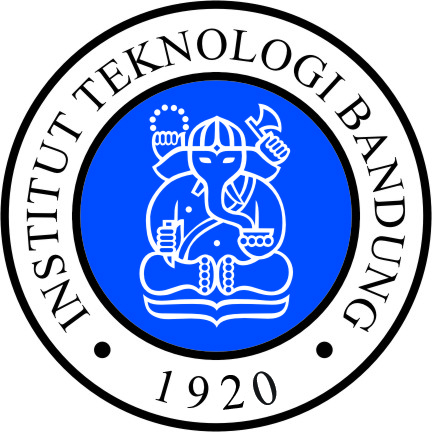
**IF4020 Kriptografi**

**Tugas Kecil 1 Vigenere Cipher Applet**



Disusun Oleh:

Timothy Pratama / 13512032

**Program Studi Teknik Informatika**

**Institut Teknologi Bandung**

**Jl. Ganesha 10, Bandung 40132**

# **Source Code Program Java**

## **Engine.java**

Engine.java adalah kelas yang mengimplementasikan *method* – *method* pada vigenere cipher.

package engine;

import java.io.File;

import java.io.FileNotFoundException;

import java.io.PrintWriter;

import java.io.UnsupportedEncodingException;

import java.util.Scanner;

import java.util.logging.Level;

import java.util.logging.Logger;

/\*\*

\*

\* @author timothy.pratama

\*/

public class Engine {

//Attributes

private String ciphertext;

private String plaintext;

private String key; //maksimal 25 huruf

private int mode; //1: standard, 2: extended, 3: autokey

private int display; //1: apa adanya, 2: tanpa spasi, 3: dalam kelompok 5 huruf

private char[][] vigenereSquare26;

private char[][] vigenereSquare256;

//Methods

/\* Constructors \*/

public Engine() {

ciphertext = "";

plaintext = "";

key = "";

mode = 0;

display = 0;

createVigenereSquare();

}

public Engine(String ciphertext, String plaintext, String key, int mode, int display) {

this.ciphertext = ciphertext;

this.plaintext = plaintext;

this.key = key;

this.mode = mode;

this.display = display;

}

/\* Initialize Vigenere Square (26 and 256 characters) \*/

private void createVigenereSquare26() {

vigenereSquare26 = new char[26][26];

int c;

for (int b=0; b<26; b++) { //range 97 - 122

for (int k=0; k<26; k++) {

c = b + k + 97;

if (c > 122) {

c -= 26;

}

vigenereSquare26[b][k] = (char) c;

}

}

}

private void createVigenereSquare256() {

vigenereSquare256 = new char[256][256];

int c;

String output = "";

for (int b=0; b<256; b++) {

for (int k=0; k<256; k++) {

c = b + k;

if (c > 255) {

c -= 256;

}

vigenereSquare256[b][k] = (char) c;

}

}

}

private void createVigenereSquare() {

createVigenereSquare26();

createVigenereSquare256();

}

/\* Getter and Setter \*/

public String getCiphertext() {

String temp = "";

if(display == 1) {

temp = ciphertext;

} else if(display == 2) {

temp = ciphertext.replaceAll("\\s+","");

} else if(display == 3) {

temp = ciphertext.replaceAll("\\s+","");

int interval = 5;

int idx = 0;

String result = "";

while(idx + interval < temp.length()) {

result += temp.substring(idx, idx + interval) + " ";

idx += interval;

}

result += temp.substring(idx);

temp = result;

}

return temp;

}

public void setCiphertext(String ciphertext) {

this.ciphertext = ciphertext;

}

public String getPlaintext() {

return plaintext;

}

public void setPlaintext(String plaintext) {

this.plaintext = plaintext.toLowerCase();

}

public String getKey() {

return key;

}

public void setKey(String key) {

this.key = key;

}

public int getMode() {

return mode;

}

public void setMode(int mode) {

this.mode = mode;

}

public int getDisplay() {

return display;

}

public void setDisplay(int display) {

this.display = display;

}

public void readFile(String path) {

plaintext = "";

