                    etPesan.setText(bPesan);
                }
            }
        });
        Button buttonSisipkan = (Button) findViewById(R.id.btnEncode);
        buttonSisipkan.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View v)
            {

```

```

        closeContextMenu();
        closeOptionsMenu();
        progressBar=new MobiProgress
        progressBar(EncodeActivity.this);
        progressBar.setMax(100);
        progressBar.setMessage(context.getString(R.string.encoding));
        progressBar.show();
        Thread tt = new Thread(new Runnable() {
            public void run() {
                encode();
                handler.post(mShowAlert);
            }
        });
        tt.start();
    }
});
}

final Runnable mShowAlert = new Runnable() {
    public void run() {
        progressBar.dismiss();
        AlertDialog.Builder builder =
        new AlertDialog.Builder(
            context);
        builder.setMessage(context.getText(R.string.saved))
        .setCancelable(false).set
        PositiveButton(
            context.getText(R.string.ok),
            new DialogInterface.OnClickListener() {
                public void onClick(
                    DialogInterface dialog,
                    int id) {
                    //EncodeActivity.this.finish();
                    //imageView.setImageBitmap(sourceBitmap);
                    Uri imgUri=Uri.parse(absoluteFilePathSource);
                    //imageView.setImageURI(null);
                    imageView.setImageURI(imgUri);
                    linearLayout.setVisibility(View.VISIBLE);
                }
            });
        AlertDialog alert = builder.create();
        alert.show();
    }
};

```

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent intent) {
    super.onActivityResult(requestCode, resultCode, intent);

    switch (requestCode) {
        case (PICK_IMAGE):
            if (resultCode == RESULT_OK) {
                linearLayout.setVisibility(View.GONE);
                Uri photoUri = intent.getData();
                if (photoUri != null) {
                    try {
                        EditText etImage = this.findViewById(R.id.editImage);
                        Cursor cursor = getContentResolver().query(photoUri, null, null, null, null);
                        cursor.moveToFirst();

                        int idx = cursor.getColumnIndex(MediaStore.Images.ImageColumns.DATA);
                        absoluteFilePathSource = cursor.getString(idx);
                        etImage.setText(absoluteFilePathSource);

                        BitmapFactory.Options opt = new BitmapFactory.Options();
                        opt.inDither = false;
                        opt.inScaled = false;
                        opt.inDensity = 0;
                        opt.inJustDecodeBounds = false;
                        opt.inPurgeable = false;
                        opt.inSampleSize = 1;
                        opt.inScreenDensity = 0;
                        opt.inTargetDensity = 0;

                        sourceBitmap = BitmapFactory.decodeFile(absoluteFilePathSource, opt);
                    } catch (Exception e) {
                        e.printStackTrace();
                    }
                }
            }
    }
}

```



```

        }
    }
}
break;

case (PICK_TEXT):
    if (resultCode == RESULT
_OK) {
        FileName = intent.getDa
ta().getLastPathSegment().replace("p
rimary:", "");
        Log.e("Filename", FileN
ame);
        FileExtension=FileName
.substring(FileName.lastIndexOf(".")
);
        Log.e("Extension", FileE
xtension);
        EditText etText = this.fi
ndViewById(R.id.editText);
        etText.setText(sdcard +
FileName);
        if (FileExtension.equals(
".sql")){
            //encrypt.setEnabled(t
rue);
        }

        //Get the text file

        File file = new File(sdca
rd, FileName);
        //Read text from file
        StringBuilder text = new
        StringBuilder();
        try {
            BufferedReader br = n
ew BufferedReader(new FileReader(
file));
            String line;
            while ((line = br.read
Line()) != null) {
                text.append(line);
                text.append("\n");
            }
            br.close();
        }
        catch (IOException e) {
            //You'll need to add pr
oper error handling here
        }
        //Find the view by its id
        EditText etPesan = find
ViewById(R.id.editPesan);
        //Set the text
        etPesan.setText(text.toSt
ring());
    }
    break;
}

```

```

    }

    final Runnable mIncrementProgress = new Runnable() {
        public void run() {
            progressBar.incrementProgressBy(1);
        }
    };

    final Runnable mInitializeProgress = new Runnable() {
        public void run() {
            progressBar.setMax(100);
        }
    };

    final Runnable mSetIndeterminate = new Runnable() {
        public void run() {
            progressBar.setMessage(context.getString(R.string.saving));
            progressBar.setIndeterminate(true);
        }
    };

    private Uri encode() {
        Uri result=null;

        EditText pesan = (EditText) findViewById(R.id.editPesan);
        String s = pesan.getText().toString();

        int width = sourceBitmap.getWidth();
        int height = sourceBitmap.getHeight();

        int[] oneD = new int[width * height];
        sourceBitmap.getPixels(oneD, 0, width, 0, 0, width, height);
        int density=sourceBitmap.getDensity();
        sourceBitmap.recycle();
        byte[] byteImage = LSB2bit.encodeMessage(oneD, width, height, s, new ProgressHandler() {
            private int mysize;
            private int actualSize;

            public void increment(final int inc) {
                actualSize+=inc;
                if(actualSize%mysize==0)
                    handler.post(mIncrementProgress);
            }
        });
    }

