

**IMPLEMENTASI JAVA REMOTE METHOD  
INVOCATION (RMI) PADA SISTEM  
TERDISTRIBUSI  
STUDI KASUS KRIPTOGRAFI HILL CIPHER**

**LISTING PROGRAM**

## Project API (Application Programming Interface)

### Kelas HillCipherInterface.java

```
package Interface;
import java.rmi.*;

public interface HillCipherInterface extends Remote{
    public void klienkonek(String IP)
        throws RemoteException;
    public String enkripdekrip(int ordo, String plainText,
float[][] key)
        throws RemoteException;
}
```

## Project HillServer1App

### 1. Kelas Implements.java

```
package ServerSide;
import Interface.HillCipherInterface;
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class Implements extends UnicastRemoteObject implements
HillCipherInterface{
    public int server;
    public Implements(int i) throws RemoteException {
        this.server = i;
    }
    public String enkripdekrip(int ordo, String plainText,
float[][] key){
        return enkripdekrip.proses(ordo, plainText, key);
    }

    public void klienkonek(String IP) {
        System.out.println("Klien dengan IP Address = "+ IP + "
telah terkoneksi");
    }
}
```

### 2. Kelas enkripdekrip.java

```
package ServerSide;

public class enkripdekrip {
    public static String proses (int ordo, String plainText,
float[][] key){
        System.out.println("Server menerima teks");

        String letter1[] = {"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"};
        System.out.println("Server sedang memproses...");
        String cipherText="";
```

```

int p[][]=new int [ordo][ordo];
int c[][]= new int [ordo][ordo];
for(int i=0;i<plainText.length();i+=ordo){
    for(int j=0;j<ordo;j++){
        String index1;
        if(i+j<plainText.length()){
            char a = plainText.charAt(i+j);
            String va = String.valueOf(a);
            try {
                index1=findIndex(va);
                int k=0;

p[j][k]=Integer.parseInt(String.valueOf(index1));
            } catch (Exception e) {
                System.out.println("Hanya bisa menerima
karakter");
                return "Server 1 : PERINGATAN! Hanya bisa
menerima karakter!!!!";
            }
        }
    }

    int sum=0;
    for(int l=0; l<ordo; l++){
        for(int j=0; j<l; j++){
            for(int k=0; k<ordo; k++){
                sum = (int) (sum + key[l][k] * p[k][j]);
            }
            c[l][j] = sum % 26;//converting to ciphert etxt
            cipherText = cipherText + letter1[sum % 26];
            sum = 0;
        }
    }
    System.out.println("Server mengembalikan teks yang sudah
diproses");
    return cipherText;
}

public static String findIndex(String test){
    String letter[] = {"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"};
    String k;
    for(int j=0; j<letter.length; j++){
        if(test.equalsIgnoreCase(letter[j])){
            k=String.valueOf(j);
            return k;
        }
    }
    return null;
}
}

```

### 3. Kelas enkripdekrip.java

```
package ServerSide;
import java.rmi.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class Server {

    public static void main(String[] args) throws RemoteException,
NotBoundException {
        int i = 1;
        Registry registry = LocateRegistry.createRegistry(1099);
        Implements hc = new Implements(i);
        registry.rebind("Server",hc);
        System.out.println("Server 1 telah berjalan...");
        i++;
    }
}
```

## Project HillServer2App

### 1. Kelas Implements.java

```
package ServerSide;
import Interface.HillCipherInterface;
import java.rmi.*;
import java.rmi.server.UnicastRemoteObject;

public class Implements extends UnicastRemoteObject implements
HillCipherInterface{
    public int server;
    public Implements(int i) throws RemoteException {
        this.server = i;
    }
    public String enkripdekrip(int ordo, String plainText,
float[][] key){
        return enkripdekrip.proses(ordo, plainText, key);
    }

    public void klienkonek(String IP) {
        System.out.println("Klien dengan IP Address = "+ IP + "
telah terkoneksi");
    }
}
```

### 2. Kelas enkripdekrip.java

```
package ServerSide;

public class enkripdekrip {
    public static String proses (int ordo, String plainText,
float[][] key){
        System.out.println("Server menerima teks");
    }
}
```

```

        String letter1[] = {"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"};
        System.out.println("Server sedang memproses...");
        String cipherText="";
        int p[][]=new int [ordo][ordo];
        int c[][]= new int [ordo][ordo];
        for(int i=0;i<plainText.length();i+=ordo){
            for(int j=0;j<ordo;j++){
                String index1;
                if(i+j<plainText.length()){
                    char a = plainText.charAt(i+j);
                    String va = String.valueOf(a);
                    try {
                        index1=findIndex(va);
                        int k=0;

p[j][k]=Integer.parseInt(String.valueOf(index1));
                    } catch (Exception e) {
                        System.out.println("Hanya bisa menerima
karakter");
                        return "Server 1 : PERINGATAN! Hanya bisa
menerima karakter!!!!";
                    }
                }
            }

            int sum=0;
            for(int l=0; l<ordo; l++){
                for(int j=0; j<l; j++){
                    for(int k=0; k<ordo; k++){
                        sum = (int) (sum + key[l][k] * p[k][j]);
                    }
                    c[l][j] = sum % 26;//converting to ciphert etxt
                    cipherText = cipherText + letter1[sum % 26];
                    sum = 0;
                }
            }
            System.out.println("Server mengembalikan teks yang sudah
diproses");
            return cipherText;
        }

        public static String findIndex(String test){
            String letter[] = {"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"};
            String k;
            for(int j=0; j<letter.length; j++){
                if(test.equalsIgnoreCase(letter[j])){
                    k=String.valueOf(j);
                    return k;
                }
            }
            return null;
        }
    }
}

```

### 3. Kelas enkripdekrip.java

```
package ServerSide;
import java.rmi.*;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
public class Server {

    public static void main(String[] args) throws RemoteException,
NotBoundException {
        int i = 1;
        Registry registry = LocateRegistry.createRegistry(1099);
        Implements hc = new Implements(i);
        registry.rebind("Server",hc);
        System.out.println("Server 2 telah berjalan...");
        i++;
    }
}
```