File file = new File(path);

try {

Scanner input = new Scanner(file);

do {

plaintext += input.nextLine() + "\n";

} while (input.hasNextLine());

plaintext = plaintext.toLowerCase();

} catch (FileNotFoundException ex) {

Logger.getLogger(Engine.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\* save functions \*/

public void saveFile(String path) {

try {

PrintWriter writer = new PrintWriter(path, "UTF-8");

writer.print(ciphertext);

writer.close();

} catch (FileNotFoundException | UnsupportedEncodingException ex) {

Logger.getLogger(Engine.class.getName()).log(Level.SEVERE, null, ex);

}

}

/\* Cryptography functions \*/

public void encrypt() {

createKey();

ciphertext = "";

int plaintextLength = plaintext.length();

char c;

char k;

int j = 0;

if(mode == 1 || mode == 3) { //standard

for(int i = 0; i<plaintextLength; i++) {

c = plaintext.charAt(i);

if(c == ' ' || c == '\n') {

ciphertext += c;

} else {

k = key.charAt(j);

ciphertext += vigenereSquare26[((int) k) - 97][((int) c) - 97];

j++;

}

}

} else if (mode == 2 || mode == 4) { //extended

for(int i = 0; i<plaintextLength; i++) {

c = plaintext.charAt(i);

if(c == ' ' || c == '\n') {

ciphertext += c;

} else {

k = key.charAt(j);

ciphertext += vigenereSquare256[((int) k)][((int) c)];

j++;

}

}

}

}

public void decrypt() {

if(mode <= 2) {

createKey();

} else {

decodeKey();

}

plaintext = "";

int ciphertextLength = ciphertext.length();

char c;

char k;

int j = 0;

if(mode == 1 || mode == 3) { //standard

for(int i = 0; i<ciphertextLength; i++) {

c = ciphertext.charAt(i);

if(c == ' ' || c == '\n') {

plaintext += c;

} else {

k = key.charAt(j);

for(int l=0; l<26; l++) {

if(c == vigenereSquare26[((int)k)-97][l]) {

plaintext += (char) (l+97);

break;

}

}

j++;

}

}

} else if (mode == 2 || mode == 4) { //extended

for(int i = 0; i<ciphertextLength; i++) {

c = ciphertext.charAt(i);

if(c == ' ' || c == '\n') {

plaintext += c;

} else {

k = key.charAt(j);

for(int l=0; l<256; l++) {

if(c == vigenereSquare256[(int)k][l]) {

plaintext += (char) (l);

break;

}

}

j++;

}

}

}

}

private void createKey() {

int keyLength = key.length();

int plaintextLength = plaintext.replaceAll("\\s+","").length();

int ciphertextLength = ciphertext.replaceAll("\\s+","").length();

int length;

if(plaintextLength > ciphertextLength) {

length = plaintextLength;

} else {

length = ciphertextLength;

}

int j = 0;

if(mode <= 2) {

for(int i=key.length(); i<length; i++) {

key += key.charAt(j % keyLength);

j++;

}

}

else {

String temp = plaintext.replaceAll("\\s+","");

plaintextLength = temp.length();

for(int i=keyLength; i<plaintextLength; i++) {

while(plaintext.charAt(j) == ' ' || plaintext.charAt(j) == '\n') {

j++;

}

key += plaintext.charAt(j);

j++;

}

}

}

public void decodeKey() {

String tempCiphertext = ciphertext.replaceAll("\\s+","");

int ciphertextLength = tempCiphertext.length();

int initialKeyLength = key.length();

String currentKey = key;

String decrypt = "";

int counter = 0;

if(mode == 3) {//autokey - standard

while(currentKey.length() < ciphertextLength) {

decrypt = "";

for(int i=0; i<currentKey.length(); i++) {

char k = currentKey.charAt(i);

char c = tempCiphertext.charAt(i);

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

for(int j=0; j<26; j++) {

if(c == vigenereSquare26[charToInteger(k)][j]) {

decrypt += (char) (j+97);

System.out.println("decrypt: " + decrypt);

break;

}

}

}

currentKey += decrypt.substring(counter);

counter+= initialKeyLength;

