**Tugas Kecil 1**

**Vigenere Cipher**

**IFxxxx Kriptografi**

Disusun Oleh :

Rikysamuel / 13512089



**Program Studi Teknik Informatika - Institut Teknologi Bandung**

**Jl. Ganesha 10, Bandung 40132**

# Source Code

/\*

\* 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 vigenerecipher;

import java.io.FileNotFoundException;

import java.io.FileReader;

import java.util.Scanner;

/\*\*

\*

\* @author Rikysamuel

\*/

public class VigenereCipher {

private String key;

private String str;

private String encStr;

public String getEncStr() {

return encStr;

}

public String getStr() {

return str;

}

public String getKey() {

return key;

}

public void setEncStr(String encStr) {

this.encStr = encStr;

}

public void setKey(String key) {

if (key.length()>25){

this.key = key.substring(0,25);

} else{

this.key = key;

}

}

public void setStr(String str) {

this.str = str;

}

public VigenereCipher(){

}

// Baca file dari path tertentu

public String FileReader(String filename) throws FileNotFoundException{

Scanner input = new Scanner(new FileReader(filename));

str = null;

while (input.hasNext()){

if (str == null){

str = input.next().toUpperCase();

} else{

str = str.concat(" ");

str = str.concat(input.next()).toUpperCase();

}

}

return str;

}

// Convert nilai ke 26 huruf alfabet, output huruf kapital

public char convToAlph(int in){

if(in<1){ // in < 'A'

in = in + 26;

}

if(in>26){ // in > 'Z'

in = in - 26;

}

in = in + 64;

char ch;

ch = (char)in;

return ch;

}

// Covert suatu char ke nilai tertentu, masukan huruf kapital.

public int convToInt(char ch){

return (int)ch - 64;

}

// Convert ke ASCII

public char convToASCII(int in){

char ch;

ch = (char)in;

return ch;

}

// Convert dari ASCII char ke integer

public int convToIntASCII(char ch){

return (int)ch;

}

// Enkripsi karakter dengan Standard VigenereCipher

public char Standard(char in, char key){

int val = ((convToInt(in) + convToInt(key)) % 26) - 1;

return convToAlph(val);

}

// Decripsi karakter dengan Standard VigenereCipher

public char decStandard(char in, char key){

int val = ((convToInt(in) - convToInt(key)) % 26) + 1;

return convToAlph(val);

}

// Enkripsi karakter dengan Extended VigenereCipher

public char Extended(char in, char key){

int val = ((convToIntASCII(in) + convToIntASCII(key)) % 256) -1;

return convToASCII(val);

}

public char decExtended(char in, char key){

int val = ((convToIntASCII(in) - convToIntASCII(key)) % 256) + 1;

return convToASCII(val);

}

// Generate key supaya paling tidak panjangnya sama dengan panjang kalimat

public String genKey(){

String strnospace = str.replace(" ", "");

String secKey = key;

while (secKey.length() < strnospace.length()){

secKey = secKey + key;

}

key = secKey;

return key;

}

// Memberikan spasi pada hasil enkripsi sesuai dengan plain text

public void encSpace(){

StringBuilder sb = new StringBuilder();

// jika encStr tidak null, sb = encStr

if (encStr != null){

sb.insert(0, encStr);

}

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

if(str.charAt(i)== ' '){

sb.insert(i, " ", 0, 1);

}

}

encStr = sb.toString();

}

// Memberikan spasi pada hasil enkripsi per 4 huruf

public void encSpace5(){

int j = 1;

String newStr = null;

// jika encStr tidak null, sb = encStr

if (encStr != null){

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

if (newStr == null){

newStr = String.valueOf(encStr.charAt(i));

}

else{

if (j<5){

j++;

newStr = newStr.concat(String.valueOf(encStr.charAt(i)));

} else{

newStr = newStr.concat(" ");

j = 0;

i--;

}

}

}

}

encStr = newStr;

}

// Proses enkripsi suatu kalimat dengan VigenereCipher Standard

public String processStandard(){

String strnospace = str.replace(" ", "");

String enc = null;

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

if (enc == null){

enc = String.valueOf(Standard(strnospace.charAt(i),key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(Standard(strnospace.charAt(i),key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

// Proses dekripsi suatu kalimat dengan VigenereCipher Standard

public String processDecStd(){

String strnospace = str.replace(" ", "");

String enc = null;

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

if (enc == null){

enc = String.valueOf(decStandard(strnospace.charAt(i),key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(decStandard(strnospace.charAt(i),key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

