Lampiran

Program MD5Hash

```
class MD5Hash {
  private byte[] aBuf;
  private long Hitbyte;
  private static final int PANJ_DIGEST = 16;
  private int bBuf, aOff, A, B, C, D;
  private int[] X = new int[16];
  private static final char[] H = {
 '0','1','2','3','4','5','6','7','8','9',
 'a','b','c','d','e','f' };
public MD5Hash(){
  aBuf = new byte[4];
  bBuf = 0;
Awal();
}
public void Awal(){
 Hitbyte = 0;
 bBuf = 0;
 for(int i = 0; i < aBuf.length; i++ ){  aBuf[i]= 0; }
A = 0x67452301; B = 0xefcdab89;
C = 0x98badcfe;
D = 0x10325476;
 aOff = 0;
for (int i = 0; i != X.length; i++){ X[i] = 0; }
}
public String hitung(String m){
 Ambil(m.getBytes(), 0, m.getBytes().length);
 byte[] nilaiDigest = new byte[PANJ_DIGEST];
 Final(nilaiDigest, 0);
 return bytesKHex(nilaiDigest);
}
```

```
  awIn += aBuf.length; pan -= aBuf.length;
  Hitbyte += aBuf.length;
}
public void Ambil(byte[] in, int awIn, int pan){
 while ((bBuf != 0) && (pan > 0)){
  Ambil(in[awIn]);
while (pan > 0)
 Ambil(in[awIn]);
 awIn++;pan--;
}
  awIn++;pan--;
}
 while (pan > aBuf.length){
  prosesW(in, awIn);
}
public void Ambil(byte in){
 aBuf[bBuf++] = in;
 if (bBuf == aBuf.length){
 prosesW(aBuf, 0);
 bBuf = 0;
}
 Hitbyte++;
}
protected void prosesW(byte[] in, int awIn){
X[aOff++] = (in[awIn] & 0xff) | ((in[awIn + 1] & 0xff) << 8)
| ((in[awIn + 2] & 0xff) << 16)
| ((in[awIn + 3] & 0xff) << 24);
if (aOff == 16){
prosesBlock();
}
}
protected void prosesPanjang(long bitPan){
if (aOff > 14){ prosesBlock(); }
X[14] = (int)(bitPan & 0xffffffff);
X[15] = (int)(bitPan >>> 32);
}
private void Hash(int w, byte[] out, int kOff){
 out[kOff] = (byte)w;
 out[kOff + 1] = (byte)(w >>> 8);
 out[kOff + 2] = (byte)(w >>> 16);
 out[kOff + 3] = (byte)(w >>> 24);
}
```

```
public void Akhir (){
 long bitPan = (Hitbyte << 3);
 Ambil((byte)0x80);
 while (bBuf != 0){
 Ambil((byte)0);
}
 prosesPanjang(bitPan);prosesBlock();
}
public int Final(byte[] out, int kOff){
Akhir();
 Hash(A, out, kOff);
 Hash(B, out, kOff + 4);
 Hash(C, out, kOff + 8);
 Hash(D, out, kOff + 12);
Awal();
return PANJ_DIGEST;
}
public String bytesKHex(byte[] r) {
 int len = r.length;
 char[] hex = new char[len * 2];
 for (int i = 0; i < len; i++) {
 int nilai = (r[i] + 256) % 256;
 int hIndex = nilai >> 4;
 int lIndex = nilai & 0x0f;
 hex[i * 2 + 0] = H[hIndex];
 hex[i * 2 + 1] = H[lIndex];
}
 return (new String(hex)).toString();
}
//  1
private static final int S11 = 7;