## Project HillClientApp

### 1. Kelas MainFrame.java

```
import java.awt.Dimension;
import java.awt.Toolkit;
import java.io.BufferedReader;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.IOException;
import javax.swing.JFileChooser;
import javax.swing.JOptionPane;
import javax.swing.UIManager;
import cekMatriks.*;
import java.text.DecimalFormat;

// import-an untuk RMI
import Interface.HillCipherInterface;
import java.net.InetAddress;
import java.net.MalformedURLException;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.NotBoundException;

public class MainFrame extends javax.swing.JFrame {
    // ===== obyek untuk koneksi ke server =====
    private HillCipherInterface hciServ1;
    private HillCipherInterface hciServ2;
    // =====

    // ===== menentukan jumlah ordo =====
    private int ordo = 50;
    private void setOrdo(){
        if(cboOrdo.getSelectedIndex()==0) {
            ordo = 2;
        } else if(cboOrdo.getSelectedIndex()==1) {
            ordo = 3;
        }
    }
}
```

```

    } else if(cboOrdo.getSelectedIndex()==2) {
        ordo = 4;
    } else if(cboOrdo.getSelectedIndex()==3) {
        ordo = 5;
    } else if(cboOrdo.getSelectedIndex()==4) {
        ordo = 6;
    } else if(cboOrdo.getSelectedIndex()==5) {
        ordo = 7;
    } else if(cboOrdo.getSelectedIndex()==6) {
        ordo = 10;
    } else if(cboOrdo.getSelectedIndex()==7) {
        ordo = 50;
    }
}
// =====

// ===== array key =====
private float key[][] = new float[ordo][ordo];
// =====

// ===== untuk pecah pecah teks =====
private String teks1, teks2;
private void pecahTeks(String utuh){
    // deklarasi awal ben resik
    teks1 = "";
    teks2 = "";

    int separo = utuh.length()/2;

    // pecahan pertama
    while(separo%ordo!=0) {
        separo++;
    }
    String temp1 = "";
    for(int i = 0; i < separo; i++){
        char pecah = utuh.charAt(i);
        temp1 = (new
StringBulder()).append(temp1).append(pecah).toString();
        teks1 = ((new
StringBulder()).append("").append(temp1).toString());
    }

    // pecahan kedua
    int sisa1 = utuh.length()-separo;
    int sisa2 = sisa1;
    while(sisa1%ordo!=0) {
        sisa1++;
    }
    // kurang berapa
    int kurang = sisa1 - sisa2;

    String temp2 = "";
    for(int i = separo; i < utuh.length(); i++){
        char pecah = utuh.charAt(i);
        temp2 = (new
StringBulder()).append(temp2).append(pecah).toString();
        teks2 = ((new
StringBulder()).append("").append(temp2).toString());
    }
}

```

```

        for(int i=0; i<kurang; i++) {
            teks2 = (new
StringBuildeR()).append(teks2).append("X").toString();
        }
    }
    // =====

    // ===== untuk browse file =====
    private JFileChooser fc;
    private File file;
    final String[] EXT = { ".txt" };
    // =====

    // ===== radio grup =====
    private void radioGroup() {
        rdgED.add(rdoDekrip);
        rdgED.add(rdoEnkrip);
    }
    // =====

    // ===== ben windowe tengah =====
    private void makeCenter() {
        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = this.getSize();
        if(frameSize.height > screenSize.height){
            frameSize.height = screenSize.height;
        }

        if(frameSize.width > screenSize.width){
            frameSize.width = screenSize.width;
        }

        this.setLocation((screenSize.width - frameSize.width)/2,
30);
    }
    // =====

    // ===== get array from file =====
    private void getArray(String berkas){
        String data = null;
        BufferedReader br = null;
        try {
            br = new BufferedReader(new FileReader(berkas));
            int karakter;
            for(int i = 0; i < ordo; i++) {
                data = br.readLine();
                String[] records = data.split(" ");
                for(karakter=0; karakter<records.length; ) {
                    for(int j=0; j<ordo; j++) {
                        key[i][j] =
Integer.parseInt(records[karakter++]);
                    }
                }
            }
        } catch (FileNotFoundException fne) {
            JOptionPane.showMessageDialog(this, "Tidak bisa
menemukan berkas "+ berkas,
"ERROR", JOptionPane.ERROR_MESSAGE);

```



```

    } catch (IOException ioe) {
        System.out.println("Eksepsi tidak diketahui : "+ ioe);
    } catch (ArrayIndexOutOfBoundsException arr) {
        System.out.println(arr);
    } finally {
        if(br!=null){
            try {
                br.close();
            } catch (IOException err) {
                System.out.println("Eksepsi tidak diketahui " +
err);
            }
        }
    }
}
// =====

// ===== menghilangkan spasi di teks =====
private String trimSpace(String s){
    String[] choppedUpString = s.trim().split(" ");
    String trimmedString = "";
    for (int i = 0; i < choppedUpString.length; i++){
        trimmedString = trimmedString + choppedUpString[i];
    }
    return trimmedString;
}

// ===== get own IP =====
private void getOwnIP() {
    try{
        InetAddress ownIP = InetAddress.getLocalHost();

txtOwnIP.setText(String.valueOf(ownIP.getHostAddress()));
    } catch (Exception e){
        System.out.println("Exception caught
="+e.getMessage());
    }
}
// =====

// ===== cek apakah teks ada karakter selain huruf =====
private boolean cekTeks(String teks){
    for(int i = 0; i < teks.length(); i++){
        char a = teks.charAt(i);
        String va = String.valueOf(a);
        if(findIndex(va)==null){
            return false;
        }
    }
    return true;
}

private String findIndex(String test){
    String letter[] = {"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"};
    String k;
    for(int j=0; j<letter.length; j++){
        if(test.equalsIgnoreCase(letter[j])){