// Proses enkripsi suatu kalimat dengan VigenereCipher Extended

public String processExtended(){

String strnospace = str.replace(" ", "");

String enc = null;

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

if (enc == null){

enc = String.valueOf(Extended(strnospace.charAt(i),key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(Extended(strnospace.charAt(i),key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

// Proses enkripsi suatu kalimat dengan VigenereCipher Extended

public String processDecExtended(){

String strnospace = str.replace(" ", "");

String enc = null;

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

if (enc == null){

enc = String.valueOf(decExtended(strnospace.charAt(i),key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(decExtended(strnospace.charAt(i),key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

// Proses enckripsi suatu kalimat dengan Autokey VeigenereCipher

public String processAKStandard(){

String strnospace = str.replace(" ", "");

String enc = null;

String \_key = null;

if (key!=null){

\_key = key.concat(strnospace);

}

if (\_key != null){

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

if (enc == null){

enc = String.valueOf(Standard(strnospace.charAt(i),\_key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(Standard(strnospace.charAt(i),\_key.charAt(i))));

}

}

}

encStr = enc;

return encStr;

}

// Proses enckripsi suatu kalimat dengan Autokey VeigenereCipher

public String processDecAKStandard(){

String strnospace = str.replace(" ", "");

String enc = null;

String \_key = null;

if (key != null){

\_key = key;

}

if (\_key != null){

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

if (enc == null){

enc = String.valueOf(decStandard(strnospace.charAt(i),\_key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(decStandard(strnospace.charAt(i),\_key.charAt(i))));

}

\_key = \_key.concat(String.valueOf(decStandard(strnospace.charAt(i),\_key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

// Proses enckripsi suatu kalimat dengan Autokey VeigenereCipher

public String processAKExtended(){

String strnospace = str.replace(" ", "");

String enc = null;

String \_key = null;

if (key!=null){

\_key = key.concat(strnospace);

}

System.out.println(\_key);

if (\_key != null){

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

if (enc == null){

enc = String.valueOf(Extended(strnospace.charAt(i),\_key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(Extended(strnospace.charAt(i),\_key.charAt(i))));

}

}

}

encStr = enc;

return encStr;

}

// Proses enckripsi suatu kalimat dengan Autokey VeigenereCipher

public String processDecAKExtended(){

String strnospace = str.replace(" ", "");

String enc = null;

String \_key = null;

if (key != null){

\_key = key;

}

if (\_key != null){

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

if (enc == null){

enc = String.valueOf(decExtended(strnospace.charAt(i),\_key.charAt(i)));

} else{

enc = enc.concat(String.valueOf(decExtended(strnospace.charAt(i),\_key.charAt(i))));

}

\_key = \_key.concat(String.valueOf(decExtended(strnospace.charAt(i),\_key.charAt(i))));

}

}

encStr = enc;

return encStr;

}

/\*\*

\* @param args the command line arguments

\* @throws java.io.FileNotFoundException

\*/

public static void main(String[] args) throws FileNotFoundException {

}

}

# C:\Users\Rikysamuel\Pictures\Kripto\1.pngPrint Screen Antarmuka

# Contoh Plain Text dan Cipher Text

## Contoh Kecil

Plain Text: Nama Saya Rikysamuel

Key: serigala

Cipher Text Vigenere Cipher Standard: FEDI YAJA JMBGYAXUWP

Cipher Text Vigenere Cipher Extended:   ¤ ¤¡ (Tidak dapat ditampilkan)

Cipher Text Autokey Vigenere Cipher Standard: FEDI YAJA EIWYKAKUVT

Cipher Text Autokey Vigenere Cipher Extended:   ¤ ¥¥ (Tidak dapat ditampilkan)

# Contoh Sedang

Plain Text:

Twitter Bootstrap is the most popular front end frameworks currently. It is sleek, intuitive, and powerful mobile first front-end framework for faster and easier web development. It uses HTML, CSS and Javascript. This tutorial will teach you basics of Bootstrap Framework using which you can create web projects with ease. Tutorial is divided into sections such as Bootstrap Basic Structure, Bootstrap CSS, Bootstrap Layout Components and Bootstrap Plugins. Each of these sections contain related topics with simple and useful examples.

