

ALGORITMA SIMETRIS PADA APLIKASI ENKRIPSI FILE DENGAN MENGGUNAKAN J2SDK

ABSTRAK

Media komunikasi sebagai alat pengiriman data atau informasi menjadi hal yang banyak dilakukan mengingat waktu yang dibutuhkan untuk mengirim lebih cepat dan biaya yang dikeluarkan lebih murah dibandingkan dengan penyampaian secara lisan. Akan tetapi jika menyampaikan pesan melalui media komunikasi akan dapat dengan mudah data-data tersebut diketahui oleh orang-orang yang tidak berwenang jika data-data tersebut tidak dilindungi. Enkripsi adalah salah satu cara mencegah terjadinya hal-hal yang tidak diinginkan sehingga data tersebut tidak mudah untuk diketahui oleh orang lain dan file tidak dapat diakses oleh pihak-pihak yang tidak diinginkan. Penelitian enkripsi dibuat dengan menggunakan bahasa pemrograman J2SDK (Java 2 Standard Development Kit), yakni dengan memanfaatkan fasilitas-fasilitas yang telah disediakan oleh bahasa pemrograman J2SDK.

Kata Kunci : Aplikasi, Enkripsi, J2SDK

ABSTRACT

Communications media as a means of sending data or information becomes a lot done considering the time required to transmit faster and cheaper cost compared with oral delivery. However, if the message across through the medium of communication will be able to easily these data are known by people who are not authorized if data is not protected. Encryption is one way of preventing things that are not diinginkan so that data is not easy to be known by others and the file can not be accessed by parties that are not desirable. Encryption research created using programming languages ??J2SDK (Java 2 Standard Development Kit), ie, by utilizing the facilities provided by programming languages ??J2SDK.

Keywords: Application, Encryption, J2SDK

PENDAHULUAN

Berkembangnya Teknologi Informasi yang sangat pesat menimbulkan berbagai dampak dalam kehidupan manusia, baik dampak positif, maupun negatif. Contoh dampak positif yang ditimbulkan dan mempengaruhi kehidupan manusia adalah semakin mudahnya manusia dalam menjalankan berbagai aktivitas seperti dalam hal penyimpanan file-file penting. Manusia cukup menyimpannya dalam sebuah database dengan memberikan *password* sebagai salah satu teknik penyimpanan yang aman dengan tujuan file-file tersebut tidak diketahui oleh orang lain. Sedangkan dampak yang ditimbulkan adalah banyak pihak yang tidak bertanggung jawab mencoba untuk menerobos *password* dari file-file penting tersebut sehingga kemungkinan berhasil diketahui.

Untuk menghindari hal tersebut di atas maka aplikasi enkripsi file dirasa tepat guna mengamankan file-file penting tersebut walaupun *password* dapat diterobos oleh *hacker*. Aplikasi ini menggunakan bahasa pemrograman Java, mengingat bahasa Java memiliki keunggulan yang tidak dimiliki bahasa pemrograman lain seperti dapat berjalan di berbagai platform sistem operasi serta merupakan bahasa pemrograman yang berorientasi objek. Aplikasi ini akan merubah *extension* pada file yang ingin kita amankan. Netbeans digunakan sebagai editor mengingat Netbeans merupakan editor yang "pintar". Jika kita membuat suatu project Java yang terdiri dari beberapa file dengan Netbeans maka secara otomatis Netbeans akan

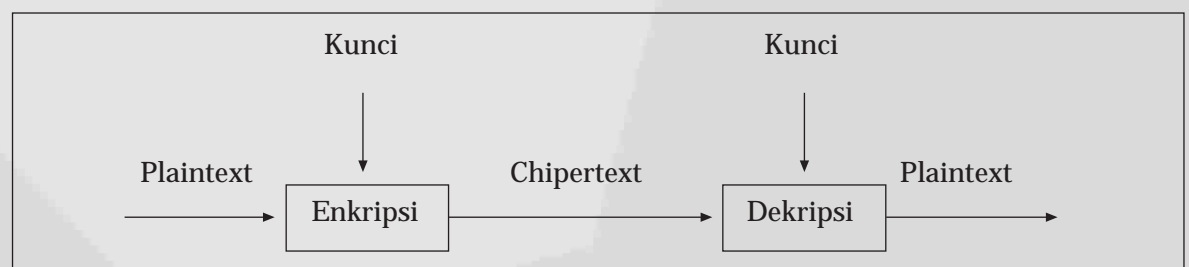
menggabungkannya ke dalam satu project sekaligus membuatnya satu *main class*. Berbeda dengan text editor yang lain yang harus kita buat secara manual.