```

```

        k=String.valueOf(j);
        return k;
    }
}
return null;
}
// =====

// ===== ngecek kunci =====
private void getArrayCek(String berkas){
    String data = null;
    float kunci[][] = new float[ordo][ordo];
    BufferedReader br = null;
    try {
        br = new BufferedReader(new FileReader(berkas));
        int karakter;
        for(int i = 0; i < ordo; i++) {
            data = br.readLine();
            String[] records = data.split("[ ]");
            for(karakter=0; karakter<records.length; ) {
                for(int j=0; j<ordo; j++) {
                    kunci[i][j] =
Float.parseFloat(records[karakter++]);
                }
            }
        }
        // cek determinannya
        cekDeterminan cd = new cekDeterminan();
        if(cd.cekdeterm(kunci)){
            JOptionPane.showMessageDialog(this, "Kunci matriks
valid");
        } else {
            JOptionPane.showMessageDialog(this, "Kunci matriks
tidak valid karena nilai determinannya 0");
        }

    } catch (FileNotFoundException fne) {
        JOptionPane.showMessageDialog(this, "Tidak bisa
menemukan berkas "+ berkas,
            "ERROR", JOptionPane.ERROR_MESSAGE);
    } catch (IOException ioe) {
        System.out.println("Eksepsi tidak diketahui : "+ ioe);
    } catch (ArrayIndexOutOfBoundsException arr) {
        JOptionPane.showMessageDialog(this, "Jumlah elemen
matriks tidak sama dengan jumlah ordo yang ditentukan",
            "ERROR", JOptionPane.ERROR_MESSAGE);
    } finally {
        if(br!=null){
            try {
                br.close();
            } catch (IOException err) {
                System.out.println("Eksepsi tidak diketahui " +
err);
            }
        }
    }
}
// =====

```

```

// ===== method enkripsi =====
private void enkripsi(){
    String plain = trimSpace(txtPlaintext.getText());
    if(!cekTeks(plain)){
        JOptionPane.showMessageDialog(this, "Tidak bisa
memproses karena ada karakter bukan huruf!",
        "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        pecahTeks(plain);
        try {
            txtCiphertext.setText(((new
StringBuilder()).append(hciServ1.enkripdekrip(ordo, teks1,
key)).append(hciServ2.enkripdekrip(ordo, teks2, key)).toString()));
        } catch (Exception re) {
            JOptionPane.showMessageDialog(this, re + "\nBelum
terkoneksi dengan server",
            "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}
// =====

// ===== method dekripsi =====
private void dekripsi(float[][] invers){
    String cipher = trimSpace(txtCiphertext.getText());
    if(!cekTeks(cipher)){
        JOptionPane.showMessageDialog(this, "Tidak bisa
memproses karena ada karakter bukan huruf!",
        "Error", JOptionPane.ERROR_MESSAGE);
    } else {
        pecahTeks(cipher);
        try {
            txtPlaintext.setText(((new
StringBuilder()).append(hciServ1.enkripdekrip(ordo, teks1,
invers)).append(hciServ2.enkripdekrip(ordo, teks2,
invers)).toString()));
        } catch (Exception re) {
            JOptionPane.showMessageDialog(this, re + "\nBelum
terkoneksi dengan server",
            "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

/** Creates new form MainFrame */
public MainFrame() {
    initComponents();
    radioGroup();
    makeCenter();
    getOwnIP();
    fc = new JFileChooser();
    fc.addChoosableFileFilter(new jenisFile((EXT)));
    fc.setAcceptAllFileFilterUsed(false);
}

/** This method is called from within the constructor to
* initialize the form.
* WARNING: Do NOT modify this code. The content of this method
is

```

```

    * always regenerated by the Form Editor.
    */
    @SuppressWarnings("unchecked")
    // <editor-fold defaultstate="collapsed" desc="Generated Code">
    private void initComponents() {

        rdgED = new javax.swing.ButtonGroup();
        jPanel1 = new javax.swing.JPanel();
        rdoEnkrip = new javax.swing.JRadioButton();
        rdoDekrip = new javax.swing.JRadioButton();
        jScrollPane1 = new javax.swing.JScrollPane();
        txtPlaintext = new javax.swing.JTextArea();
        jLabel1 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        txtKey = new javax.swing.JTextField();
        btnBrowse = new javax.swing.JButton();
        btnCekKunci = new javax.swing.JButton();
        jLabel3 = new javax.swing.JLabel();
        jScrollPane2 = new javax.swing.JScrollPane();
        txtCiphertext = new javax.swing.JTextArea();
        btnClear = new javax.swing.JButton();
        btnProcess = new javax.swing.JButton();
        jPanel3 = new javax.swing.JPanel();
        btnGenerate = new javax.swing.JButton();
        jPanel4 = new javax.swing.JPanel();
        jLabel4 = new javax.swing.JLabel();
        cboOrdo = new javax.swing.JComboBox();
        jPanel5 = new javax.swing.JPanel();
        jLabel5 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        txtOwnIP = new javax.swing.JTextField();
        txtIPServ1 = new javax.swing.JTextField();
        jLabel7 = new javax.swing.JLabel();
        txtIPServ2 = new javax.swing.JTextField();
        lblServer1Status = new javax.swing.JLabel();
        lblServer2Status = new javax.swing.JLabel();
        btnKonekServer1 = new javax.swing.JButton();
        btnKonekServer2 = new javax.swing.JButton();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        txtPortServ1 = new javax.swing.JTextField();
        jLabel10 = new javax.swing.JLabel();
        txtServiceNameServ1 = new javax.swing.JTextField();
        jLabel11 = new javax.swing.JLabel();
        jLabel12 = new javax.swing.JLabel();
        txtPortServ2 = new javax.swing.JTextField();
        jLabel13 = new javax.swing.JLabel();
        txtServiceNameServ2 = new javax.swing.JTextField();
        jSeparator1 = new javax.swing.JSeparator();
        txtStatus = new javax.swing.JTextField();

        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        setTitle("Implementasi Java RMI pada Sistem Terdistribusi  
Studi Kasus : Kriptografi Hill Cipher");
    }

```

```

jPanell1.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "Pilih Metode",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null, new
java.awt.Color(0, 0, 0)));

        rdoEnkrip.setSelected(true);
        rdoEnkrip.setText("Enkripsi");

        rdoDekrip.setText("Dekripsi");

        javax.swing.GroupLayout jPanell1Layout = new
javax.swing.GroupLayout(jPanell1);
        jPanell1.setLayout(jPanell1Layout);
        jPanell1Layout.setHorizontalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(jPanell1Layout.createSequentialGroup()
                .addGap(10, 10, 0)
                .addComponent(rdoEnkrip)
                .addGap(10, 10, 0)
                .addComponent(rdoDekrip)
            )
        );
        jPanell1Layout.setVerticalGroup(

jPanell1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(jPanell1Layout.createSequentialGroup()
                .addGap(10, 10, 0)
                .addComponent(rdoEnkrip)
                .addGap(10, 10, 0)
                .addComponent(rdoDekrip)
            )
        );

        txtPlaintext.setColumns(20);
        txtPlaintext.setLineWrap(true);
        txtPlaintext.setRows(7);
        txtPlaintext.setText("ABCDEFGHJIJ");
        jScrollPane1.setViewportView(txtPlaintext);

        jLabel1.setText("Plainteks :");

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "Ambil Kunci dari File",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null,
java.awt.Color.black));

        jLabel2.setText("Key :");