System.out.println("current key: " + currentKey);

if(currentKey.length() > ciphertextLength) {

currentKey = currentKey.substring(0, ciphertextLength);

}

}

key = currentKey;

} else { //autokey - extended

while(currentKey.length() < ciphertextLength) {

decrypt = "";

for(int i=0; i<currentKey.length(); i++) {

char k = currentKey.charAt(i);

char c = tempCiphertext.charAt(i);

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

for(int j=0; j<256; j++) {

if(c == vigenereSquare256[(int)(k)][j]) {

decrypt += (char) (j);

System.out.println("decrypt: " + decrypt);

break;

}

}

}

currentKey += decrypt.substring(counter);

counter+= initialKeyLength;

System.out.println("current key: " + currentKey);

if(currentKey.length() > ciphertextLength) {

currentKey = currentKey.substring(0, ciphertextLength);

}

}

key = currentKey;

}

}

private int charToInteger(char c) {

int temp = ((int)c) - 97;

return temp;

}

}

## **VigenereCipher.java**

Kelas ini mengimplementasikan menggabungkan kelas Engine.java dengan GUI.

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package gui;

import engine.Engine;

import java.io.File;

import javax.swing.JFileChooser;

/\*\*

\*

\* @author timothy.pratama

\*/

public class VigenereCipher extends javax.swing.JApplet {

private Engine engine = new Engine();

private int option;

private int display;

private void getOptions() {

if(standardRadio.isSelected()) {

option = 1;

} else if (extendedRadio.isSelected()) {

option = 2;

} else if(autokeyRadio.isSelected()){

option = 3;

} else {

option = 4;

}

if(normalRadio.isSelected()) {

display = 1;

} else if(removeRadio.isSelected()) {

display = 2;

} else {

display = 3;

}

}

/\*\*

\* Initializes the applet VigenereCipher

\*/

@Override

public void init() {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(VigenereCipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(VigenereCipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(VigenereCipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(VigenereCipher.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Set the class private attributes value \*/

option = 0;

display = 0;

/\* Create and display the applet \*/

try {

java.awt.EventQueue.invokeAndWait(new Runnable() {

public void run() {

initComponents();

resize(800, 520);

}

});

} catch (Exception ex) {

ex.printStackTrace();

}

}

/\*\*

\* This method is called from within the init() method to initialize the

\* form. WARNING: Do NOT modify this code. The content of this method is

\* always regenerated by the Form Editor.

\*/

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

ciphersVariant = new javax.swing.ButtonGroup();

ciphertextOption = new javax.swing.ButtonGroup();

buttonGroup1 = new javax.swing.ButtonGroup();

backgroundPanel = new javax.swing.JPanel();

title = new javax.swing.JLabel();

saveButtons = new javax.swing.JPanel();

version = new javax.swing.JLabel();

standardRadio = new javax.swing.JRadioButton();

extendedRadio = new javax.swing.JRadioButton();

autokeyRadio = new javax.swing.JRadioButton();

jSeparator1 = new javax.swing.JSeparator();

version1 = new javax.swing.JLabel();

normalRadio = new javax.swing.JRadioButton();

removeRadio = new javax.swing.JRadioButton();

groupRadio = new javax.swing.JRadioButton();

jSeparator2 = new javax.swing.JSeparator();

jLabel1 = new javax.swing.JLabel();

loadButton = new javax.swing.JButton();

saveButton = new javax.swing.JButton();

encryptButton = new javax.swing.JButton();

decryptButton = new javax.swing.JButton();

autokeyExtendedButton = new javax.swing.JRadioButton();

jPanel1 = new javax.swing.JPanel();

jLabel2 = new javax.swing.JLabel();

jScrollPane1 = new javax.swing.JScrollPane();

plaintextTextArea = new javax.swing.JTextArea();

jScrollPane2 = new javax.swing.JScrollPane();

keyTextArea = new javax.swing.JTextArea();

jLabel3 = new javax.swing.JLabel();

jPanel2 = new javax.swing.JPanel();

jScrollPane3 = new javax.swing.JScrollPane();

ciphertextTextArea = new javax.swing.JTextArea();