```

```

        public void setTotal(fina
l int tot) {
            mysize=tot/50;
            handler.post(mInitiali
zeProgress);
        }

        public void finished() {
        }
    });
    oneD=null;
    sourceBitmap=null;
    int[] oneDMod = LSB2bit.byte
ArrayToIntArray(byteImage);
    byteImage=null;
    Log.v("Encode", "" + oneDMod
[0]);
    Log.v("Encode Alpha", "" + (on
eDMod[0] >> 24 & 0xFF));
    Log.v("Encode Red", "" + (one
DMod[0] >> 16 & 0xFF));
    Log.v("Encode Green", "" + (on
eDMod[0] >> 8 & 0xFF));
    Log.v("Encode Blue", "" + (one
DMod[0] & 0xFF));

    System.gc();
    Log.v("Free memory", Runtime
.getRuntime().freeMemory()+"");

        Log.v("Image mesure", (width*
height*32/8)+"");

        Bitmap destBitmap = Bitmap.cr
eateBitmap(width, height,
        Config.ARGB_8888);

        destBitmap.setDensity(density);
        int partialProgr=height*width/5
0;
        int masterIndex = 0;
        for (int j = 0; j < height; j++)
            for (int i = 0; i < width; i++){
                // The unique way to write
correctly the sourceBitmap, android
bug!!!
                destBitmap.setPixel(i, j, Co
lor.argb(0xFF,
                    oneDMod[masterInde
x] >> 16 & 0xFF,
                    oneDMod[masterInde
x] >> 8 & 0xFF,
                    oneDMod[masterInde
x++] & 0xFF));
                if(masterIndex%partialPro
gr==0)
                    handler.post(mIncremen
tProgress);
            }
        handler.post(mSetInderminate);
    }
}

```

```
Log.v("Encode", "" + destBitmap.getPixel(0, 0));
```

```
Log.v("Encode Alpha", "" + (destBitmap.getPixel(0, 0) >> 24 & 0xFF));
```

```
Log.v("Encode Red", "" + (destBitmap.getPixel(0, 0) >> 16 & 0xFF));
```

```
Log.v("Encode Green", "" + (destBitmap.getPixel(0, 0) >> 8 & 0xFF));
```

```
Log.v("Encode Blue", "" + (destBitmap.getPixel(0, 0) & 0xFF));
```

```
String sdcardState = android.os.Environment.getExternalStorageState();
```

```
String destPath = null;
```

```
int indexSepar=absoluteFilePathSource.lastIndexOf(File.separator);
```

```
int indexPoint=absoluteFilePathSource.lastIndexOf(".");
```

```
if(indexPoint<=1)
```

```
indexPoint=absoluteFilePathSource.length();
```

```
String fileNameDest=absoluteFilePathSource.substring(indexSepar+1, indexPoint);
```

```
fileNameDest+="_nadyah";
```

```
if (sdcardState.contentEquals(android.os.Environment.MEDIA_MOUNTED))
```

```
destPath = android.os.Environment.getExternalStorageDirectory() + File.separator + fileNameDest+".png";
```

```
OutputStream fout = null;
```

```
try {
```

```
Log.v("Path", destPath);
```

```
fout = new FileOutputStream(destPath);
```

```
destBitmap.compress(Bitmap.CompressFormat.PNG, 100, fout);
```

```
//Media.insertImage(getContentResolver(),destPath, fileNameDest, "MobiStego Encoded");
```

```
result=Uri.parse("file://" + destPath);
```

```
sendBroadcast(new Intent(Intent.ACTION_MEDIA_SCANNER_SCAN_FILE, result));
```

```
fout.flush();
```

```
fout.close();
```

```
} catch (IOException e) {
```

```
e.printStackTrace();
```

```
}
```

```
destBitmap.recycle();
```

```

        return result;
    }

    static String encrypt(String text, final String key) {
        String res = "";
        for (int i = 0, j = 0; i < text.length(); i++) {
            int p = text.charAt(i);
            int k = key.charAt(j);
            if (p < 'a' || p > 'z') {
                res += (char) p;
            } else {
                res += (char)((k - p + 26) % 26 + 'a');
                j = ++j % key.length();
            }
        }
        return res;
    }
}