Jenis algoritma kriptografi yang digunakan dalam penelitian ini adalah algoritma simetris atau disebut juga algoritma kriptografi konvensional yaitu algoritma yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses dekripsi. Pada proses enkripsi, data jelas (*plaintext*) ditransformasikan ke dalam bentuk data sandi (*ciphertext*) yang tidak dapat dikenali. Sementara pada proses dekripsi, data sandi (*ciphertext*) ditransformasikan ke dalam bentuk data jelas (*plaintext*) yang dapat dikenali kembali. Proses enkripsi-dekripsi algoritma kriptografi simetris dapat dilihat pada gambar 1. dibawah ini :

Membuat class-class model, yaitu class-class yang akan direpresentasikan menjadi tabel-tabel dalam database yang terdiri dari empat class yaitu EncFile, Encrypter, In Password, dan proses, (3) Menggunakan algoritma kriptografi simetris (konvensional), dengan metode DES (Data Encryption Standar).

HASIL DAN PEMBAHASAN

Aplikasi Enkripsi ini terdiri dari beberapa form (tampilan), yaitu form utama sebagai form yang pertama kali akan muncul pada saat pertama kali kita menjalankan program ini. Form *Browse*, sebagai form untuk mencari/memilih file yang akan di *enkripsi* atau di *dekripsi*. Form *password*, sebagai form yang akan muncul



Gambar 1. Algoritma kriptografi simetris

Metode dalam algoritma simetris yang dipakai adalah DES (Data Encryption Standar).

METODE PENELITIAN

Metode penelitian yang digunakan adalah (1) Membuat aplikasi berbasis Java yang menerapkan teknologi pengamanan data serta Netbeans sebagai editor, (2)

pada saat kita akan melaksanakan proses enkripsi atau dekripsi, form ini berguna sebagai kunci (key) dari proses enkripsi atau dekripsi. Form *About*, sebagai form yang berisikan informasi pembuat program dan versi dari program, dan beberapa form mengenai informasi jika pemakai melakukan kesalahan atau sukses dalam pemakaian. Jika kita telah meng-*enkripsi* suatu file dengan

Yuli Maharetta Arianti

Jurusan Sistem Informasi
Universitas Gunadarma
Jl. Margonda Raya 100
Pondok Cina Depok 16424
yuli_maharetta@gunadarma.ac.id

password yang telah kita tentukan, maka jika kita ingin men-*dekripsi* kembali kembali file tersebut kita harus menggunakan password yang sama dengan password yang kita masukkan pada saat kita melakukan enkripsi.

Algoritma Program

Jenis algoritma program yang digunakan adalah algoritma simetris yang menggunakan kunci untuk proses enkripsi sama dengan kunci untuk proses deskripsi serta metode dalam algoritma simetris yang dipakai adalah DES (Data Encryption Standar).

Algoritma enkripsi file adalah (1) meng-*encrypt* password yang telah diinput, dimana nilai bit dari tiap-tiap karakter pada password dioperasikan menggunakan operator XOR, (2) menciptakan file baru atau output yang namanya sama dengan nama file input, tetapi penamaan file output ditambahkan akhiran *.cry*; (3) pada file output tersebut diisi dengan nilai password yang telah diencrypt, dan (4) membaca karakter isi file input dengan mengoperasikan bit dari tiap karakter dengan menggunakan operator XOR dengan nilai integer hasil penggabungan kunci (*key*) tetap.