private static final int S12 = 12;
private static final int S13 = 17;
private static final int S14 = 22;
//  2
private static final int S21 = 5;
private static final int S22 = 9;
private static final int S23 = 14;
private static final int S24 = 20;
//  3
private static final int S31 = 4;
private static final int S32 = 11;
```

```
private static final int S33 = 16;
private static final int S34 = 23;
//  4
private static final int S41 = 6;
private static final int S42 = 10;
private static final int S43 = 15;
private static final int S44 = 21;
// putar int x kiri n bits.
private int Kiri(int x,int n){
return (x << n) | (x >>> (32 - n)); }
// F, G, H dan I
private int F(int u, int v, int w){ return (u & v) | (~u &
w); }
private int G(int u, int v, int w){ return (u & w) | (v &
~w); }
private int H(int u, int v, int w){ return u ^ v ^ w; }
private int K(int u, int v, int w){ return v ^ (u | ~w); }
protected void prosesBlock(){
int a = A; int b = B;
int c = C; int d = D;
//  F
a = Kiri((a + F(b, c, d) + X[ 0] + 0xd76aa478), S11) + b;
d = Kiri((d + F(a, b, c) + X[ 1] + 0xe8c7b756), S12) + a;
c = Kiri((c + F(d, a, b) + X[ 2] + 0x242070db), S13) + d;
b = Kiri((b + F(c, d, a) + X[ 3] + 0xc1bdceee), S14) + c;
a = Kiri((a + F(b, c, d) + X[ 4] + 0xf57c0faf), S11) + b;
d = Kiri((d + F(a, b, c) + X[ 5] + 0x4787c62a), S12) + a;
c = Kiri((c + F(d, a, b) + X[ 6] + 0xa8304613), S13) + d;
b = Kiri((b + F(c, d, a) + X[ 7] + 0xfd469501), S14) + c;
a = Kiri((a + F(b, c, d) + X[ 8] + 0x698098d8), S11) + b;
d = Kiri((d + F(a, b, c) + X[ 9] + 0x8b44f7af), S12) + a;
c = Kiri((c + F(d, a, b) + X[10] + 0xffff5bb1), S13) + d;
b = Kiri((b + F(c, d, a) + X[11] + 0x895cd7be), S14) + c;
a = Kiri((a + F(b, c, d) + X[12] + 0x6b901122), S11) + b;
d = Kiri((d + F(a, b, c) + X[13] + 0xfd987193), S12) + a;
c = Kiri((c + F(d, a, b) + X[14] + 0xa679438e), S13) + d;
b = Kiri((b + F(c, d, a) + X[15] + 0x49b40821), S14) + c;
// 2 G
a = Kiri((a + G(b, c, d) + X[ 1] + 0xf61e2562), S21) + b;
d = Kiri((d + G(a, b, c) + X[ 6] + 0xc040b340), S22) + a;
c = Kiri((c + G(d, a, b) + X[11] + 0x265e5a51), S23) + d;
b = Kiri((b + G(c, d, a) + X[ 0] + 0xe9b6c7aa), S24) + c;
```

```
a = Kiri((a + G(b, c, d) + X[ 5] + 0xd62f105d), S21) + b;
d = Kiri((d + G(a, b, c) + X[10] + 0x02441453), S22) + a;
c = Kiri((c + G(d, a, b) + X[15] + 0xd8a1e681), S23) + d;
b = Kiri((b + G(c, d, a) + X[ 4] + 0xe7d3fbc8), S24) + c;