```

```

        txtKey.setText("D:/Skripsi/key.txt");

        btnBrowse.setText("Pilih File");
        btnBrowse.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                btnBrowseActionPerformed(evt);
            }
        });

        btnCekKunci.setText("Cek Kunci");
        btnCekKunci.addActionListener(new
java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent
evt) {
                btnCekKunciActionPerformed(evt);
            }
        });

        javax.swing.GroupLayout jPanel2Layout = new
javax.swing.GroupLayout(jPanel2);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2Layout.setHorizontalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel2)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(txtKey,
javax.swing.GroupLayout.PREFERRED_SIZE, 185,
javax.swing.GroupLayout.PREFERRED_SIZE)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnBrowse)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnCekKunci)

                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
            .addContainerGap());

        jPanel2Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {btnBrowse, btnCekKunci});

        jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
            .addGroup(jPanel2Layout.createSequentialGroup()
                .addGap(10, 10, 10)
                .addComponent(jLabel2)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(txtKey)

                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnBrowse)
                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                .addComponent(btnCekKunci)

                .addContainerGap(10, Short.MAX_VALUE))
            .addContainerGap());

```

```

.addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
        .addComponent(jLabel2)
        .addComponent(txtKey,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnBrowse)
        .addComponent(btnCekKunci))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );

    jLabel3.setText("Cipherteks :");

    txtCiphertext.setColumns(20);
    txtCiphertext.setLineWrap(true);
    txtCiphertext.setRows(7);
    jScrollPane2.setViewportView(txtCiphertext);

    btnClear.setText("Kosongkan");
    btnClear.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            btnClearActionPerformed(evt);
        }
    });

    btnProcess.setText("Proses");
    btnProcess.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            btnProcessActionPerformed(evt);
        }
    });

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "Buat Kunci Otomatis",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null,
java.awt.Color.black));

    btnGenerate.setText("Generate Kunci");
    btnGenerate.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            btnGenerateActionPerformed(evt);
        }
    });

    javax.swing.GroupLayout jPanel3Layout = new
javax.swing.GroupLayout(jPanel3);
    jPanel3.setLayout(jPanel3Layout);

```

```

        jPanel3Layout.setHorizontalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(btnGenerate)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );
        jPanel3Layout.setVerticalGroup(

jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addContainerGap()
            .addComponent(btnGenerate)

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE))
        );

jPanel4.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "Pilih Ordo Matriks",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null, new
java.awt.Color(0, 0, 0)));

        jLabel4.setText("Ordo");

        cboOrdo.setModel(new javax.swing.DefaultComboBoxModel(new
String[] { "2 x 2", "3 x 3", "4 x 4", "5 x 5", "6 x 6", "7 x 7",
"10 x 10", "50 x 50" }));

        javax.swing.GroupLayout jPanel4Layout = new
javax.swing.GroupLayout(jPanel4);
        jPanel4.setLayout(jPanel4Layout);
        jPanel4Layout.setHorizontalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addComponent(jLabel4)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)
            .addComponent(cboOrdo,
javax.swing.GroupLayout.PREFERRED_SIZE, 79,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addContainerGap(51, Short.MAX_VALUE))
        );
        jPanel4Layout.setVerticalGroup(

jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel4Layout.createSequentialGroup()
            .addContainerGap()

```



```

.addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
        .addComponent(jLabel4)
        .addComponent(cboOrdo,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE,
Short.MAX_VALUE)
    );

jPanel5.setBorder(javax.swing.BorderFactory.createTitledBorder(null
, "Koneksi ke Server",
javax.swing.border.TitledBorder.DEFAULT_JUSTIFICATION,
javax.swing.border.TitledBorder.DEFAULT_POSITION, null, new
java.awt.Color(0, 0, 0)));

    jLabel5.setText("IP Klien");

    jLabel6.setText("Server 1");

    txtOwnIP.setEditable(false);

    jLabel7.setText("Server 2");

    lblServer1Status.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/stopped.png")))
; // NOI18N

    lblServer2Status.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/stopped.png")))
; // NOI18N

    btnKonekServer1.setText("Koneksikan");
    btnKonekServer1.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            btnKonekServer1ActionPerformed(evt);
        }
    });

    btnKonekServer2.setText("Koneksikan");
    btnKonekServer2.addActionListener(new
java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent
evt) {
            btnKonekServer2ActionPerformed(evt);
        }
    });

    jLabel8.setText("rmi://");

    jLabel9.setText(":");

    txtPortServ1.setColumns(4);

```

```

jLabel10.setText("/");

txtServiceNameServ1.setColumns(6);

jLabel11.setText("rmi://");

jLabel12.setText(":");

txtPortServ2.setColumns(4);

jLabel13.setText("/");

txtServiceNameServ2.setColumns(6);

    javax.swing.GroupLayout jPanel5Layout = new
javax.swing.GroupLayout(jPanel5);
    jPanel5.setLayout(jPanel5Layout);
    jPanel5Layout.setHorizontalGroup(

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
        .addGroup(jPanel5Layout.createSequentialGroup()
            .addComponent(jLabel5)
            .addComponent(jLabel6)
            .addComponent(jLabel7)
            .addGap(32, 32, 32)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING, false)
            .addGroup(jPanel5Layout.createSequentialGroup()
                .addComponent(jLabel8)
                .addComponent(jLabel11)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING, false)
            .addComponent(txtIPServ2)
            .addComponent(txtIPServ1,
javax.swing.GroupLayout.DEFAULT_SIZE, 106, Short.MAX_VALUE))
            .addComponent(txtOwnIP)

        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

        .addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING, false)
            .addGroup(jPanel5Layout.createSequentialGroup()
                .addComponent(jLabel9)