messageLabel = new javax.swing.JLabel();

setMaximumSize(new java.awt.Dimension(800, 500));

setPreferredSize(new java.awt.Dimension(800, 500));

backgroundPanel.setPreferredSize(new java.awt.Dimension(800, 500));

title.setFont(new java.awt.Font("Arial", 0, 24)); // NOI18N

title.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

title.setText("Vigenere Cipher");

saveButtons.setBorder(javax.swing.BorderFactory.createTitledBorder("Options"));

version.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N

version.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

version.setText("Vigenere Cipher's Variant");

ciphersVariant.add(standardRadio);

standardRadio.setSelected(true);

standardRadio.setText("Standard");

standardRadio.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

standardRadioActionPerformed(evt);

}

});

ciphersVariant.add(extendedRadio);

extendedRadio.setText("Extended");

extendedRadio.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

extendedRadioActionPerformed(evt);

}

});

ciphersVariant.add(autokeyRadio);

autokeyRadio.setText("Autokey(Standard)");

version1.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N

version1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

version1.setText("Ciphertext Format Options");

ciphertextOption.add(normalRadio);

normalRadio.setSelected(true);

normalRadio.setText("Normal");

ciphertextOption.add(removeRadio);

removeRadio.setText("Remove all whitespaces");

ciphertextOption.add(groupRadio);

groupRadio.setText("Group in 5 characters");

jLabel1.setFont(new java.awt.Font("Arial", 1, 12)); // NOI18N

jLabel1.setHorizontalAlignment(javax.swing.SwingConstants.CENTER);

jLabel1.setText("Save | Load | Encrypt | Decrypt");

loadButton.setText("Load");

loadButton.setCursor(new java.awt.Cursor(java.awt.Cursor.DEFAULT\_CURSOR));

loadButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

loadButtonActionPerformed(evt);

}

});

saveButton.setText("Save");

saveButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

saveButtonActionPerformed(evt);

}

});

encryptButton.setText("Encrypt");

encryptButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

encryptButtonActionPerformed(evt);

}

});

decryptButton.setText("Decrypt");

decryptButton.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

decryptButtonActionPerformed(evt);

}

});

ciphersVariant.add(autokeyExtendedButton);

autokeyExtendedButton.setText("Autokey(Extended)");

javax.swing.GroupLayout saveButtonsLayout = new javax.swing.GroupLayout(saveButtons);

saveButtons.setLayout(saveButtonsLayout);

saveButtonsLayout.setHorizontalGroup(

saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(version, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jSeparator1, javax.swing.GroupLayout.Alignment.TRAILING)

.addComponent(version1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jSeparator2, javax.swing.GroupLayout.Alignment.TRAILING)

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

.addGap(0, 0, Short.MAX\_VALUE)

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

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

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(saveButtonsLayout.createSequentialGroup()

.addComponent(saveButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 72, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(encryptButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 72, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGroup(saveButtonsLayout.createSequentialGroup()

.addComponent(loadButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 72, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addComponent(decryptButton, javax.swing.GroupLayout.PREFERRED\_SIZE, 72, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGap(29, 29, 29))

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

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(removeRadio)

.addComponent(groupRadio)

.addComponent(normalRadio))

.addGap(34, 34, 34))))

.addGroup(saveButtonsLayout.createSequentialGroup()

.addContainerGap()

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jLabel1, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addGroup(saveButtonsLayout.createSequentialGroup()

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(standardRadio)

.addComponent(extendedRadio))

.addGap(18, 18, 18)

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(autokeyRadio)

.addComponent(autokeyExtendedButton))

.addGap(0, 0, Short.MAX\_VALUE)))

.addContainerGap())

);

saveButtonsLayout.setVerticalGroup(

saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(saveButtonsLayout.createSequentialGroup()

.addContainerGap()

.addComponent(version)

.addGap(18, 18, 18)

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

.addGroup(saveButtonsLayout.createSequentialGroup()

.addComponent(standardRadio)

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

.addComponent(extendedRadio))

.addGroup(saveButtonsLayout.createSequentialGroup()

.addComponent(autokeyRadio)