```

DecodeActivity.java

```

package keamanan.pesan;

import keamanan.pesan.alg.LSB2bit;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;

```

```

import android.content.DialogInterface;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.net.Uri;
import android.os.Bundle;
import android.os.Handler;
import android.provider.MediaStore;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class DecodeActivity extends Activity implements Runnable {

```

```

    private Context context;
    private Handler handler;
    private ProgressDialog dd;
    private Uri photoUri;

    EditText pesan, ekstrak;

    private final Runnable runnableDismiss = new Runnable() {

```

```

        public void run() {
            dd.dismiss();
        }
    };

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_decode);
        context = this;

        pesan = findViewById(R.id.editPesan);
        ekstrak = findViewById(R.id.editEkstrak);

        handler = new Handler();
        dd = new ProgressDialog(this);
        dd.setIndeterminate(true);
        dd.setMessage(context.getText(R.string.decoding));
        dd.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        dd.show();
        photoUri = getIntent().getData();

        Thread tt = new Thread(this, "Decoding");

        tt.start();

        Button buttonDecode = findViewById(R.id.btnDecode);
        buttonDecode.setOnClickListener(new Button.OnClickListener() {

            public void onClick(View v) {
                String eks = ekstrak.getText().toString();
                pesan.setText(eks);
            }
        });

        Button buttonDekripsi = findViewById(R.id.btnDekripsi);
        buttonDekripsi.setOnClickListener(new Button.OnClickListener() {

            public void onClick(View v) {
                EditText etPesan = findViewById(R.id.editPesan);
                EditText etKunci = findViewById(R.id.editKunci);
                String pesan = etPesan.getText().toString();
                String kunci = etKunci.getText().toString();
            }
        });
    }
}

```



```

        if (etPesan.length() == 0) {
            Toast.makeText(getApplicationContext(), "Pilih gambar yang
memiliki pesan!", Toast.LENGTH_S
HORT).show();
        } else if (etKunci.length()
== 0) {
            Toast.makeText(getApplicationContext(), "Input kunci!", Toa
st.LENGTH_SHORT).show();
        } else {
            String bPesan = decrypt(
pesan, kunci);
            etPesan.setText(bPesan);
        }
    });
}

public void run() {
    Bitmap image = null;
    try {
        TextView t = (TextView) this
.findViewById(R.id.edit
Image);
        Cursor cursor = getContentRe
solver().query(photoUri, null, null,
        null, null);
        cursor.moveToFirst();

        int idx = cursor
            .getColumnIndex(Media
Store.Images.ImageColumns.DATA)
            ;
        String absoluteFilePath = cur
sor.getString(idx);
        t.setText(absoluteFilePath);

        BitmapFactory.Options opt =
new BitmapFactory.Options();
        opt.inDither = false;
        opt.inScaled = false;
        opt.inDensity = 0;
        opt.inJustDecodeBounds = fa
lse;
        opt.inPurgeable = false;
        opt.inSampleSize = 1;
        opt.inScreenDensity = 0;
        opt.inTargetDensity = 0;
        image = BitmapFactory.deco
deFile(absoluteFilePath, opt);
    } catch (Exception e) {
        e.printStackTrace();
    }

    int[] pixels = new int[image.get
Width() * image.getHeight()];
    image.getPixels(pixels, 0, image
.getWidth(), 0, 0, image.getWidth(),

```

```

        image.getHeight());
        Log.v("Decode", "" + pixels[0]
;
        Log.v("Decode Alpha", "" + (pi
xels[0] >> 24 & 0xFF));
        Log.v("Decode Red", "" + (pixe
ls[0] >> 16 & 0xFF));
        Log.v("Decode Green", "" + (pi
xels[0] >> 8 & 0xFF));
        Log.v("Decode Blue", "" + (pix
els[0] & 0xFF));
        Log.v("Decode", "" + pixels[0]
;
        Log.v("Decode", "" + image.get
Pixel(0, 0));
        byte[] b = null;
        try {
            b = LSB2bit.convertArray(pi
xels);
        } catch (OutOfMemoryError er)
        {
            AlertDialog.Builder builder =
new AlertDialog.Builder(context);
            builder.setMessage(context.g
etText(R.string.errorImageTooLarge)
)
                .setCancelable(false).set
PositiveButton(
                    context.getText(R.s
tring.ok),
                new DialogInterface
                    e.OnClickListener() {
                        public void onCli
ck(DialogInterface dialog,
                            int id) {
                                DecodeActivit
                                    y.this.finish();
                            }
                        });
            handler.post(runnableDismmi
ss);
            AlertDialog alert = builder.cr
eate();
            alert.show();
            return;
        }
        final String vvv = LSB2bit.deco
deMessage(b, image.getWidth(), ima
ge
            .getHeight());
            handler.post(runnableDismiss
);
            if (vvv == null) {
                handler.post(new Runnable()
                {
                    public void run() {
                        AlertDialog.Builder buil
der = new AlertDialog.Builder(
                            context);
                        builder.setMessage(