```

```

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(txtPortServ1,
javax.swing.GroupLayout.PREFERRED_SIZE, 38,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel5Layout.createSequentialGroup()
        .addComponent(jLabel12)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(txtPortServ2, 0, 0,
Short.MAX_VALUE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
        .addGroup(jPanel5Layout.createSequentialGroup()
        .addComponent(jLabel10)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(txtServiceNameServ1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(jPanel5Layout.createSequentialGroup()
        .addComponent(jLabel13)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(txtServiceNameServ2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)))
        .addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
        .addComponent(btnKonekServer1)
        .addComponent(btnKonekServer2))
        .addGap(18, 18, 18)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.LEADING)
        .addComponent(lblServer1Status)
        .addComponent(lblServer2Status))
        .addContainerGap()
    );

jPanel5Layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {btnKonekServer1, btnKonekServer2});

jPanel5Layout.setVerticalGroup(

```

```

jPanel5Layout.createParallelGroup(javax.swing.GroupLayout.Alignment
.LEADING)
    .addGroup(jPanel5Layout.createSequentialGroup()
        .addContainerGap()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.TRAILING)
    .addComponent(lblServer1Status)
    .addGroup(jPanel5Layout.createSequentialGroup()

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
    .addComponent(jLabel5)
    .addComponent(txtOwnIP,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELAT
ED)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
    .addComponent(jLabel6)
    .addComponent(jLabel8)
    .addComponent(txtIPServ1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel9)
    .addComponent(txtPortServ1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel10)
    .addComponent(txtServiceNameServ1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(btnKonekServer1))))
    .addGap(12, 12, 12)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.TRAILING)

.addGroup(jPanel5Layout.createParallelGroup(javax.swing.GroupLayout
.Alignment.BASELINE)
    .addComponent(jLabel7)
    .addComponent(jLabel11)
    .addComponent(txtIPServ2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
    .addComponent(jLabel12)
    .addComponent(txtPortServ2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

```

```

        .addComponent(jLabel13)
        .addComponent(txtServiceNameServ2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(btnKonekServer2))
        .addComponent(lblServer2Status))
        .addContainerGap(12, Short.MAX_VALUE))
    );

    txtStatus.setEditable(false);

    javax.swing.GroupLayout layout = new
javax.swing.GroupLayout(getContentPane());
    getContentPane().setLayout(layout);
    layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup())
        .addContainerGap()

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)
            .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

        .addGroup(layout.createSequentialGroup())

        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING, false)
            .addComponent(jPanel4,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
            .addComponent(jPanel1,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(18, 18, 18)
            .addComponent(jPanel5,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(layout.createSequentialGroup())
            .addComponent(jPanel2,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)
            .addComponent(jPanel3,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))

        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
layout.createSequentialGroup())

```

```

        .addComponent(txtStatus,
javax.swing.GroupLayout.DEFAULT_SIZE, 649, Short.MAX_VALUE)
        .addGap(18, 18, 18)
        .addComponent(btnProcess)
        .addGap(18, 18, 18)
        .addComponent(btnClear))
        .addComponent(jLabel1)
        .addComponent(jScrollPane1,
javax.swing.GroupLayout.DEFAULT_SIZE, 855, Short.MAX_VALUE)
        .addComponent(jLabel3)
        .addComponent(jScrollPane2,
javax.swing.GroupLayout.DEFAULT_SIZE, 855, Short.MAX_VALUE))
        .addContainerGap())
        .addGroup(layout.createSequentialGroup())
        .addComponent(jSeparator1,
javax.swing.GroupLayout.DEFAULT_SIZE, 649, Short.MAX_VALUE)
        .addGap(216, 216, 216))))
);

layout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {btnClear, btnProcess});

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
    .addGroup(layout.createSequentialGroup()
        .addContainerGap())

    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING, false)
        .addGroup(layout.createSequentialGroup()
            .addComponent(jPanel1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

            .addComponent(jPanel4,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE))
            .addComponent(jPanel5,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
            .addGap(18, 27, Short.MAX_VALUE)
            .addComponent(jLabel1)

            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED
)

            .addComponent(jScrollPane1,
javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.PREFERRED_SIZE)
            .addGap(18, 18, 18)

            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignm
ent.LEADING)

```

```

        .addComponent(jPanel3,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jPanel2,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE))
        .addGap(18, 18, 18)
        .addComponent(jLabel3)

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED
)
        .addComponent(jScrollPane2,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javafx.swing.GroupLayout.Alignm
ent.LEADING, false)
        .addGroup(layout.createSequentialGroup()
        .addComponent(jSeparator1,
javafx.swing.GroupLayout.PREFERRED_SIZE, 10,
javafx.swing.GroupLayout.PREFERRED_SIZE)

.addPreferredGap(javafx.swing.LayoutStyle.ComponentPlacement.RELATED
, javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(txtStatus,
javafx.swing.GroupLayout.PREFERRED_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.PREFERRED_SIZE))
        .addComponent(btnClear,
javafx.swing.GroupLayout.DEFAULT_SIZE,
javafx.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(btnProcess,
javafx.swing.GroupLayout.DEFAULT_SIZE, 33, Short.MAX_VALUE))
        .addContainerGap()
    );

    pack();
} // </editor-fold>

private void
btnProcessActionPerformed(java.awt.event.ActionEvent evt) {
    setOrdo();
    DecimalFormat runtimeDF = new DecimalFormat("###0.000000");
    // pokoke masalah kunci !!!
    String berkas = txtKey.getText();
    getArray(berkas);

    // enkrip dekrip pokoke
    if(rdoEnkrip.isSelected()) {
        long start = System.currentTimeMillis();
        enkripsi();
        long finish = System.currentTimeMillis();
        double newRunTime = (double)(finish - start) / 1000;
        txtStatus.setText("Waktu eksekusi : " +
runtimeDF.format(newRunTime) + " detik");
    } else if (rdoDekrip.isSelected()) {