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

.addComponent(autokeyExtendedButton)))

.addGap(31, 31, 31)

.addComponent(jSeparator1, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(version1)

.addGap(18, 18, 18)

.addComponent(normalRadio)

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

.addComponent(removeRadio)

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

.addComponent(groupRadio)

.addGap(18, 18, 18)

.addComponent(jSeparator2, javax.swing.GroupLayout.PREFERRED\_SIZE, 10, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel1)

.addGap(18, 18, 18)

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(saveButton)

.addComponent(encryptButton))

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

.addGroup(saveButtonsLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(loadButton)

.addComponent(decryptButton))

.addContainerGap(18, Short.MAX\_VALUE))

);

jPanel1.setBorder(javax.swing.BorderFactory.createTitledBorder("Input"));

jLabel2.setFont(new java.awt.Font("Dialog", 1, 11)); // NOI18N

jLabel2.setText("Plaintext");

plaintextTextArea.setColumns(20);

plaintextTextArea.setRows(5);

jScrollPane1.setViewportView(plaintextTextArea);

jScrollPane2.setVerticalScrollBarPolicy(javax.swing.ScrollPaneConstants.VERTICAL\_SCROLLBAR\_NEVER);

keyTextArea.setColumns(20);

keyTextArea.setRows(5);

jScrollPane2.setViewportView(keyTextArea);

jLabel3.setFont(new java.awt.Font("Dialog", 1, 11)); // NOI18N

jLabel3.setText("Key");

javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);

jPanel1.setLayout(jPanel1Layout);

jPanel1Layout.setHorizontalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addContainerGap()

.addGroup(jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane1)

.addComponent(jScrollPane2, javax.swing.GroupLayout.DEFAULT\_SIZE, 495, Short.MAX\_VALUE)

.addComponent(jLabel2)

.addComponent(jLabel3))

.addContainerGap())

);

jPanel1Layout.setVerticalGroup(

jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(jPanel1Layout.createSequentialGroup()

.addComponent(jLabel2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE)

.addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED\_SIZE, 101, javax.swing.GroupLayout.PREFERRED\_SIZE)

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

.addComponent(jLabel3)

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

.addComponent(jScrollPane2, javax.swing.GroupLayout.PREFERRED\_SIZE, 55, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addContainerGap())

);

jPanel2.setBorder(javax.swing.BorderFactory.createTitledBorder("Ciphertext"));

ciphertextTextArea.setColumns(20);

ciphertextTextArea.setRows(5);

jScrollPane3.setViewportView(ciphertextTextArea);

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(12, 12, 12)

.addComponent(jScrollPane3, javax.swing.GroupLayout.PREFERRED\_SIZE, 495, javax.swing.GroupLayout.PREFERRED\_SIZE))

);

jPanel2Layout.setVerticalGroup(

jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jScrollPane3, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED\_SIZE, 142, javax.swing.GroupLayout.PREFERRED\_SIZE)

);

messageLabel.setText(" ");

javax.swing.GroupLayout backgroundPanelLayout = new javax.swing.GroupLayout(backgroundPanel);

backgroundPanel.setLayout(backgroundPanelLayout);

backgroundPanelLayout.setHorizontalGroup(

backgroundPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(backgroundPanelLayout.createSequentialGroup()

.addGap(22, 22, 22)

.addGroup(backgroundPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(messageLabel, javax.swing.GroupLayout.PREFERRED\_SIZE, 177, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(title, javax.swing.GroupLayout.PREFERRED\_SIZE, 776, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(saveButtons, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(backgroundPanelLayout.createSequentialGroup()

.addGap(247, 247, 247)

.addGroup(backgroundPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))))

.addGap(2, 2, 2))

);

backgroundPanelLayout.setVerticalGroup(

backgroundPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(backgroundPanelLayout.createSequentialGroup()

.addComponent(title, javax.swing.GroupLayout.PREFERRED\_SIZE, 43, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(backgroundPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addComponent(saveButtons, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGroup(backgroundPanelLayout.createSequentialGroup()

.addGap(244, 244, 244)

.addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

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

.addComponent(messageLabel)

.addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))

);

getContentPane().add(backgroundPanel, java.awt.BorderLayout.CENTER);

}// </editor-fold>

private void standardRadioActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void extendedRadioActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void saveButtonActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fc = new JFileChooser("C:\\Users\\timothy.pratama\\Desktop\\Tucil 1 Kriptografi\\Vigenere Cipher");

int retval = fc.showSaveDialog(backgroundPanel);

if(retval == JFileChooser.APPROVE\_OPTION) {

File file = fc.getSelectedFile();

engine.saveFile(file.getAbsolutePath());

messageLabel.setText("Save Successful");

plaintextTextArea.setText(engine.getPlaintext());

}

else {

}

}

private void encryptButtonActionPerformed(java.awt.event.ActionEvent evt) {

getOptions();

engine.setPlaintext(plaintextTextArea.getText());

engine.setKey(keyTextArea.getText());

engine.setMode(option);

engine.setDisplay(display);

if(keyTextArea.getText().length()==0) {

messageLabel.setText("Key tidak boleh kosong");

} else {

engine.encrypt();

messageLabel.setText(" ");

String result = engine.getCiphertext();

ciphertextTextArea.setText(result);

}

}

private void decryptButtonActionPerformed(java.awt.event.ActionEvent evt) {

getOptions();

engine.setCiphertext(ciphertextTextArea.getText());

engine.setKey(keyTextArea.getText());

engine.setMode(option);

if(keyTextArea.getText().length()==0) {

messageLabel.setText("Key tidak boleh kosong");

} else {

engine.decrypt();

messageLabel.setText(" ");

String result = engine.getPlaintext();

plaintextTextArea.setText(result);

}

}

private void loadButtonActionPerformed(java.awt.event.ActionEvent evt) {

JFileChooser fc = new JFileChooser("C:\\Users\\timothy.pratama\\Desktop\\Tucil 1 Kriptografi\\Vigenere Cipher");

int retval = fc.showOpenDialog(backgroundPanel);

if(retval == JFileChooser.APPROVE\_OPTION) {

File file = fc.getSelectedFile();

engine.readFile(file.getAbsolutePath());

messageLabel.setText("Load Successful");

plaintextTextArea.setText(engine.getPlaintext());

}

else {

}

}

// Variables declaration - do not modify

private javax.swing.JRadioButton autokeyExtendedButton;

private javax.swing.JRadioButton autokeyRadio;

private javax.swing.JPanel backgroundPanel;

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.ButtonGroup ciphersVariant;

private javax.swing.ButtonGroup ciphertextOption;

private javax.swing.JTextArea ciphertextTextArea;

private javax.swing.JButton decryptButton;

private javax.swing.JButton encryptButton;

private javax.swing.JRadioButton extendedRadio;

private javax.swing.JRadioButton groupRadio;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel3;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel2;

private javax.swing.JScrollPane jScrollPane1;

private javax.swing.JScrollPane jScrollPane2;

private javax.swing.JScrollPane jScrollPane3;

private javax.swing.JSeparator jSeparator1;

private javax.swing.JSeparator jSeparator2;

private javax.swing.JTextArea keyTextArea;

private javax.swing.JButton loadButton;

private javax.swing.JLabel messageLabel;

private javax.swing.JRadioButton normalRadio;

private javax.swing.JTextArea plaintextTextArea;

private javax.swing.JRadioButton removeRadio;

private javax.swing.JButton saveButton;

private javax.swing.JPanel saveButtons;

private javax.swing.JRadioButton standardRadio;

private javax.swing.JLabel title;

private javax.swing.JLabel version;