Algoritma deskripsi file adalah (1) melakukan encrypt yang telah diinputkan (langkah 1 pada proses enkripsi file), (2) menciptakan file baru atau output yang sama dengan nama file yang telah diencrypt, bedanya file output ini didapat dari pengurangan 4 karakter terakhir pada saat diinput, (3) melakukan pembacaan karakter isi file input sebanyak panjang password yang telah di encrypt pada langkah awal, kemudian dilakukan pemeriksaan apakah password pada file input sesuai dengan password pada saat encrypt tadi. Jika beda, maka proses dihentikan dengan mengeluarkan pesan *Error*; tetapi jika benar, maka proses akan dilanjutkan, (4) apabila pembacaan karakter sesuai pada file input, kemudian akan dilakukan proses descrypt pada karakter-karakter tersebut dengan menggunakan metode yang sama pada saat proses encrypt, dan (4) karakter yang telah didescrypt dituliskan pada file output.

Penulisan Program

Di dalam program enkripsi ini terdapat 4 (empat) buah Java Class yang saling berhubungan dalam setiap tahapan prosesnya yaitu (1) *EncFile* atau disebut sebagai Main Class, (2) *Encrypter*; (3) *InPassword*, (4) *Proses*.

Encrypter File (*menu utama*)

Pembuatan nama Java Class dengan cara penulisan kode program pada masing-masing Java Class dengan langkah-langkah sebagai berikut :

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.io.*;
import EncryptDecrypt.InPassword;
import EncryptDecrypt.Encrypter;
import javax.swing.
```

```
filechooser.FileFilter;
```

Javax.swing dan *java.awt* digunakan dalam pemrograman yang menggunakan GUI (Grafical User Interface), kedua package ini telah disediakan oleh Java dalam setiap produknya. *Java.io* digunakan untuk menangani proses input output.

Buat class turunan yang dinamakan *FileNameFilter* yang merupakan turunan dari class *FileFilter* dengan kode di bawah ini :

```
import javax.swing.filechooser.
FileFilter;
```

Lakukan pengecekan tipe file dengan semua extention (All Files)

```
class FileNameFilter extends
FileFilter
{
public boolean accept(File f){
//Untuk directory tidak perlu filter
if (f.isDirectory())
{
return true;
}
}
```

Lakukan pengecekan tipe file dengan extention *.cry*.

```
//Lakukan Filter untuk file
String fileName =
f.getName().toLowerCase();
if (fileName.endsWith(".cry"))
return true;
return false;
}
public String getDescription(){
return "File Encrypter(*.cry)";
}
};
```

Buat Label pada *interface* Menu Utama.

```
class EncFile extends JFrame
implements ActionListener {
boolean Crypted=true;
JLabel LabFName = new
JLabel("File Name :");
```

Buat TextBox pada *interface* Menu Utama.

```
JTextField TxtFName = new
JTextField();
```

Buat Command Button pada *interface* Menu Utama.

```
JButton ButtBrowse = new
JButton("Browse");
JButton ButtEnc = new
JButton("Encrypt File");
JButton ButtDec = new
JButton("Decrypt File");
JButton ButtAbout = new
JButton("About");
JButton ButtExit = new
JButton("Exit");
JFileChooser choose = new
JFileChooser();
```

Tentukan posisi *Interface* Menu Utama agar terletak ditengah monitor dengan perintah sebagai berikut :

```
FlowLayout layout = new
FlowLayout(FlowLayout.CENTER);
```

Buat Event pada Button. Contoh bila menekan Alt+A, maka tampilan akan sama bila menekan Button Tentang.

```
ButtEnc.addActionListener(this);
ButtEnc.setMnemonic(KeyEvent.VK_E);
ButtDec.addActionListener(this);
ButtDec.setMnemonic(KeyEvent.VK_D);
ButtBrowse.addActionListener(this);
ButtBrowse.setMnemonic(KeyEvent.VK_B);
ButtAbout.addActionListener(this);
ButtAbout.setMnemonic(KeyEvent.VK_A);
ButtExit.addActionListener(this);
ButtExit.setMnemonic(KeyEvent.VK_X);
```

Lakukan pemberian Name pada Objek-objek yang digunakan.