```

```

        long start = System.currentTimeMillis();
        Nginverse inv = new Nginverse();
        float invers[][] = new float[ordo][ordo];
        invers = inv.nginvers(key, ordo);

        // pembulatan
        DecimalFormat twoDForm = new DecimalFormat("#");
        for(int i=0; i<ordo; i++) {
            for(int j=0; j<ordo; j++) {
                invers[i][j] =
Float.valueOf(twoDForm.format(invers[i][j]));
            }
        }

        dekripsi(invers);
        long finish = System.currentTimeMillis();
        double newRunTime = (double)(finish - start) / 1000;
        txtStatus.setText("Waktu eksekusi : " +
runtimeDF.format(newRunTime) + " detik");
    }
}

private void
btnGenerateActionPerformed(java.awt.event.ActionEvent evt) {
    // menentukan ordo
    setOrdo();
    cekMatriks cm = new cekMatriks();
    if(ordo<=10) {
        if(cm.buatFile(ordo)==true) {
            txtKey.setText("D:/Skripsi/key.txt");
            JOptionPane.showMessageDialog(this, "File key.txt
berhasil dibuat di D:/Skripsi/key.txt");
        } else {
            JOptionPane.showMessageDialog(this, "Gagal membuat
file key.txt karena tidak menemukan folder D:/Skripsi/");
        }
    } else {
        cmToFile(ordo);
        txtKey.setText("D:/Skripsi/key.txt");
        JOptionPane.showMessageDialog(this, "File key.txt
berhasil dibuat di D:/Skripsi/key.txt");
    }
}

private void
btnBrowseActionPerformed(java.awt.event.ActionEvent evt) {
    if(fc.showOpenDialog(this)==JFileChooser.APPROVE_OPTION) {
        file = fc.getSelectedFile();
        txtKey.setText(file.toString());
    }
}

private void
btnKonekServer1ActionPerformed(java.awt.event.ActionEvent evt) {
    String IPAddress = txtIPServ1.getText();
    String port = txtPortServ1.getText();
    String serviceName = txtServiceNameServ1.getText();
    String rmiAddress = "rmi://" + IPAddress + ":" + port + "/"
+ serviceName;
}

```



```

        try {
            hciServ1 = (HillCipherInterface)
Naming.lookup(rmiAddress);
            btnKonekServer1.setText("Terkoneksi");
            btnKonekServer1.setEnabled(false);
            lblServer1Status.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/started.png")))
;
            txtIPServ1.setEnabled(false);
            txtPortServ1.setEnabled(false);
            txtServiceNameServ1.setEnabled(false);
            hciServ1.klienkoneksi(txtOwnIP.getText());
            txtStatus.setText("Terkoneksi dengan Server 1 : " +
IPAddress);
        } catch (MalformedURLException murle) {
            JOptionPane.showMessageDialog(this,
"MalformedURLException!!", "Error", JOptionPane.ERROR_MESSAGE);
            txtStatus.setText("Server 1 : MalformedURLException! "
+ murle.toString());
        } catch (RemoteException re) {
            JOptionPane.showMessageDialog(this,
"RemoteException!!", "Error", JOptionPane.ERROR_MESSAGE);
            txtStatus.setText("Server 1 : RemoteException! " +
re.toString());
        } catch (NotBoundException nbe){
            JOptionPane.showMessageDialog(this,
"NotBoundException!!", "Error", JOptionPane.ERROR_MESSAGE);
            txtStatus.setText("Server 1 : NotBoundException! " +
nbe.toString());
        } catch (Exception e) {
            JOptionPane.showMessageDialog(this, e.toString(),
"Error", JOptionPane.ERROR_MESSAGE);
            txtStatus.setText("Server 1 : Unknown Error : " +
e.toString());
        }
    }

    private void
btnKonekServer2ActionPerformed(java.awt.event.ActionEvent evt) {
        String IPAddress = txtIPServ2.getText();
        String port = txtPortServ2.getText();
        String serviceName = txtServiceNameServ2.getText();
        String rmiAddress = "rmi://" + IPAddress + ":" + port + "/"
+ serviceName;
        try {
            hciServ2 = (HillCipherInterface)
Naming.lookup(rmiAddress);
            btnKonekServer2.setText("Terkoneksi");
            btnKonekServer2.setEnabled(false);
            lblServer2Status.setIcon(new
javax.swing.ImageIcon(getClass().getResource("/icon/started.png")))
;
            txtIPServ2.setEnabled(false);
            txtPortServ2.setEnabled(false);
            txtServiceNameServ2.setEnabled(false);
            hciServ2.klienkoneksi(txtOwnIP.getText());
            txtStatus.setText("Terkoneksi dengan Server 2 : " +
IPAddress);
        } catch (MalformedURLException murle) {

```

```

        JOptionPane.showMessageDialog(this,
"MalformedURLException!!", "Error", JOptionPane.ERROR_MESSAGE);
        txtStatus.setText("Server 2 : MalformedURLException! "
+ murle.toString());
    } catch (RemoteException re) {
        JOptionPane.showMessageDialog(this,
"RemoteException!!", "Error", JOptionPane.ERROR_MESSAGE);
        txtStatus.setText("Server 2 : RemoteException! " +
re.toString());
    } catch (NotBoundException nbe){
        JOptionPane.showMessageDialog(this,
"NotBoundException!!", "Error", JOptionPane.ERROR_MESSAGE);
        txtStatus.setText("Server 2 : NotBoundException! " +
nbe.toString());
    } catch (Exception e) {
        JOptionPane.showMessageDialog(this, e.toString(),
"Error", JOptionPane.ERROR_MESSAGE);
        txtStatus.setText("Server 2 : Unknown Error : " +
e.toString());
    }
}

private void
btnCekKunciActionPerformed(java.awt.event.ActionEvent evt) {
    setOrdo();
    String berkas = txtKey.getText();
    getArrayCek(berkas);
}

private void btnClearActionPerformed(java.awt.event.ActionEvent
evt) {
    rdoEnkrip.setSelected(true);
    cboOrdo.setSelectedIndex(0);
    txtPlaintext.setText("");
    txtPlaintext.requestFocus();
    txtCiphertext.setText("");
    txtKey.setText("D:/Skripsi/key.txt");
}

/**
 * @param args the command line arguments
 */
public static void main(String args[]) {
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            try {

UIManager.setLookAndFeel("com.sun.java.swing.plaf.windows.WindowsLo
okAndFeel");
                } catch (Exception e) {
                    e.printStackTrace();
                }
                new MainFrame().setVisible(true);
            }
        });
}

// Variables declaration - do not modify
private javax.swing.JButton btnBrowse;