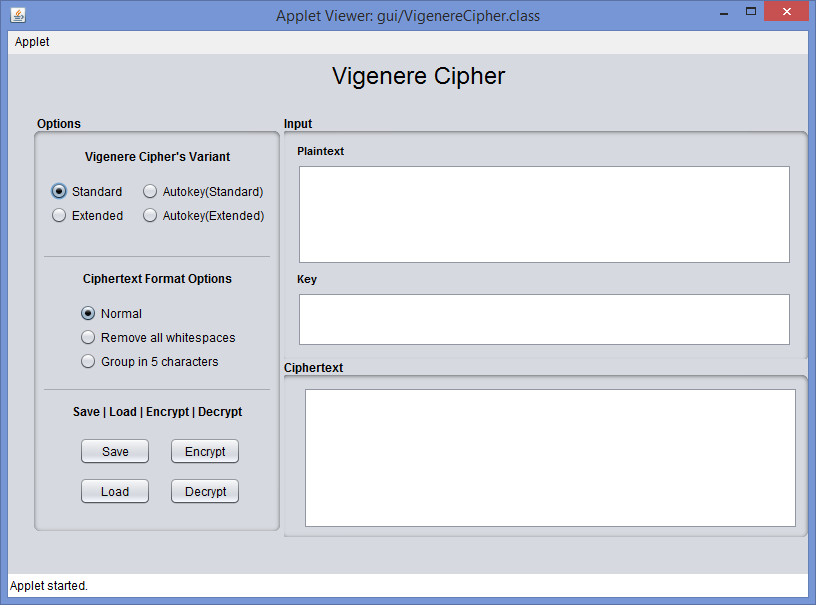
private javax.swing.JLabel version1;