```
getContentPane().add(LabFName);
getContentPane().add(TxtFName);
getContentPane().add(ButtBrowse);
getContentPane().add(ButtEnc);
getContentPane().add(ButtDec);
getContentPane().add(ButtAbout);
getContentPane().add(ButtExit);
```

Tutup Frame dengan perintah di bawah ini

```
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Tampilkan *Interface* bila user menekan Button Browse.

```
public void
actionPerformed(ActionEvent e) {
int result;
if (e.getActionCommand().equals("Browse"))
{
choose.addChoosableFileFilter(new
FileNameFilter());
result=choose.showOpenDialog(this);
if (result ==
JFileChooser.APPROVE_OPTION)
{
File file = choose.getSelectedFile();
TxtFName.setText(file.getAbsolutePath());
}
}
}
```

Tutup aplikasi dengan perintah

```
else if (e.getActionCommand().equals("Exit"))
System.exit(0);
```

Lakukan pemanggilan terhadap File Encrypt dan InPassword.

```
else if (e.getActionCommand().equals("Encrypt File"))
{
Crypted = true;
if (new
Encrypter("").CekFile(TxtFName.getText()))
```



```

{
new InPassword("Password to
Encrypt File",

TxtFName.getText(),Crypted);
}
}

```

Lakukan pemanggilan terhadap File Decrypt dan InPassword

```

else if
(e.getActionCommand().equals
("Decrypt File"))
{
Crypted = false;
if (new
Encrypter("").CekFile(TxtFName.
getText()))
{
new InPassword("Input Password
to Decrypt File",
TxtFName.getText(),Crypted);
}
}
}

```

Ketik kode program dibawah ini dimana merupakan main Class dari Java, bila skrip ini tidak ada maka program tidak dapat dijalankan.

```

public static void main(String args[])
{
new EncFile("Encrypter File");
}
}

```

Encrypter

Pada *syntax* Encrypter berisi proses enkripsi file dimana nanti akan ada proses memasukkan password yang menandakan proses berlangsung.

Javax.swing dan *java.awt* merupakan package Java yang digunakan dalam pemrograman menggunakan GUI (Grafical User Interface), kedua package ini telah disediakan oleh Java dalam setiap produknya. *Java.io* merupakan package Java yang digunakan untuk menangani proses input output.

```

package EncryptDecrypt;
import java.io.*;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

```

Kode Program dibawah ini berfungsi sebagai pendeklarasian variabel.

```

private boolean crypt;
private String Pass,NewFile;
static final String key = "Encrypt";
static int Len,IntValue,mKey;
/**
* Constructs dari Encrypter.
* param aKey merupakan key
*/
public Encrypter(String aKey) {
mKey = KeyValue(aKey);

```

Lakukan pengecekan terhadap value yang dimasukkan. Program akan mengecek panjang password dan akan menyimpannya di IntValue.

```

private static int KeyValue(String
str)
{
IntValue = 0;
Len = str.length();
for (int i=Len-1;i>=0;i--)
{
IntValue = (IntValue * Len) +
(byte)(str.charAt(i));
if (IntValue>255) IntValue/=Len;
}
return (IntValue);
}

```

Lakukan proses enkripsi dengan memasukkan data yang akan diubah dengan menggunakan metode XOR(^). Proses berlangsung dengan memanipulasi data string menjadi Byte, dan kemudian dikembalikan lagi menjadi string.

```

private static String
encryptPassword(String str)
{
int KeyChar;
str = str + "$";
StringBuffer sb = new StringBuffer
(str);
int lenStr = str.length();
int lenKey = key.length();
for (int i = 0, j = 0; i < lenStr; i++,
j++)
{
if (j >= lenKey) j = 0;
sb.setCharAt(i, (char)(str.charAt(i)
^ key.charAt(j)));
str = sb.toString();
KeyChar = mKey ^
(byte)(str.charAt(i));
sb.setCharAt(i, (char)(KeyChar));
}
return sb.toString();
}

```

Rubah semua file yang ber-*extention* *.cry*.

```

public void encryptFile(String
inFile,String Password)
throws IOException {
InputStream in = null;
OutputStream out = null;
String outFile = inFile + ".cry";
Pass = encryptPassword(Password);
try {
in = new BufferedInputStream(new
FileInputStream(inFile));
out = new
BufferedOutputStream(new
FileOutputStream(outFile));

encryptStream(in, out);
} finally {
if (in != null) {
in.close();
}
if (out != null) {
out.flush();
out.close();
}
}
}
}

