LISTING PROGRAM

1. Ternarycommacodes.java

```
import android.annotation.TargetApi;
import android.os.Build;
import android.os.Environment;
import android.util.Log;

import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.ObjectInputStream;
import java.io.ObjectOutputStream;
import java.util.*;

class TernaryCommaCodes {
   public static int[] fr;
   public static String cs;
   public static String[] tccc;

   public static String getCharSet(String st) {
      StringBuffer s = new StringBuffer();
      int n = st.length();
      for (int i = 0; i < n; i++) {
         String c = Character.toString(st.charAt(i));
         if (s.indexOf(c) == -1)
            s.append(c);
```

```
    }
    return s.toString();
}

public static int countChar(String s, char ch) {
    String c = Character.toString(ch);
    return (s.length() - s.replace(c, "").length()) / c.length();
}

public static int[] countFreq(String st) {
    String charset = getCharSet(st);
    int n = charset.length();
    int[] freq = new int[n];
    for (int i = 0; i < n; i++)
        freq[i] = countChar(st, charset.charAt(i));
    return freq;
}

public static void InsertionSort(int[] freq, String charset) {
    fr = freq;
    cs = charset;
    int n = charset.length();
    StringBuffer sb = new StringBuffer(charset);
    for (int i = 1; i < n; i++)
        for (int j = 0; j < i; j++)
            if (fr[i] > fr[j]){
                int temp = fr[i];
                fr[i] = fr[j];
                fr[j] = temp;
                char ctemp = sb.charAt(i);
                sb.setCharAt(i, sb.charAt(j));
                sb.setCharAt(j, ctemp);
            }
```

```
    cs = sb.toString();
}

public static String string2stb(String st, String c, String[] s) {
    StringBuffer stb = new StringBuffer();
    for (int i = 0; i < st.length(); i++) {
        String z = Character.toString(st.charAt(i));
        int k = c.indexOf(z);
        stb.append(s[k]);
    }
    return stb.toString();
}

private static int find(String s, char c, int i) {
    for (int pos = i; pos < s.length(); pos++)
        if (s.charAt(pos) == c)
            return pos;
    return -1;
}

public static String encode(String stb) {
    StringBuffer code = new StringBuffer();
    int i = 0;
    int index = 0;
    while (i < stb.length()) {
        index = find(stb, 'c', i);
        String x = stb.substring(i, index);
        char y = (char)(tcc2dec(x));
        code.append(y);
        i = index + 1;
    }
    return code.toString();
```

```
}

public static String decode(String code) {
   StringBuffer sb = new StringBuffer();
   int t = code.length();
   for (int i = 0; i < t; i++) {
      char x = code.charAt(i);
      StringBuffer y = new StringBuffer(dec2tcc((int)x));
      sb.append(y);
   }
   return sb.toString();
}

public static String decompress(String stb, String c, String[] s) {
   StringBuffer st = new StringBuffer();
   StringBuffer bit = new StringBuffer();
   for (int i = 0; i < stb.length(); i++) {
      bit.append(stb.charAt(i));
      if (Arrays.asList(s).contains(bit.toString())) {
         int x = Arrays.asList(s).indexOf(bit.toString());
         st.append(c.charAt(x));
         bit = new StringBuffer();
      }
   }
   return st.toString();
}

public static String dec2tcc(int dec) {
   if (dec == 0)
      return "c";
   if (dec == 1)
      return "1c";
   if (dec == 2)
```

```
        return "2c";
    StringBuffer tcc = new StringBuffer();

    int r = dec % 3;
    tcc = new StringBuffer(Integer.toString(r) + tcc);
    dec = dec / 3;
    while (dec != 0){
        r = dec % 3;
        tcc = new StringBuffer(Integer.toString(r) + tcc);
        dec = dec / 3;
    }
    return tcc.toString() + "c";
}

public static int tcc2dec(String tcc) {
    StringBuffer t = new StringBuffer(tcc);
    t.reverse();
    int n = 1;
    int dec = 0;
    for (int i = 0; i < t.length(); i++) {
        if (t.charAt(i) == 'c')
            continue;
        dec = dec + Character.getNumericValue(t.charAt(i)) * n;
        n = n * 3;
    }
    return dec;
}

public static void TCC(String st) {
    int[] freq = countFreq(st);
    String charset = getCharSet(st);
    int t = charset.length();
    InsertionSort(freq, charset);
```

```
        freq = fr;
        charset = cs;
        tccc = new String[t];
        for (int i = 0; i < t; i++)
            tccc[i] = dec2tcc(i);
    }

    public static String proses_dekompresi (String AlamatFileKompresi, String