a = Kiri((a + G(b, c, d) + X[ 9] + 0x21e1cde6), S21) + b;
d = Kiri((d + G(a, b, c) + X[14] + 0xc33707d6), S22) + a;
c = Kiri((c + G(d, a, b) + X[ 3] + 0xf4d50d87), S23) + d;
b = Kiri((b + G(c, d, a) + X[ 8] + 0x455a14ed), S24) + c;
a = Kiri((a + G(b, c, d) + X[13] + 0xa9e3e905), S21) + b;
d = Kiri((d + G(a, b, c) + X[ 2] + 0xfcefa3f8), S22) + a;
c = Kiri((c + G(d, a, b) + X[ 7] + 0x676f02d9), S23) + d;
b = Kiri((b + G(c, d, a) + X[12] + 0x8d2a4c8a), S24) + c;
// 3  H
a = Kiri((a + H(b, c, d) + X[ 5] + 0xfffa3942), S31) + b;
d = Kiri((d + H(a, b, c) + X[ 8] + 0x8771f681), S32) + a;
c = Kiri((c + H(d, a, b) + X[11] + 0x6d9d6122), S33) + d;
b = Kiri((b + H(c, d, a) + X[14] + 0xfde5380c), S34) + c;
a = Kiri((a + H(b, c, d) + X[ 1] + 0xa4beea44), S31) + b;
d = Kiri((d + H(a, b, c) + X[ 4] + 0x4bdecfa9), S32) + a;
c = Kiri((c + H(d, a, b) + X[ 7] + 0xf6bb4b60), S33) + d;
b = Kiri((b + H(c, d, a) + X[10] + 0xbebfbc70), S34) + c;
a = Kiri((a + H(b, c, d) + X[13] + 0x289b7ec6), S31) + b;
d = Kiri((d + H(a, b, c) + X[ 0] + 0xeaa127fa), S32) + a;
c = Kiri((c + H(d, a, b) + X[ 3] + 0xd4ef3085), S33) + d;
b = Kiri((b + H(c, d, a) + X[ 6] + 0x04881d05), S34) + c;
a = Kiri((a + H(b, c, d) + X[ 9] + 0xd9d4d039), S31) + b;
d = Kiri((d + H(a, b, c) + X[12] + 0xe6db99e5), S32) + a;
c = Kiri((c + H(d, a, b) + X[15] + 0x1fa27cf8), S33) + d;
b = Kiri((b + H(c, d, a) + X[ 2] + 0xc4ac5665), S34) + c;
// K
a = Kiri((a + K(b, c, d) + X[ 0] + 0xf4292244), S41) + b;
d = Kiri((d + K(a, b, c) + X[ 7] + 0x432aff97), S42) + a;
c = Kiri((c + K(d, a, b) + X[14] + 0xab9423a7), S43) + d;
b = Kiri((b + K(c, d, a) + X[ 5] + 0xfc93a039), S44) + c;
a = Kiri((a + K(b, c, d) + X[12] + 0x655b59c3), S41) + b;
d = Kiri((d + K(a, b, c) + X[ 3] + 0x8f0ccc92), S42) + a;
c = Kiri((c + K(d, a, b) + X[10] + 0xffeff47d), S43) + d;
b = Kiri((b + K(c, d, a) + X[ 1] + 0x85845dd1), S44) + c;
a = Kiri((a + K(b, c, d) + X[ 8] + 0x6fa87e4f), S41) + b;
d = Kiri((d + K(a, b, c) + X[15] + 0xfe2ce6e0), S42) + a;
c = Kiri((c + K(d, a, b) + X[ 6] + 0xa3014314), S43) + d;
b = Kiri((b + K(c, d, a) + X[13] + 0x4e0811a1), S44) + c;
```

```
a = Kiri((a + K(b, c, d) + X[ 4] + 0xf7537e82), S41) + b;
d = Kiri((d + K(a, b, c) + X[11] + 0xbd3af235), S42) + a;
c = Kiri((c + K(d, a, b) + X[ 2] + 0x2ad7d2bb), S43) + d;
b = Kiri((b + K(c, d, a) + X[ 9] + 0xeb86d391), S44) + c;
A += a; B += b; C += c; D += d;
aOff = 0;
for (int i = 0; i != X.length; i++) { X[i] = 0; }
}
}
```