```

```

private javax.swing.JButton btnCekKunci;
private javax.swing.JButton btnClear;
private javax.swing.JButton btnGenerate;
private javax.swing.JButton btnKonekServer1;
private javax.swing.JButton btnKonekServer2;
private javax.swing.JButton btnProcess;
private javax.swing.JComboBox cboOrdo;
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPanel jPanel4;
private javax.swing.JPanel jPanel5;
private javax.swing.JScrollPane jScrollPane1;
private javax.swing.JScrollPane jScrollPane2;
private javax.swing.JSeparator jSeparator1;
private javax.swing.JLabel lblServer1Status;
private javax.swing.JLabel lblServer2Status;
private javax.swing.ButtonGroup rdgED;
private javax.swing.JRadioButton rdoDekrip;
private javax.swing.JRadioButton rdoEnkrip;
private javax.swing.JTextArea txtCiphertext;
private javax.swing.JTextField txtIPServ1;
private javax.swing.JTextField txtIPServ2;
private javax.swing.JTextField txtKey;
private javax.swing.JTextField txtOwnIP;
private javax.swing.JTextArea txtPlaintext;
private javax.swing.JTextField txtPortServ1;
private javax.swing.JTextField txtPortServ2;
private javax.swing.JTextField txtServiceNameServ1;
private javax.swing.JTextField txtServiceNameServ2;
private javax.swing.JTextField txtStatus;
// End of variables declaration
}

```

## 2. Kelas Nginvers.java

```

import java.text.NumberFormat;
import javax.swing.JOptionPane;

public class Nginverse {
    private static int decimals = 3;
    private int iDF = 0;
    private static NumberFormat nf;
    static MainFrame mn = new MainFrame();
}

```

```

public float[][] nginvers(float[][] matrix, int n) {
    float matriks[][] = new float[n][n];
    for(int i=0; i<n; i++) {
        for(int j=0; j<n; j++) {
            matriks[i][j] = matrix[i][j];
        }
    }

    Nginverse app = new Nginverse();
    nf = NumberFormat.getInstance();
    nf.setMinimumFractionDigits(1);
    nf.setMaximumFractionDigits(decimals);

    // Nggoleki Adjoin
    float adjoin[][] = new float[n][n];
    try {
        adjoin = app.Adjoint(matriks);
    } catch (Exception ex) {
        System.out.println(ex.toString());
    }

    // nggoleki nilai di
    float di = app.Invert(matriks);

    // di * adjoint % 26
    float inverse[][] = new float[n][n];

    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            inverse[i][j] = di * adjoin[i][j] % 26;
        }
    }

    // jika kurang dari 0, ditambah dengan 26
    for (int i=0; i<n; i++) {
        for (int j=0; j<n; j++) {
            if(inverse[i][j] < 0)
                inverse[i][j] += 26;
        }
    }
    // mengembalikan nilai inverse
    return inverse;
}

// itung-itungane
private float Invert(float[][] matriks) {
    // method untuk nyari nilai di
    float dot = Determinant(matriks);
    dot = ((dot%26)+26)%26;
    //System.out.println("Det = "+ dot);

    int di = 0;

    for(int i=0;i<26;i++){
        if((dot*i)%26 == 1)
            di = i;
    }

    if(di == 0){

```

```

        JOptionPane.showMessageDialog(mn, "Tidak bisa invert",
                                     "Error",
JOptionPane.ERROR_MESSAGE);
        return 0;
    } else {
        return di;
    }
}

private float[][] Transpose(float[][] a) {
    float m[][] = new float[a[0].length][a.length];

    for (int i = 0; i < a.length; i++)
        for (int j = 0; j < a[i].length; j++)
            m[j][i] = a[i][j];
    return m;
}

private float[][] Adjoint(float[][] a) throws Exception {
    int tms = a.length;

    float m[][] = new float[tms][tms];

    int ii, jj, ia, ja;
    float det;

    for (int i = 0; i < tms; i++)
        for (int j = 0; j < tms; j++) {
            ia = ja = 0;

            float ap[][] = new float[tms - 1][tms - 1];

            for (ii = 0; ii < tms; ii++) {
                for (jj = 0; jj < tms; jj++) {
                    if ((ii != i) && (jj != j)) {
                        ap[ia][ja] = a[ii][jj];
                        ja++;
                    }
                }
                if ((ii != i) && (jj != j)) {
                    ia++;
                }
                ja = 0;
            }

            det = Determinant(ap);
            m[i][j] = (float) Math.pow(-1, i + j) * det;
        }

    m = Transpose(m);

    return m;
}

private float[][] UpperTriangle(float[][] m) {
    float f1 = 0;
    float temp = 0;
    int tms = m.length; // get This Matrix Size (could be
smaller than global)

```

```

int v = 1;

iDF = 1;

for (int col = 0; col < tms - 1; col++) {
    for (int row = col + 1; row < tms; row++) {
        v = 1;

        outahere: while (m[col][col] == 0) // check if 0 in
diagonal
        { // if so switch until not
            if (col + v >= tms) // check if switched all
rows
                {
                    iDF = 0;
                    break outahere;
                } else {
                    for (int c = 0; c < tms; c++) {
                        temp = m[col][c];
                        m[col][c] = m[col + v][c]; // switch
rows
                        m[col + v][c] = temp;
                    }
                    v++; // count row switchs
                    iDF = iDF * -1; // each switch changes
determinant factor
                }
            }

            if (m[col][col] != 0) {
                try {
                    f1 = (-1) * m[row][col] / m[col][col];
                    for (int i = col; i < tms; i++) {
                        m[row][i] = f1 * m[col][i] + m[row][i];
                    }
                } catch (Exception e) {
                    System.out.println("Still Here!!!");
                }
            }
        }
    }

    return m;
}

private float Determinant(float[][] matrix) {
    int tms = matrix.length;

    float det = 1;

    matrix = UpperTriangle(matrix);

    for (int i = 0; i < tms; i++) {
        det = det * matrix[i][i];
    } // multiply down diagonal

    det = det * iDF; // adjust w/ determinant factor