```

Lakukan pemrosesan terhadap input output file.

```

in = new BufferedInputStream(new

```

```

FileInputStream(inFile));
out = new
BufferedOutputStream(new
FileOutputStream(outFile));

```

Tampilkan pesan error, jika dalam poses enkripsi atau deskripsi tidak terdapat file yang akan diproses.

```

public boolean CekFile(String inFile)
{
boolean Stat = true;
try{
FileReader CheckedFile = new
FileReader(inFile);
}
catch(java.io.FileNotFoundException
Exception e){
JOptionPane.showMessageDialog(
null, "Operation File Failed\n" +
"Check file and attribute",
"Error",JOptionPane.ERROR_MES
SAGE);
Stat = false;
}
return (Stat);
}
}

```

Lakukan proses dekripsi dimana data yang dimasukkan akan diubah dengan menggunakan metode XOR(^). Proses berlangsung dengan memanipulasi Data string menjadi Byte, dan kemudian dikembalikan lagi menjadi string. Ini merupakan kebalikan dari proses enkripsi.

```

public void decryptFile(String
inFile,String Password)
throws IOException {
InputStream in = null;
OutputStream out = null;
String outFile =
inFile.substring(0,inFile.length()-4);
Pass = encryptPassword(Password);
try {
in = new BufferedInputStream(new
FileInputStream(inFile));
out = new
BufferedOutputStream(new
FileOutputStream(outFile));
decryptStream(in, out);
}finally {
if (in != null) {
in.close();
}
if (out != null) {
out.flush();
out.close();
}
}
}
}

```

Tampilkan pesan "Encrypter File Succeed".

```

public void
encryptStream(InputStream in,
OutputStream out)
throws IOException {
boolean done = false;
for (int i = 0;i<Pass.length();i++)
{
out.write((byte)Pass.charAt(i));
}
while (!done) {
int next = in.read();
if (next == -1) {
done = true;
JOptionPane.showMessageDialog(

```

```

null, "Encrypter File Succeed",
"Succeed",JOptionPane.INFORMA
TION_MESSAGE);
} else {
byte b = (byte)next;
byte c = encrypt(b);
out.write(c);
}
}
}

```

Tampilkan pesan "Decrypter File Succeed" dan pesan "Wrong Password", yang dikarenakan kesalahan memasukkan password.

```

public void
decryptStream(InputStream in,
OutputStream out)
throws IOException {
boolean done = false;
String textPass = "";
for (int i = 0; i < Pass.length(); i++)
textPass = textPass +
(char)(in.read());

if (textPass.equals(Pass))
{
while (!done) {
int next = in.read();
if (next == -1) {
done = true;
JOptionPane.showMessageDialog(
null, "Decrypter File Succeed",
"Succeed",JOptionPane.INFORMA
TION_MESSAGE);
} else {
byte b = (byte)next;
byte c = encrypt(b);
out.write(c);
}
}
} else
{
JOptionPane.showMessageDialog(
null, "Wrong Password",
"Process
Failed",JOptionPane.WARNING_
MESSAGE);
}
}
}

```

Syntax ini berfungsi sebagai penampungan nilai string yang telah dimanipulasi menjadi byte, kemudian nilai byte dioperasikan dengan operator XOR kemudian disimpan ke variabel b, kemudian variabel b dikembalikan ke file proses.

```

public byte encrypt(int b) {
String KeyStr = "";
int LenKey = key.length();
for (int i=0; i<LenKey; i++)
{
KeyStr = KeyStr + key.charAt(i);
b = b ^ KeyValue(KeyStr);
}
return (byte)(b ^ mKey);
}
public static void main(String[] args)
throws IOException {
// new Encrypter("Febri
").encryptFile("abc.txt", "annur");
}
}

```

InPassword

Pada *syntax* InPassword berisikan tentang proses penginputan password yang akan menandakan proses enkripsi atau dekripsi berlangsung.