NamaFileDekompresi){

        long startTime = System.currentTimeMillis();

        String h = readFromFile(AlamatFileKompresi);
        String dekom = dekompresi(h);
        tulis(dekom, NamaFileDekompresi);

        long endTime = System.currentTimeMillis();
        long totalTime = endTime - startTime;

        return "Decompression. Time = "+totalTime + " ms";
    }

    public static void tulis(String tulis, String nama_file){
        try {

            String content = tulis;

            File root = new File(Environment.getExternalStorageDirectory(),
"dekompresi");
            if (!root.exists()) {
                root.mkdirs();
            }
```

```
        File file = new File(root, nama_file);

        // if file doesnt exists, then create it
        if (!file.exists()) {
            file.createNewFile();
        }

        FileWriter fw = new FileWriter(file.getAbsoluteFile());
        BufferedWriter bw = new BufferedWriter(fw);
        bw.write(content);
        bw.close();

        System.out.println("Done");

    } catch (IOException e) {
        e.printStackTrace();
    }
}

public static String bacaFHD(){
    String baca = "";
    String namafile = "header.txt";
    File root = new File(Environment.getExternalStorageDirectory(), "header");
    if (!root.exists()) {
        root.mkdirs();
    }
    File file = new File(root, namafile);
    FileInputStream fis = null;

    try {
        fis = new FileInputStream(file);

        int content;
```

```
        while ((content = fis.read()) != -1) {
            // convert to char and display it
            baca +=  String.valueOf((char) content);
        }

    } catch (IOException e) {
        e.printStackTrace();
    } finally {
        try {
            if (fis != null)
                fis.close();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
    return baca;
}

public static String dekompresi(String stb){
    String sc = bacaFHD();

    String namafile = "header2.txt";
    File root = new File(Environment.getExternalStorageDirectory(), "header");
    if (!root.exists()) {
        root.mkdirs();
    }
    File file = new File(root, namafile);
    String[] tccc ={};
    String ak = "";
    try {

        FileInputStream fos = new FileInputStream(file);
        ObjectInputStream oos = new ObjectInputStream(fos);
```

```
            tccc = (String[]) oos.readObject();
            ak += tccc;
            oos.close();

        }
        catch(Exception e){}

        String dc =          decode(stb);
        System.out.println("========= DECODE STRING==============");
        System.out.println(dc);
        System.out.println("======================\n");

        String ds = decompress(dc, sc, tccc);
        System.out.println("======================");
        System.out.println(sc);
        System.out.println("======================\n");

        System.out.println("======================");
        System.out.println(Arrays.toString(tccc));
        System.out.println("======================\n");

        System.out.println("======================");
        System.out.println(ds);
        System.out.println("======================\n");

        return ds;
    }

    public static String BacaPlain(String namafile){
        String baca = "";
        File root = new File(Environment.getExternalStorageDirectory(), "fileasal");
        if (!root.exists()) {
```

```
            root.mkdirs();
        }
        File file = new File(namafile);
        FileInputStream fis = null;

        try {
            fis = new FileInputStream(file);

            System.out.println("Total file size to read (in bytes) : "
                    + fis.available());

            int content;
            while ((content = fis.read()) != -1) {
                // convert to char and display it
                baca +=  String.valueOf((char) content);
            }
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            try {
                if (fis != null)
                    fis.close();
            } catch (IOException ex) {
                ex.printStackTrace();
            }
        }
        return baca;
    }

    public static String kompresi(String st){
        String kompresi = "";
        TCC(st);
```

```
        String stb = string2stb(st, cs, tccc);
        System.out.print("======"+stb);
        Log.d("ooo",st);
        Log.d("ooo",stb);
        kompresi = stb;