Program MD5Converter

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class MD5Converter implements ActionListener {
  JFrame converterFrame;
  JPanel converterPanel;
  JTextField  tempString;
 JTextField  atempString;
JLabel stringLabel,hashLabel,ahashLabel,bandingLabel,JC_MySalvoLabel;
  JButton convertTemp,bandingTemp,aconvertTemp;
  public MD5Converter() {
   //Create and set up the window.
    converterFrame = new JFrame("Message Digest 5");
converterFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);converterFra
me.setSize(new Dimension(100,100));
//Mebuat panel.
 converterPanel = new JPanel(new GridLayout(10,5));
 addWidgets();
converterFrame.getRootPane().setDefaultButton(convertTemp);
converterFrame.getRootPane().setDefaultButton(aconvertTemp);
converterFrame.getRootPane().setDefaultButton(bandingTemp);
converterFrame.getContentPane().add(converterPanel,BorderLayout.CENTER);
 converterFrame.pack();
 converterFrame.setVisible(true);
}
/**
 *Membuat dan menambah  widgets.
*/
private void addWidgets() {
```

```
//Membuat widgets.
 tempString = new JTextField();
 atempString = new JTextField();
 stringLabel = new JLabel("String",SwingConstants.LEFT);
 convertTemp = new JButton("Convert Input 1");
 aconvertTemp = new JButton("Convert Input 2");
 bandingTemp = new JButton("BANDING");
 hashLabel = new JLabel("Md5Hash 1",SwingConstants.LEFT);
 ahashLabel = new JLabel("Md5HASH 2",SwingConstants.LEFT);
 bandingLabel = new JLabel("",SwingConstants.LEFT);
JC_MySalvoLabel = new JLabel("JC_MySalvo",SwingConstants.CENTER);
//Aksi dari tombol convert.
 convertTemp.addActionListener(this);
 aconvertTemp.addActionListener(this);
 bandingTemp.addActionListener(this);
//Menambah widgets.
//Location
tempString.setLocation(0,0);
converterPanel.add(tempString);
converterPanel.add(atempString);
converterPanel.add(convertTemp);
converterPanel.add(aconvertTemp);
converterPanel.add(hashLabel);
converterPanel.add(ahashLabel);
converterPanel.add(bandingTemp);
converterPanel.add(bandingLabel);
converterPanel.add(JC_MySalvoLabel);
stringLabel.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
hashLabel.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
ahashLabel.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
bandingLabel.setBorder(BorderFactory.createEmptyBorder(5,5,5,5));
}
 public void actionPerformed(ActionEvent event) {
  //Memasukan class MD5Digest
    MD5Digest m = new MD5Digest();
    String s;String b;
     s=tempString.getText();b=atempString.getText();
     //Tombol convertTemp.
     if(event.getSource()== convertTemp){
     hashLabel.setText(m.calculate(s) + " MD5Hash");}
```

```
        //Tombol aconvertTemp.
     if(event.getSource()==aconvertTemp){
ahashLabel.setText(m.calculate(b) + " MD5Hash 2");}
     //Tombol Banding.
      if(event.getSource()==bandingTemp){sama();}

}
 public void sama(){
    String x;String y;
    int v,w;
 x=tempString.getText();y=atempString.getText();
 v=x.hashCode();w=y.hashCode();
 {if(v==w) bandingLabel.setText(" MD5 SAMA");
  else bandingLabel.setText("MD5 TIDAK SAMA");}
}
private static void createAndShowGUI() {
 JFrame.setDefaultLookAndFeelDecorated(true);
 MD5Converter converter = new MD5Converter();
}
public static void main (String[]args) {
javax.swing.SwingUtilities.invokeLater(new Runnable() {
 public void run() {
  createAndShowGUI();
   }
  });
 }
}
```

Memo Ron Rivest

RFC 1321 MD5 Message-Digest Algorithm April 1992

A.5 Test suite

The MD5 test suite (driver option "-x") should print the following results:

```
MD5 test suite:
MD5 ("") = d41d8cd98f00b204e9800998ecf8427e
MD5 ("a") = 0cc175b9c0f1b6a831c399e269772661
```

MD5 ("abc") = 900150983cd24fb0d6963f7d28e17f72
MD5 ("message digest") = f96b697d7cb7938d525a2f31aaf161d0
MD5 ("abcdefghijklmnopqrstuvwxyz") = c3fcd3d76192e4007dfb496cca67e13b
MD5
("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz01234567
89") =
d174ab98d277d9f5a5611c2c9f419d9f
MD5
("12345678901234567890123456789012345678901234567890123456789012345678
78901234567890") = 57edf4a22be3c955ac49da2e2107b67a

Security Considerations

The level of security discussed in this memo is considered to be sufficient for implementing very high security hybrid digital-signature schemes based on MD5 and a public-key cryptosystem.

Author's Address

Ronald L. Rivest
Massachusetts Institute of Technology
Laboratory for Computer Science
NE43-324
545 Technology Square
Cambridge, MA 02139-1986

Phone: (617) 253-5880
EMail: rivest@theory.lcs.mit.edu