Javax.swing dan *java.awt* merupakan package Java yang digunakan dalam pemrograman menggunakan GUI (Grafical User Interface), kedua package ini telah disediakan oleh Java dalam setiap produknya. *Java.io* merupakan package Java yang digunakan untuk menangani proses input output.

```

package EncryptDecrypt;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import EncryptDecrypt.Encrypter;
import java.io.*;

```

Kode program dibawah ini digunakan untuk membuat tampilan pada Form Password, dimana pada Form Password tersebut nantinya akan terdapat tampilan seperti label, button.

```

public class InPassword extends
JFrame implements ActionListener
{
JLabel label = new JLabel("Input
your password: ");
JPasswordField pass = new
JPasswordField(20);
JButton ButtOk = new
JButton("OK");
JButton ButtCancel = new
JButton("Cancel");
String FName;
boolean Crypted;

```

Kode dibawah ini di gunakan untuk membuat label di Button.

```

public InPassword(String title,String
FileName,boolean encrypted) {
super(title);
FName = FileName;
Crypted = encrypted;
ButtOk.addActionListener(this);
ButtOk.setMnemonic(KeyEvent.VK
_O);
ButtCancel.addActionListener(this);
ButtCancel.setMnemonic(KeyEvent
.VK_C);
getContentPane().setLayout(new
FlowLayout(FlowLayout.CENTER));
getContentPane().add(label);
getContentPane().add(pass);
getContentPane().add(ButtOk);
getContentPane().add(ButtCancel);

```

Kode program dibawah ini digunakan untuk menutup frame saat menekan tombol close, dan akan kembali ke menu utama.

```

setDefaultCloseOperation(JFrame.
DISPOSE_ON_CLOSE);
setSize(280,110);
show();
}

```

Cek nilai yang terdapat pada text box password, bahwa textbox password telah terisi.

```

public void actionPerformed
(ActionEvent e){

```

```

if
(e.getActionCommand().equals
("OK"))
{
try{
String Pwd = new
String(pass.getPassword());
if (!Crypted && Pwd.length() != 0)
{
dispose();
new Encrypter(Pwd).decryptFile
(FName,Pwd);
}
}
}

```

Kode Program dibawah ini berisikan pesan yang akan ditampilkan bila dalam melakukan proses enkripsi atau deskripsi user tidak memasukkan password. Pesan yang ditampilkan adalah "Password Have to be filled".

```

else if (Crypted && Pwd.length
() != 0)
{
dispose();
new Encrypter(Pwd).encryptFile
(FName,Pwd);
}
else
JOptionPane.showMessageDialog(
this, "Password Harus Diisi!",

```

```

"EmptyPassword",JOptionPane.INFO
RMATION_MESSAGE);

```

Syntax ini menampilkan pesan yang menyatakan bahwa password yang dimasukkan pada proses dekripsi tidak sesuai dengan password yang dilakukan pada saat enkripsi.

```

catch(IOException err){
JOptionPane.showMessageDialog(
null, "File fail operated \n"
+ " Check file and
attribute", "Error",JOption
Pane.ERROR_MESSAGE);
dispose();
}
}
else
dispose();
}
}

```

Proses

Pada *syntax* Proses ini, berisikan tentang proses-proses yang dilakukan untuk melakukan perintah enkripsi, deskripsi, pencarian file dan memasukkan password.

Javax.swing dan *java.awt* merupakan package Java yang digunakan dalam pemrograman menggunakan GUI (Grafical User Interface), kedua package ini telah disediakan oleh Java dalam setiap produknya. *Java.io* merupakan package Java yang digunakan untuk menangani proses input output.

```

package EncryptDecrypt;
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import EncryptDecrypt.Encrypter;
import java.io.*;

```

Buat label dan panel pada form password denan perintah di bawah ini

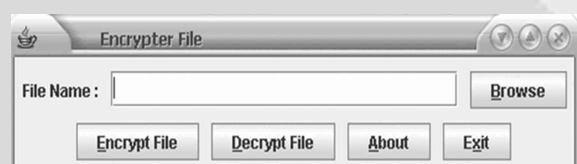

```

public class Proses extends JFrame
implements ActionListener {
JLabel label = new JLabel(" Input
your password: ");
JPanel panel = new JPanel(new
BorderLayout());
JLabel labPersen = new
JLabel("0%");;
String FName;
boolean Crypted;
Buat event handler.
public Proses(String title,long LenF)
{
super(title);
getContentPane().setLayout(new
FlowLayout(FlowLayout.CENTER));
getContentPane().add(label);
setDefaultCloseOperation(JFrame.
DISPOSE_ON_CLOSE);
setSize(180,80);
show();
for (long i=0;i<LenF;i++)
{
for (int j=0;j<=100000000;j++)
{
labPersen.setText(" "+i+"%");
getContentPane().add(labPersen);
}
dispose();
}
public void
actionPerformed(ActionEvent e){
}
}