// End of variables declaration

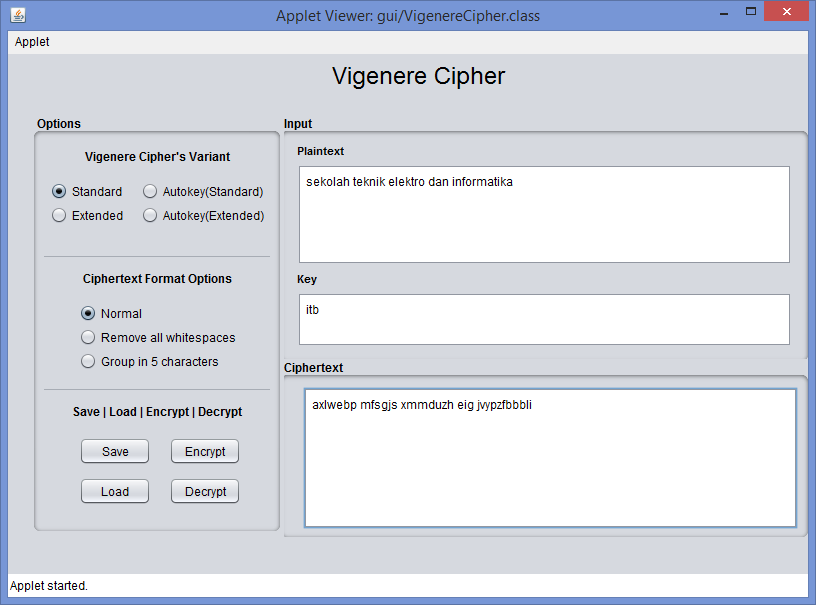
}

# Tampilan antarmuka program

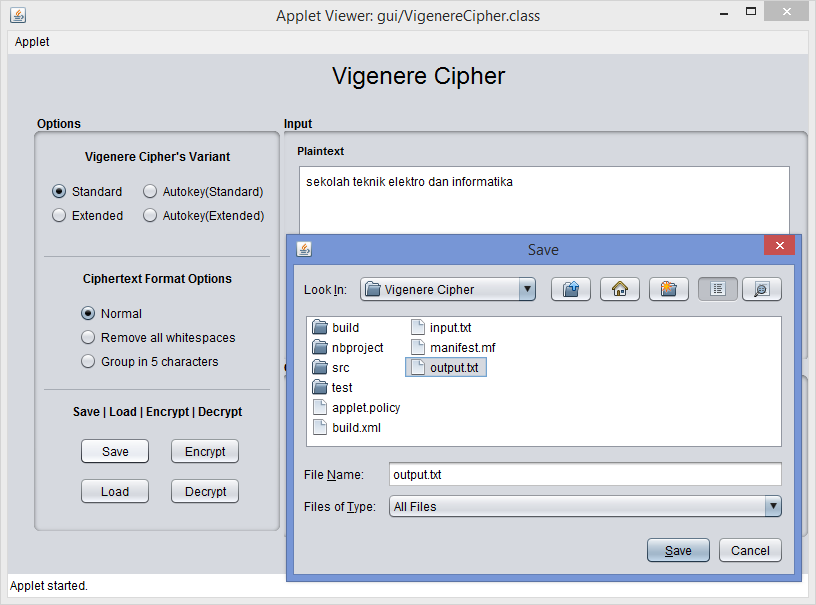
## Tampilan awal applet



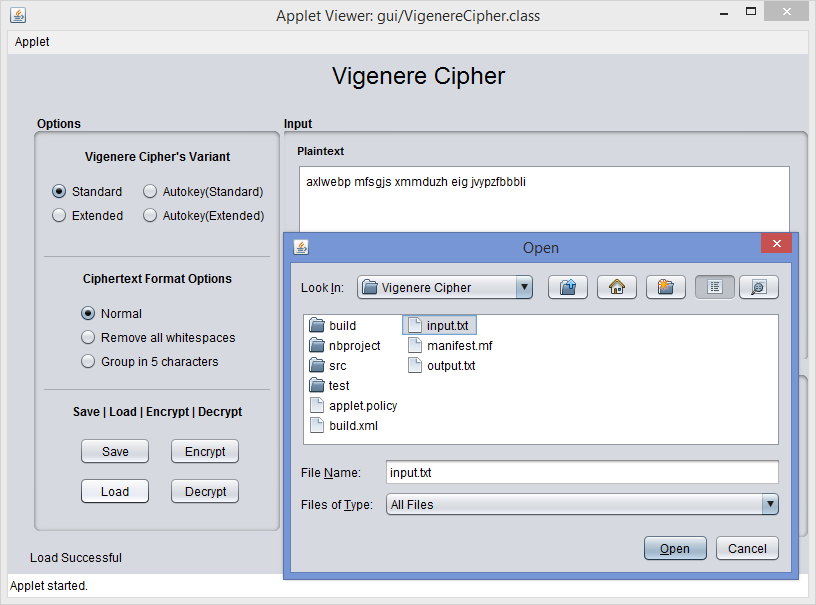
## Tampilan *encryption* dan *decryption*



## Save *ciphertext* ke sebuah *file* eksternal



## Tampilan *load* *plaintext* dari *file* eksternal



# Contoh *plaintext* dan *ciphertext*

|  |  |  |  |
| --- | --- | --- | --- |
| **Variant**  **Vigenere**  **Cipher** | **Plaintext** | **Key** | **Ciphertext** |
| Standard | sekolah teknik elektro dan informatika | itb | axlwebp mfsgjs xmmduzh eig jvypzfbbbli |
| We are currently opening opportunities for Indonesia undergraduate and master students to join our research projects in computer science | internship | ertvv pmyzt vgecf cwuqc obitf elbvx bvxww bjpvs waxwz nmult ztkeu hsamp vqfej gwyai cqxrk flvrd qahyi ewzmp zpati bblki avggf zhbbt zfvmv aul |
| Normally Mailman will remind you of your   mailing list passwords once every month although you can disable this if you prefer  This reminder will also include instructions on how to unsubscribe or change your account options  There is also a button on your options page that will email your current password to you | milis | zwcusxtj usutxif iqwt jqutvv kwf wx kwfz  emqwqfs ttal bidaoazoa gzkp mnqzj ugzbs idfpzcyt gzc umv oqkmjwm ltqd  qx kwf xjqnpz ltqd zwyqylwd ettd mtdw azkwcvq qyaldcnbaavd wf twh bg  gvdctekcqtq wc kzmvrm qacc iuowfvl axeqgza epwdm ta sxaz i tgbewf av  jwmd wabaavd xssm epsf ettd qulqd kwfz ugzcmff xlakiwcl la gzc |
| Extended |  |  |  |
|  |  |  |
|  |  |  |
| Standard, auto key |  |  |  |
|  |  |  |
|  |  |  |
| Extended, auto key |  |  |  |
|  |  |  |
|  |  |  |