        return kompresi;

    }

    public String prosesTCC(String namafile,  String nama_file_kompresi) {
        String bits = "";

        String str = BacaPlain(namafile);
        //  String nama_file_kompresi ="FILE_KOMPRESI.tccc";
        String hasil = kompresi(str);
        String code          = encode(hasil);
        buatFK(nama_file_kompresi,  code);
        buatFH( "header.txt", "" );
        buatFH2( "header2.txt", "" );
        String stb = string2stb(str, cs, tccc);

        int uncompressed_bits = str.length() * 8;
        int compressed_bits = stb.length();

        double CR = (double)(((compressed_bits * 1.0) / uncompressed_bits) * 100
)/100;
        double SS = (double)((1.0 - 1.0/CR) * 100)*(-1);
        double RC = (double) uncompressed_bits  / compressed_bits;

        bits =  " ================  \n"+
                "Uncompressed_bits = " + uncompressed_bits + "\n"
                +" ================  \n"+
                "Compressed_bits  = " + compressed_bits + "\n"
```

```
            +" ================  \n"
            +"Ratio of Compression = " + RC +"\n"
            +" ================  \n"
            +"Compression Ratio = " + CR + "%"+"\n"
            +" ================  \n"
            +"Space Savings = " + SS + " %"+"\n"
            +" ================  \n";
    return bits ;
}

public static void buatFH( String sFileName, String sBody){
    Environment.getExternalStorageState();

    File root = new File(Environment.getExternalStorageDirectory(), "header");
    if (!root.exists()) {
        root.mkdirs();
    }

    String content = cs;
    File file = new File(root,sFileName);
    // get the content in bytes
    byte[] contentInBytes = content.getBytes();
    try {
        FileOutputStream fOut = new FileOutputStream(file);
        if (!file.exists()) {
            file.createNewFile();
        }
        fOut.write(contentInBytes);
        fOut.close();
    }

    catch (Exception e) {
        // TODO Auto-generated catch block
```

```
            e.printStackTrace();
        }
    }

    public static void buatFH2( String sFileName, String sBody){
        Environment.getExternalStorageState();

        File root = new File(Environment.getExternalStorageDirectory(), "header");
        if (!root.exists()) {
            root.mkdirs();
        }

        String content = cs;
        File file = new File(root,sFileName);
        // get the content in bytes
        byte[] contentInBytes = content.getBytes();
        try {
            FileOutputStream fOut = new FileOutputStream(file);
            if (!file.exists()) {
                file.createNewFile();
            }
            ObjectOutputStream oos = new ObjectOutputStream(fOut);
            oos.writeObject(tccc);
            oos.close();
        }

        catch (Exception e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
    }

    public static void buatFK(String nama_file_kompresi, String hasil){
```

```
        try {

            File root = new File(Environment.getExternalStorageDirectory(),
"kompresi");
            if (!root.exists()) {
                root.mkdirs();
            }
            File gpxfile = new File(root,nama_file_kompresi);
            FileWriter writer = new FileWriter(gpxfile);
            writer.append(hasil);
            writer.flush();
            writer.close();

        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    @TargetApi(Build.VERSION_CODES.KITKAT)
    private  static  String readFromFile(String fileName) {
        String ret = "";
        try (BufferedReader br = new BufferedReader(new FileReader(fileName)))
        {
            String sCurrentLine;
            while ((sCurrentLine = br.readLine()) != null) {
                ret += sCurrentLine;
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
        return ret;
    }
}
```

CURRICULUM VITAE

PERSONAL DATA

Full Name	: Siti Soendari Utami
Nick Name	: Tami
Place/ Date of Birth	: Binjai/ 27 Desember 1992
Sex	: Perempuan
Religion	: Islam
Address	: Perm. Cekapung Indah No. 38 Binjai
E-mail	: siti_utami92@yahoo.co.id

EDUCATION

Bachelor of Computer Science

2014-2016 : Universitas Sumatera Utara, Medan

Diploma of Informatics Engineering

2011-2014 : Universitas Sumatera Utara, Medan

Higher Secondary Education

2008-2011 : SMA Negeri 2 Binjai

Secondary Education

2005-2008 : SMP Ahmad Yani Binjai

Primary Education

1999-2005 : SD Ahmad Yani Binjai