Key: serigala

Cipher Text Vigenere Cipher Standard:

LAZBZEC BGSKAZRLP AW KPK MZSL TFXALLR XVFVZ EYD XVRUKWZRCW TCXRPNLPPP OT TS KPVMQF TNLYZBOVPF SRU XUWPRXYC UUBTLW JZZYT QRGRKOKNO FJEDMCOCK XSI NGSEEJ EEL KADIWV NMH DPVWPFXSEYTZ MK CYED HLQCN ISD AFH AIBADCJMGBN TSIK XLBURTAD AZTR TPAUL PWA BLSAGJ WL BZOLWKZGP QRSQVEURV UKMEO CHTCZ CFC IAY CJIRBK WPB HVFRKCES OMKP KADEZ XLBURTAD MJ LOVTDWH ZVZO DEUXZWTS DUUL RA HOZTKXIIV BLSAG JBXUNTMVVN HOZTKXIIV CDSX FFWZSERST CIEOFT USDXUNPNLW RVJ BZOLWKZGP ALMKZVYH PAUL FN ZHPSW WVKZIZNK GFVZATN JICIZEO TGTZKY WTTZ WZUVLP AFH LAKFFL WBRUVLPSZ

Cipher Text Vigenere Cipher Extended: (Tidak dapat ditampilkan)

¦ ¡¥  ¥ ¦       ¦ªv   ¥l ¦l     ¡¥u ¤ £ ¤   ¨   ¥  ¦t    ¤¡t ¥ ¦   ª ¤  ¦¥  ¥     ¤  ¢  ©¥  ¦ ¤      ¥£  ¤§t ¥£ ~     ¦  ¦¥ §n     ¥   ¤ ¡     ¦  

Cipher Text Autokey Vigenere Cipher Standard:

LAZBZEC BHKBLMVRQ WG MZX DOHB HHWYXOJ YGCCN PNU KIOZXABUPJ CGVNSEDDAB ZK MF LWCLSY QFLFMXSAMS THL IWRIWFHO BCXMCJ ZTDGU NCSSBXWGI WFNFKABUP WOD JWGKOW OEI ESLMVR JHF DWDICKTNHROL TH JEIF AAUEZ UWK HGP UFXSKCELYTC TZKJ BJMVKPID PCEZ KMAND GZF UESKJQ CZ COGBULFFQ TFTEXNOGP LSURC KYSWZ GBA YHV EYCONG WRD GVOCIYXT LZHQ ICLWD BNASRAES BM WWMQDPL AQBJ AHGWQBGG KYEA IG OGGNUARSQ POLAV JTGVCLCTWY SIQMMKVFQ QGLX UFOIULJFQ ZORGNK CDXPMBYGVG MCR OSBMKTEDQ DZNYBESW TLWN WS LOISG ZSHMPSFW USPMIWA JGZNMEL GFTTCL ALMV HQOHHM TUV CETQYL RAUETQYDL

Cipher Text Autokey Vigenere Cipher Extended: (Tidak dapat ditampilkan)

¦ ¢¥¦§ ¡ § ¢ ¢¢¤ ¡¨  £¤ ¡¨ ¥   ¦r  ¦ y ¢ £¬}¢  ¡ ~¡   q  ¡¥  ¦ ¤ ¥©r ¢ ¤   ¥ ¡ l¥ ¤ ¤§|¥ ª ¥ ¡  «  ¡¦   ¨¡ ¦ ¥ ¡  ¨  ¡ ¦¢ ¦ ¨y § § ¤ ¡¡ ¥ ¡ ¡¡¨ ¦ ¤¡¦ £§§¥l ¡¦~  £¦¤l ¬¡¨¥ §¡¡  §¥ ¨}   ¦u §  § ¤§ ¡ ¦ § ¢¢¢   r

## Contoh Besar

Plain Text:

Rails is a web application development framework written in the Ruby language. It is designed to make programming web applications easier by making assumptions about what every developer needs to get started. It allows you to write less code while accomplishing more than many other languages and frameworks. Experienced Rails developers also report that it makes web application development more fun. Rails is opinionated software. It makes the assumption that there is the "best" way to do things, and it's designed to encourage that way and in some cases to discourage alternatives. If you learn "The Rails Way" you'll probably discover a tremendous increase in productivity. If you persist in bringing old habits from other languages to your Rails development, and trying to use patterns you learned elsewhere, you may have a less happy experience. The Rails philosophy includes two major guiding principles: Don't Repeat Yourself: DRY is a principle of software development which states that "Every piece of knowledge must have a single, unambiguous, authoritative representation within a system." By not writing the same information over and over again, our code is more maintainable, more extensible, and less buggy. Convention Over Configuration: Rails has opinions about the best way to do many things in a web application, and defaults to this set of conventions, rather than require that you specify every minutiae through endless configuration files.