```

```

        return det;
    }
}

```

### 3. Kelas JenisFile.java

```

import java.io.File;
import javax.swing.filechooser.FileFilter;

class jenisFile extends FileFilter {
    private String[] s;
    jenisFile(String[] sArg){
        s = sArg;
    }
    public boolean accept(File fArg){
        if(fArg.isDirectory())
            return true;
        for(int i=0; i<s.length; ++i)

if(fArg.getName().toLowerCase().indexOf(s[i].toLowerCase())>0)
            return true;
        return false;
    }

    public String getDescription(){
        String tmp = "";
        for(int i=0; i<s.length; ++i)
            tmp += "*" + s[i] + " ";
        return tmp;
    }
}

```

### 4. Kelas cekDeterminan.java

```

package cekMatriks;

public class cekDeterminan {
    private int iDF = 0;

    public boolean cekdeterm(float[][] matrik) {
        // nyari determinan
        float dot = Determinant(matrik);
        dot = ((dot%26)+26)%26;
        //System.out.println("Det = "+ det);

        int di = 0;

        for(int i=0;i<26;i++){
            if((dot*i)%26 == 1)
                di = i;
        }

        System.out.println("di = "+ di);

        if(di == 0){
            System.out.println("Tidak bisa invert");
            return false;
        } else {
            return true;
        }
    }
}

```

```

    }
}
private float Determinant(float[][] matrix) {
    int tms = matrix.length;

    float det = 1;

    matrix = UpperTriangle(matrix);

    for (int i = 0; i < tms; i++) {
        det = det * matrix[i][i];
    } // multiply down diagonal

    det = det * iDF; // adjust w/ determinant factor
    return det;
}

private float[][] UpperTriangle(float[][] m) {
    float f1 = 0;
    float temp = 0;
    int tms = m.length; // get This Matrix Size (could be
smaller than global)
    int v = 1;

    iDF = 1;

    for (int col = 0; col < tms - 1; col++) {
        for (int row = col + 1; row < tms; row++) {
            v = 1;

            outahere: while (m[col][col] == 0) // check if 0 in
diagonal
            { // if so switch until not
                if (col + v >= tms) // check if switched all
rows
                {
                    iDF = 0;
                    break outahere;
                } else {
                    for (int c = 0; c < tms; c++) {
                        temp = m[col][c];
                        m[col][c] = m[col + v][c]; // switch
rows
                        m[col + v][c] = temp;
                    }
                    v++; // count row switches
                    iDF = iDF * -1; // each switch changes
determinant factor
                }
            }

            if (m[col][col] != 0) {
                try {
                    f1 = (-1) * m[row][col] / m[col][col];
                    for (int i = col; i < tms; i++) {
                        m[row][i] = f1 * m[col][i] + m[row][i];
                    }
                } catch (Exception e) {
                    System.out.println("Still Here!!!");
                }
            }
        }
    }
}

```



```

        }
    }
}
return m;
}
}

```

## 5. Kelas cekMatriks.java

```

package cekMatriks;

import java.io.BufferedReader;
import java.io.FileNotFoundException;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Random;

public class cekMatriks {
    public boolean toFile(int ordo) {
        int bts = 10;
        Random rdm = new Random();
        // generate baris pertama
        String temp2 = String.valueOf(rdm.nextInt(bts));
        for(int j=0; j<(ordo-1); j++) {
            String temp1 = String.valueOf(rdm.nextInt(bts));
            temp2 = (new StringBuilder()).append(temp2).append("
").append(temp1).toString();
        }
        String temp3 = temp2;

        // baris kedua dan selanjutnya
        for(int i=0; i<(ordo-1); i++) {
            temp2 = String.valueOf(rdm.nextInt(bts));
            for(int j=0; j<(ordo-1); j++) {
                String temp1 = String.valueOf(rdm.nextInt(bts));
                temp2 = (new
String
Builder()).append(temp2).append("
").append(temp1).toString();
            }
            temp3 = (new
String
Builder()).append(temp3).append("\n").append(temp2).toString(
);
        }

        // menulis temp3 ke file key.txt
        PrintWriter hasil = null;
        try {
            hasil = new PrintWriter(new
FileWriter("D:/Skripsi/key.txt"));
        } catch (IOException ex) {
            System.out.println("Gagal membuat file key.txt karena
tidak menemukan folder D:/Skripsi/");
            return false;
        }
    }
}

```

```

        try {
            hasil.print(temp3);
        } catch (Exception e) {
            System.out.println("Kesalahan tulis file " +
e.toString());
            return false;
        }
    finally {
        try {
            hasil.close();
        }
        catch (Exception e) {
            System.out.println("Tidak bisa menutup key.txt "+
e.toString());
            return false;
        }
    }
    return true;
}

private float[][] fromFile(int ordo){
    float[][] key = new float[ordo][ordo];
    String args[] = null;
    String data = null;
    BufferedReader br = null;
    try {
        br = new BufferedReader(new
FileReader("D:/Skripsi/key.txt"));
        int karakter;
        for(int i = 0; i < ordo; i++) {
            data = br.readLine();
            String[] records = data.split("[ ]");
            for(karakter=0; karakter<records.length; ) {
                for(int j=0; j<ordo; j++) {
                    key[i][j] =
Integer.parseInt(records[karakter++]);
                }
            }
        }
    } catch (FileNotFoundException fne) {
        System.out.println("File : " + args[0] + " tidak
ditemukan");
    } catch (IOException ioe) {
        System.out.println("Eksepsi tidak diketahui : "+ ioe);
    } catch (ArrayIndexOutOfBoundsException arr) {
        System.out.println(arr);
    } finally {
        if(br!=null){
            try {
                br.close();
            } catch (IOException err) {
                System.out.println("Eksepsi tidak diketahui " +
err);
            }
        }
    }
    return key;
}

```

```
public boolean buatFile(int ordo) {
    boolean hasil = false;
    float key[][] = new float[ordo][ordo];

    cekMatriks cm = new cekMatriks();
    cekDeterminan cd = new cekDeterminan();
    do {
        hasil = cm.toFile(ordo);
        if(hasil==false){
            return false;
        }
        key = cm.fromFile(ordo);
        hasil = cd.cekdeterm(key);
    } while (hasil==false);

    return hasil;
}
}
```