```

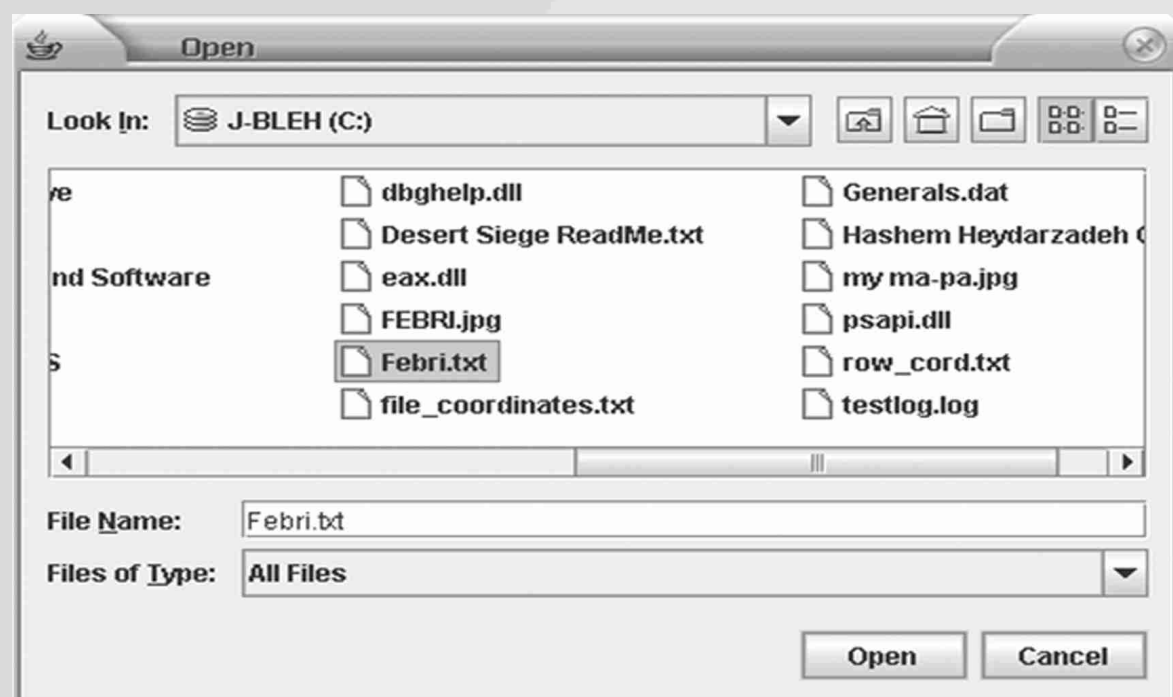
Menjalankan Program

Untuk menjalankan aplikasi Enkripsi File harus menjalankan program NETBEANS 5.0 terlebih dahulu, lalu pada project encrypt klik kanan > Build > Run. Maka akan tampil tampilan menu utama yaitu Form Encrypter File seperti ini terlihat pada gambar 2 di bawah ini :



Gambar 2. Form Encrypter File (Menu Utama)

Setelah tampilan diatas muncul, apabila kita tekan tombol *Browse*, maka akan muncul tampilan seperti terlihat pada gambar 3.



Gambar 3 Form Browse

Kemudian pilih file yang akan di enkripsi dan tekan tombol *Open*, maka akan muncul tampilan menu utama



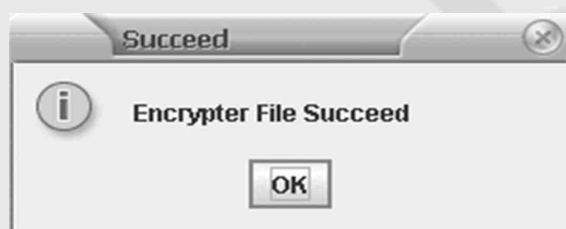
Gambar 4 Form Menu Utama (File siap di Encrypt)

Setelah menu utama kembali muncul, tekan tombol *Encrypt File*, maka akan muncul form password to encrypt seperti dibawah ini



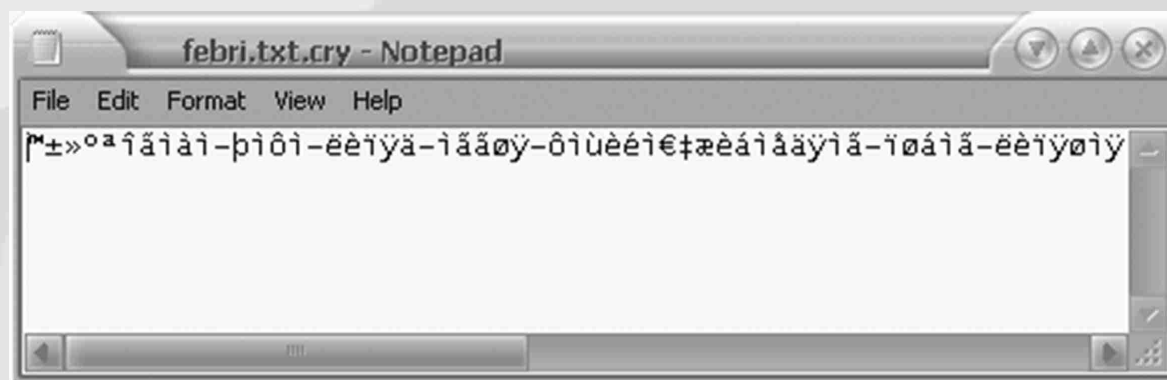
Gambar 5 Form Input Password to Encrypt File

Setelah kita masukkan password, maka proses enkripsi akan berjalan dan jika sukses akan muncul form seperti dibawah ini



Gambar 6 Form Encrypter File Succeed

Kemudian akan muncul tampilan File setelah mengalami proses Enkripsi seperti yang terlihat pada gambar 7. di bawah ini



Gambar 7. Tampilan File setelah di Enkripsi

SIMPULAN

Mengingat banyaknya pihak yang tidak bertanggung jawab yang ingin melihat file penting kita meskipun telah diberi *Password*, maka aplikasi enkripsi file merupakan salah satu alternative pencegahan terhadap jenis kejahatan tersebut. Java digunakan sebagai bahasa pemrograman dalam aplikasi ini mengingat bahasa Java memiliki keunggulan yang tidak dimiliki bahasa pemrograman lain seperti dapat berjalan di berbagai platform sistem operasi serta merupakan bahasa pemrograman yang berorientasi objek. Sebagai editor dipilih Netbeans mengingat Netbeans merupakan editor yang "pintar". Jika kita membuat suatu project Java yang terdiri dari beberapa file dengan Netbeans maka secara otomatis Netbeans akan menggabungkannya ke dalam satu project sekaligus membuatnya satu *main class*

DAFTAR PUSTAKA

Andi, *Memahami Model Enkripsi dan Security Data*, Yogyakarta, 2003

Harvey and Paul, Harvey, *JAVA How to Program Fourth Edition*, Prentice Hall, Inc, New Jersey, 2002

Indrajani, et all, *Pemrograman Berorientasi Objek dengan Java*, PT. Alex Media Komputindo, Jakarta, Desember 2003

Rangsang, Purnama, *Tuntunan Pemrograman Java Jilid 1*, Prestasi Pustaka Publisher, Jakarta, 2001

Rangsang, Purnama, *Tuntunan Pemrograman Java Jilid 2*, Prestasi Pustaka Publisher, Jakarta, 2003

Terrazas, Alejandro, et all, *Java™ Media APIs: Cross-Platform Imaging, Media, and Visualization eBook-Lib*, Sams Publishing, Arizona, 2002

Sumin, Agus, dan Suryadi, *Pengantar Algoritma dan Pemrograman*, Gunadarma, Jakarta, 1994

Raharjo, Budi, Haryono, Arif dan Heryanto, Imam, *Mudah Belajar JAVA*, Informatika, Bandung, 2007