```

```

        context.getText(R.string.errorNoMobistegoImage))
        .setCancelable(false)
        .setPositiveButton(
            context.getText(R.string.ok),
            new DialogInterface.OnClickListener() {
                public void onClick(
                    DialogInterface dialog, int id) {
                    DecodeActivity.this.finish();
                }
            });
        AlertDialog alert = builder.create();
        alert.show();
    } else {
        Log.v("Coded message", vvv);
        Runnable runnableSetText =
            new Runnable() {
                public void run() {
                    EditText textDec = findViewById(R.id.editEkstrak);
                    textDec.setText(vvv);
                }
            };
        handler.post(runnableSetText);
    }
}

```

SplashActivity.java

```

package keamanan.pesan;

import android.app.Activity;

```

```
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
```

```
public class SplashActivity extends
Activity {
    private static int TIME_OUT = 50
00; //Time to launch the another activ
ity
    @Override
    protected void onCreate(Bundle sa
vedInstanceState) {
        super.onCreate(savedInstanceSt
ate);
        setContentView(R.layout.activit
y_splash);
        //final View myLayout = findVi
ewById(R.id.startscreen);
        new Handler().postDelayed(new
Runnable() {
            @Override
            public void run() {
                Intent i = new Intent(Splas
hActivity.this, MainActivity.class);
                startActivity(i);
                finish();
            }
        }, TIME_OUT);
    }
}
```

```
} TentangActivity.java
```

```
package keamanan.pesan;
```

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

```
public class TentangActivity extends
Activity {
    /** Called when the activity is first
created. */
    @Override
    public void onCreate(Bundle save
dInstanceState) {
        super.onCreate(savedInstanceSt
ate);
```



setContentView(R.layout.activity_tentang);



2. Daftar Riwayat Hidup

CURRICULUM VITAE



Nama : Nadyah Almirah Simanjuntak
NIM : 0701172077
Tempat/Tanggal Lahir : Medan, 20 Juni 1999
Jenis Kelamin : Perempuan
Alamat : Jl. Cengkeh Mas Gg Cengkeh Turi II, Kec. Medan
Amplas, Kota Medan, Sumatera Utara
Agama : Islam
Status Menikah : Belum Menikah
Nama Orang Tua
Ayah : M. Idris Simanjuntak
Ibu : Khairun Nisha
Email : nalmira1999@gmail.com

PENDIDIKAN FORMAL
SD : SD Swasta Adhyaksa Medan, 2007-2013
SMP : SMP Swasta Adhyaksa Medan, 2013-2015
SMA : SMA Harapan Mandiri Medan, 2015-2017
Perguruan Tinggi : UIN Sumatera Utara, 2017-2022


3. Kartu Bimbingan Skripsi

KARTU BIMBINGAN SKRIPSI

Semester Gasal/Genap Tahun Akademik...../.....

Nama : Nadyah Almirah Sumanjatak	Pembimbing I : Abdul Halim Masuglan, M.Kom
NIM : 0701172097	Pembimbing II : Yusuf Ramadhan Masution, M.Kom
Prog. Studi : Ilmu Komputer	SK Pembimbing :
Judul Skripsi : Kombinasi Algoritma Beaufort Cipher dan LSB2bit Untuk Keamanan File Text	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	1/9-21	• kata pengantar • cek kembali kalimat yang kurang lengkap		21/7-21	Revisi Bab 1	
II	6/9-21	• lampirkan ekstensi untuk file text dan gambar • lampirkan nilai islami		10/8-21	• Teori algoritma masih sedikit • jelaskan cara kerja algoritma	
III	14/9-21	ACC		1/9-21	ACC	
IV		Perbaiki Bab IV		19/10	Perbaiki Bab IV & Def	
V		Perbaiki Bab IV		06/11-21	Acc - Bab IV, V, hasil Text Program	

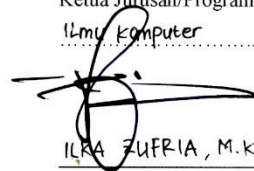
VI		Belum selesai program dgn manual	3/1	20/02	Ace - Sidang	
VII		ACC Bab II dan V berkes keputusan sidang	3/1			
VIII		ACC Bab I, II, III, IV & V berkes ACC sidang	3/1			
IX						
X						

Medan, 16 Februari 2022..

An. Dekan

Ketua Jurusan/Program Studi

Ilmu Komputer



ILFA ZULFRIA, M.KOM

NIP. 198506042015031006

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing