

DAFTAR PUSTAKA

- Adhar, D. (2019). Implementasi Algoritma Des (Data Encryption Standard) Pada Enkripsi Dan Deskripsi Sms Berbasis Android. *JTIK (Jurnal Teknik Informatika Kaputama)*, 3(2), 53–60.
- Agustina, E. R., & Kurniati, A. (2015). Pemanfaatan kriptografi dalam mewujudkan keamanan informasi pada e-voting di indonesia. *Seminar Nasional Informatika (SEMNASIF)*, 1(3).
- Aleisa, N. (2015). A Comparison of the 3DES and AES Encryption Standards. *International Journal of Security and Its Applications*, 9(7), 241–246.
- Ariyus, D. (2008). *Pengantar ilmu kriptografi: teori analisis & implementasi*. Penerbit Andi.
- Gunawan, H. A., Arifin, Z., & Astuti, I. F. (2016). Keamanan Login Web Menggunakan Metode 3DES Berbasis Teknologi Quick Response Code. *Informatika Mulawarman: Jurnal Ilmiah Ilmu Komputer*, 9(2), 18–23.
- Hasugian, B. S. (2017). Peranan Kriptografi Sebagai Keamanan Sistem Informasi Pada Usaha Kecil Dan Menengah. *Warta Dharmawangsa*, 53.
- Nasution, Y. R., Furqan, M., & Sinaga, M. (2020). Implementasi Steganografi Menggunakan Metode Spread Spectrum Dalam Pengamanan Data Teks Pada Citra Digital. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 4(2), 351–358.
- Nugroho, A. (2005). *Analisis dan Perancangan Sistem Informasi dengan Metodologi Berorientasi Objek. Edisi Revisi*.
- Pratiwi, P., & WP, D. A. (2016). Peningkatan Keamanan Data Dengan Metode Cropping Selection Pseudorandom. *Jurnal TICOM*, 4(3), 92394.
- Primartha, R. (2011). Penerapan enkripsi dan dekripsi file menggunakan algoritma Data Encryption Standard (DES). *JSI: Jurnal Sistem Informasi (E-Journal)*, 3(2).

- Rohmanu, A. (2017). Implementasi kriptografi dan steganografi dengan metode algoritma DES dan metode End Of File. *Jurnal Informatika SIMANTIK*, 2(1), 1–11.
- Setyaningsih, E., Si, S., & Kom, M. (2015). Kriptografi & implementasinya menggunakan MATLAB. *Yogyakarta: ANDI*.
- Siregar, N. (2019). PERANCANGAN APLIKASI KEAMANAN PESAN TEKS DENGAN MENGGUNAKAN ALGORITMA TRIPLE DES. *JTIK (Jurnal Teknik Informatika Kaputama)*, 3(2), 11–17.
- Sulastris, S., & Putri, R. D. M. (2018). Implementasi Enkripsi Data Secure Hash Algorithm (SHA-256) dan Message Digest Algorithm (MD5) pada Proses Pengamanan Kata Sandi Sistem Penjadwalan Karyawan. *Jurnal Teknik Elektro*, 10(2), 70–74.
- Wibowo, G. T., Rumani, M., & Saputra, R. E. (2015). Analisis Dan Implementasi Enkripsi Dan Dekripsi Ganda Kombinasi Algoritma Blowfish Dan Algoritma Triple Des Untuk Sms Pada Smartphone Android. *EProceedings of Engineering*, 2(2).
- Winafil, M., Sinurat, S., & Zebua, T. (2018). Implementasi Algoritma Advanced Encryption Standard dan Triple Data Encryption Standard Untuk Mengamankan Citra Digital. *KOMIK (Konferensi Nasional Teknologi Informasi Dan Komputer)*, 2(1).
- Yanti, N. R., Alimah, A., & Ritonga, D. A. (2018). Implementasi Algoritma Data Encryption Standard Pada Penyandian Record Database. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 2(1), 23–32.
- Yusfrizal, Y. (2019). Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android. *JTIK (Jurnal Teknik Informatika Kaputama)*, 3(2), 29–37.s

LAMPIRAN I

```
<html>

<head>
  <title>Enkripsi DES-CBC(XOR)</title>
  <meta charset="utf-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
  <link rel="stylesheet" type="text/css" href="css/kedipjam.css">
  <link rel="stylesheet" type="text/css" href="css/kedipterminal.css">
  <link rel="stylesheet" type="text/css" href="css/bgandterminal.css">
  <link rel="stylesheet" type="text/css" href="css/menu.css">
  <style type="text/css">
  body {
    background: url(image/1.jpg);
    margin: 0;
    padding: 0;
    background-size: 100%
  }
  </style>
  <script src="js/terminal.js"></script>
</head>

<body>
  <!-- Dibawah Ini Tabel Menu -->
  <font color="#00fafa" font face="ubuntu" size="2">
    <nav>
      <a href="index.php">Cryptography</a>
      <a href="about.php">tentang</a>
```

```

    <div class="animation start-home"></div>
  </nav>
</font>
<!-- Finish -->

<!--text terminal-->
<div id="wrapper">
  <div class="box">
    <span class="prefix">
      <div id="console">
        <div id="message">
          <h1>
            <center>Tentang Aplikasi<span class="a">_</span></center>
          </h1>
        </div>
        <font size="5" color="#03fc6f">
          &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Aplikasi enkripsi ini menggunakan
        algoritma DES dimana proses Enkripsi
        menggunakan file PDF sebagai file plaintext-nya dan menggunakan
        kunci.<br>
        <br>
        </font>
        <h1><center>Tentang Saya<span
class="a">_</span></center></h1>
          <font size="5" color="#03fc6f">
            Nama : Arif Wijaya Panjaitan<br>
            Jurusan : Ilmu Komputer<br>
            Nim : 0701163093<br>
          </div>
        </span>
      </div>
    </div>
  </div>

```



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

```
</div>
</body>

</html>
<?php
```

```
class DataEncryptionStandard {
    private $show_log = true;
    private $log = array();
    private $k = array();
    private $PC1 =
        array(57,49,41,33,25,17, 9,
            1, 58, 50, 42, 34, 26, 18,
            10, 2, 59, 51, 43, 35, 27,
            19, 11, 3, 60, 52, 44, 36,
            63, 55, 47, 39, 31, 23, 15,
            7, 62, 54, 46, 38, 30, 22,
            14, 6, 61, 53, 45, 37, 29,
            21, 13, 5, 28, 20, 12, 4);

    private $PC2 = array(14, 17, 11, 24, 1, 5,
        3, 28, 15, 6, 21, 10,
        23, 19, 12, 4, 26, 8,
        16, 7, 27, 20, 13, 2,
        41, 52, 31, 37, 47, 55,
        30, 40, 51, 45, 33, 48,
        44, 49, 39, 56, 34, 53,
        46, 42, 50, 36, 29, 32);
```

```
private $IP = array(58, 50, 42, 34, 26, 18, 10, 2,
60, 52, 44, 36, 28, 20, 12, 4,
62, 54, 46, 38, 30, 22, 14, 6,
64, 56, 48, 40, 32, 24, 16, 8,
57, 49, 41, 33, 25, 17, 9, 1,
59, 51, 43, 35, 27, 19, 11, 3,
61, 53, 45, 37, 29, 21, 13, 5,
63, 55, 47, 39, 31, 23, 15, 7);
```

```
private $IP1 = array(40, 8, 48, 16, 56, 24, 64, 32,
39,7,47,15,55, 23,63,31,
38, 6,46,14, 54, 22, 62, 30,
37, 5, 45, 13, 53, 21, 61,29,
36, 4, 44,12,52, 20, 60, 28,
35,3, 43,11,51, 19, 59, 27,
34, 2, 42, 10, 50, 18, 58, 26,
33, 1, 41, 9, 49, 17, 57,25);
```

```
private $E = array(32, 1, 2, 3, 4, 5,
4, 5, 6, 7, 8, 9,
8, 9, 10, 11, 12, 13,
12, 13, 14, 15, 16, 17,
16, 17, 18, 19, 20, 21,
20, 21, 22, 23, 24, 25,
24, 25, 26, 27, 28, 29,
28, 29, 30, 31, 32, 1);
```

```
private $P = array(16, 7, 20, 21,
29, 12, 28, 17,
1, 15, 23, 26,
5, 18, 31, 10,
```

```

2,      8,      24,     14,
32,    27,     3,      9,
19,    13,     30,     6,
22,    11,     4,      25);

```

```

Private $r = array( 1 => 1,

```

```

2 => 1,

```

```

3 => 2,

```

```

4 => 2,

```

```

5 => 2,

```

```

6 => 2,

```

```

7 => 2,

```

```

8 => 2,

```

```

9 => 1,

```

```

10 => 2,

```

```

11 => 2,

```

```

12 => 2,

```

```

13 => 2,

```

```

14 => 2,

```

```

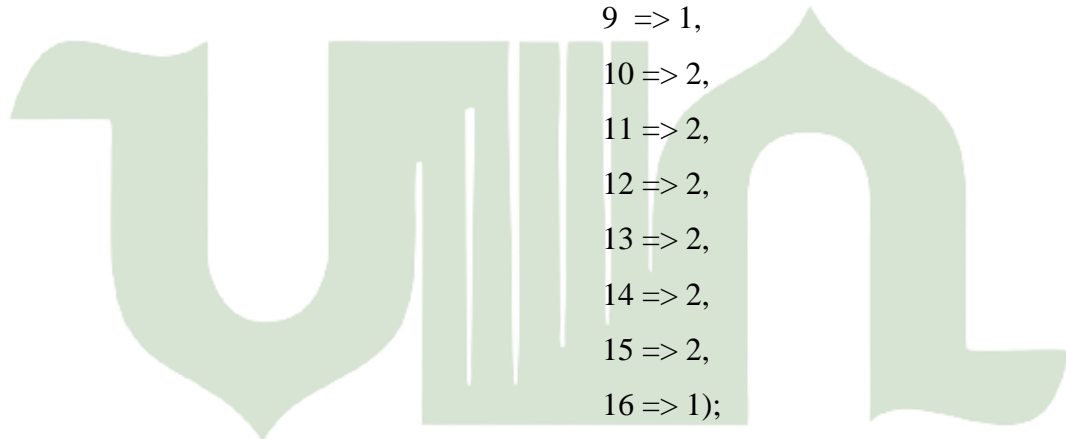
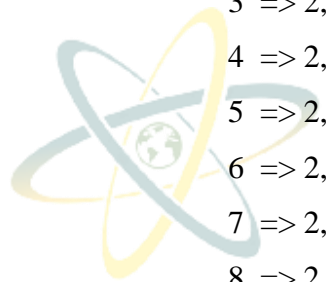
15 => 2,

```

```

16 => 1);

```



```

private function leftShift($keyPC1) {
return substr($keyPC1, 1, (strlen($keyPC1) - 1)) . substr($keyPC1,
0, 1);
}

```

```

private function permutation($array, $data) {
    $rs = "";
    foreach($array as $index) {
        $rs .= $data[$index-1];
    }
}

```

```

    }
    return $rs;
}

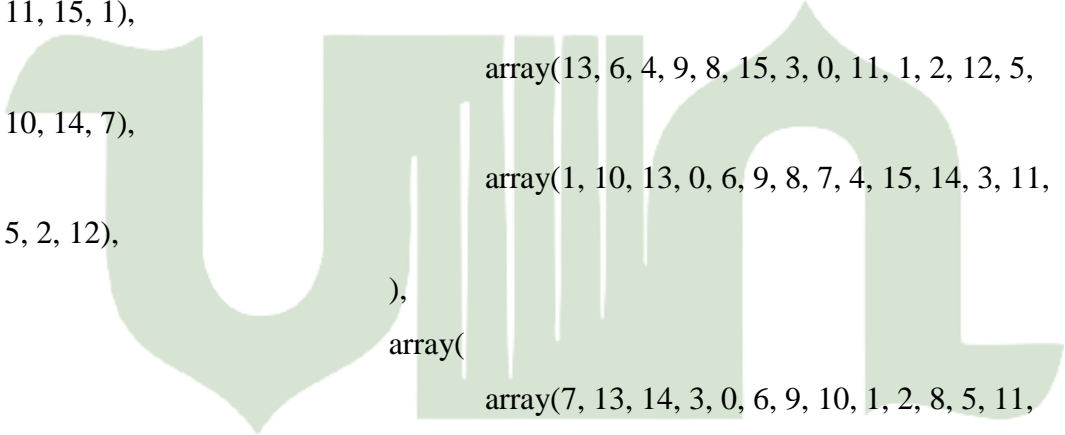
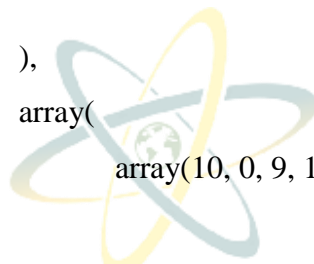
private function dataXOR($data1, $data2) {
    if(strlen($data1) != strlen($data2)) exit('xor err');

    $rs = "";
    for($i = 0; $i < strlen($data1); $i++) {
        if(($data1[$i] == '1' or $data2[$i] == '1') and ($data1[$i] ==
'0' or $data2[$i] == '0')) $logic = '1';
        else $logic = '0';
        $rs .= $logic;
    }
    return $rs;
}

private function sbox($EL) {
    $S = array(
        array(
            array(14, 4, 13, 1, 2, 15, 11, 8, 3, 10, 6, 12,
5, 9, 0, 7),
            array(0, 15, 7, 4, 14, 2, 13, 1, 10, 6, 12, 11,
9, 5, 3, 8),
            array(4, 1, 14, 8, 13, 6, 2, 11, 15, 12, 9, 7, 3,
10, 5, 0),
            array(15, 12, 8, 2, 4, 9, 1, 7, 5, 11, 3, 14, 10,
0, 6, 13),
        ),
        array(

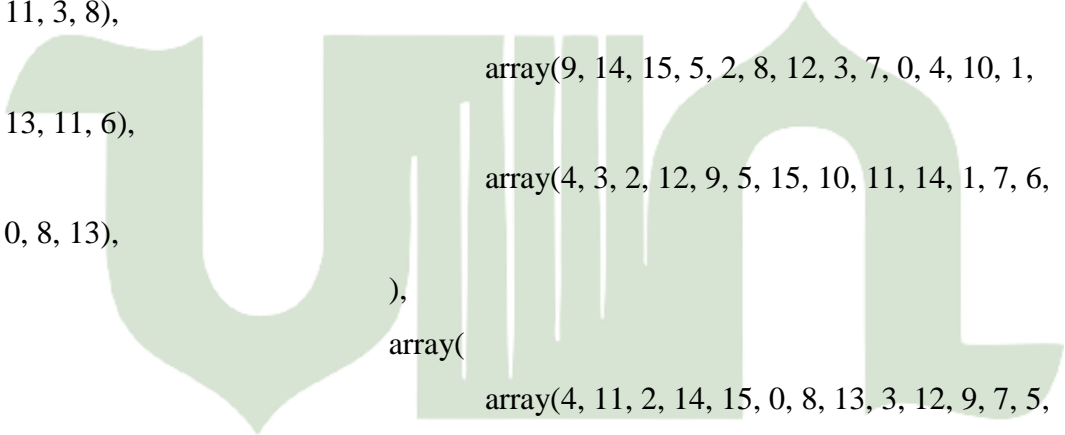
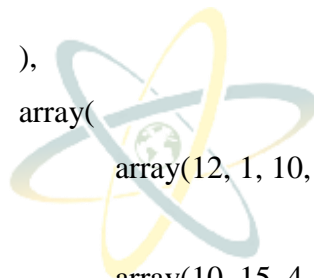
```


array(15, 1, 8, 14, 6, 11, 3, 4, 9, 7, 2, 13, 12,
0, 5, 10),
array(3, 13, 4, 7, 15, 2, 8, 14, 12, 0, 1, 10, 6,
9, 11, 5),
array(0, 14, 7, 11, 10, 4, 13, 1, 5, 8, 12, 6, 9,
3, 2, 15),
array(13, 8, 10, 1, 3, 15, 4, 2, 11, 6, 7, 12, 0,
5, 14, 9),
)
array(
array(10, 0, 9, 14, 6, 3, 15, 5, 1, 13, 12, 7,
11, 4, 2, 8),
array(13, 7, 0, 9, 3, 4, 6, 10, 2, 8, 5, 14, 12,
11, 15, 1),
array(13, 6, 4, 9, 8, 15, 3, 0, 11, 1, 2, 12, 5,
10, 14, 7),
array(1, 10, 13, 0, 6, 9, 8, 7, 4, 15, 14, 3, 11,
5, 2, 12),
)
array(
array(7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11,
12, 4, 15),
array(13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1,
10, 14, 9),
array(10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14,
5, 2, 8, 4),
array(3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12,
7, 2, 14),
)
array(



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

array(2, 12, 4, 1, 7, 10, 11, 6, 8, 5, 3, 15, 13,
0, 14, 9),
array(14, 11, 2, 12, 4, 7, 13, 1, 5, 0, 15, 10,
3, 9, 8, 6),
array(4, 2, 1, 11, 10, 13, 7, 8, 15, 9, 12, 5, 6,
3, 0, 14),
array(11, 8, 12, 7, 1, 14, 2, 13, 6, 15, 0, 9,
10, 4, 5, 3),
)
array(
array(12, 1, 10, 15, 9, 2, 6, 8, 0, 13, 3, 4, 14,
7, 5, 11),
array(10, 15, 4, 2, 7, 12, 9, 5, 6, 1, 13, 14, 0,
11, 3, 8),
array(9, 14, 15, 5, 2, 8, 12, 3, 7, 0, 4, 10, 1,
13, 11, 6),
array(4, 3, 2, 12, 9, 5, 15, 10, 11, 14, 1, 7, 6,
0, 8, 13),
)
array(
array(4, 11, 2, 14, 15, 0, 8, 13, 3, 12, 9, 7, 5,
10, 6, 1),
array(13, 0, 11, 7, 4, 9, 1, 10, 14, 3, 5, 12, 2,
15, 8, 6),
array(1, 4, 11, 13, 12, 3, 7, 14, 10, 15, 6, 8,
0, 5, 9, 2),
array(6, 11, 13, 8, 1, 4, 10, 7, 9, 5, 0, 15, 14,
2, 3, 12),
)
array(
array(



UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

```

array(13, 2, 8, 4, 6, 15, 11, 1, 10, 9, 3, 14, 5,
0, 12, 7),
array(1, 15, 13, 8, 10, 3, 7, 4, 12, 5, 6, 11, 0,
14, 9, 2),
array(7, 11, 4, 1, 9, 12, 14, 2, 0, 6, 10, 13,
15, 3, 5, 8),
array(2, 1, 14, 7, 4, 10, 8, 13, 15, 12, 9, 0, 3,
5, 6, 11),

```



```

),
);

$BI = array();
$it = 0;
for($i = 0; $i < 8; $i++){
    $row = substr($EL, $it, 6);

    $a = base_convert($row[0].$row[5], 2, 10);
    $b = base_convert(substr($row, 1, 4), 2, 10);

    $BICurrent = base_convert($S[$i][$a][$b], 10, 2);

    $BI[$i] = (strlen($BICurrent) < 4)? str_pad($BICurrent, 4,
"0", STR_PAD_LEFT) : $BICurrent;
    $it += 6;
}

return implode(", $BI);
}

private function stringToHexToBin($key) {
    $binKey = "";

```

```

for($i = 0; $i < strlen($key); $i++) {
    $num = ord($key[$i]);
    $bin = base_convert($num, 10, 2);
    if(strlen($bin) < 8) $bin = str_pad($bin, 8, "0",
STR_PAD_LEFT);

```

```

    $binKey .= $bin;
}
return $binKey;
}

```

```

private function stringToHex ($s) {
    $r = "";
    $hexes = array
("0","1","2","3","4","5","6","7","8","9","A","B","C","D","E","F");
    for ($i = 0; $i < strlen($s); $i++) {
        $r .= ($hexes [(ord($s[$i]) >> 4)] . $hexes [(ord($s[$i]) &
0xf)]);
    }
    return $r;
}

```

```

private function hexToBin($hex){

```

```

    $bin="";
    for ($i = 0; $i < strlen($hex); $i++){
        switch($hex[$i]){
            case "0" : $bin .= "0000"; break;
            case "1" : $bin .= "0001"; break;
            case "2" : $bin .= "0010"; break;
            case "3" : $bin .= "0011"; break;
            case "4" : $bin .= "0100"; break;

```

```

case "5" : $bin .= "0101"; break;
case "6" : $bin .= "0110"; break;
case "7" : $bin .= "0111"; break;
case "8" : $bin .= "1000"; break;
case "9" : $bin .= "1001"; break;
case "A" : $bin .= "1010"; break;
case "B" : $bin .= "1011"; break;
case "C" : $bin .= "1100"; break;
case "D" : $bin .= "1101"; break;
case "E" : $bin .= "1110"; break;
case "F" : $bin .= "1111"; break;
}
}

```

```

return $bin;
}

```

```

private function generateKey($key){

```

```

// Step 1 : Create 16 Subkeys

```

```

$this->log['key']['key'] = $key;

```

```

$this->log['key']['hexKey'] = $this->stringToHex($key); // string

```

key to hex

```

$binKey = $this->log['key']['binKey'] = $this-

```

```

->stringToHexToBin($key); // string key to hex then to binary

```

```

$keyPC1 = "";

```

```

foreach($this->PC1 as $index) {

```

```

// given K is permuted according to PC-1

```

```

$this->log['key']['keyPermutation (K+)'] = $keyPC1 .=

```

```

$binKey[$index-1];

```

```

}

```

```

// Step 2 : K+ splitting
$C[0] = substr($keyPC1, 0, 28);
$D[0] = substr($keyPC1, 28, 28);

// step 3 : Creating 16 subkeys using shifting
for($i = 1; $i <= 16; $i++) {
    $shiftC = $C[$i-1];
    $shiftD = $D[$i-1];

    for($shiftIndex = 1; $shiftIndex <= $this->r[$i];
$shiftIndex++) {
        $shiftC = $this->leftShift($shiftC); // left shift Cn
        $shiftD = $this->leftShift($shiftD); // left shift Dn
    }
    $C[$i] = $shiftC;
    $D[$i] = $shiftD;
}

// step 4 : PC-2 Permutation
for($i = 1; $i <= 16; $i++) {
    $concat = $C[$i] . $D[$i];

    $this->k[$i] = ""; // Kn (permuted according to the PC-2)
    foreach($this->PC2 as $index) {
        $this->k[$i] .= $concat[$index-1];
    }
}

foreach($D as $id => $row) {
    $this->log['key']['C-' . $id] = $C[$id];
    $this->log['key']['D-' . $id] = $D[$id];
}

```

```

        if(isset($this->k[$id])) $this->log['key']['K-' . $id] = $this-
>k[$id];
    }
}

private function generateKeyChiper($key){
    // Step 1 : Create 16 Subkeys
    $this->log['key']['key'] = $key;
    $this->log['key']['hexKey'] = $this->stringToHex($key); // string
key to hex

    $binKey = $this->log['key']['binKey'] = $this-
>stringToHexToBin($key); // string key to hex then to binary

    $keyPC1 = "";
    foreach($this->PC1 as $index) {
        // given K is permuted according to PC-1
        $this->log['key']['keyPermutation (K+)'] = $keyPC1 .=
$binKey[$index-1];
    }

    // Step 2 : K+ splitting
    $C[0] = substr($keyPC1, 0, 28);
    $D[0] = substr($keyPC1, 28, 28);

    // step 3 : Creating 16 subkeys using shifting
    for($i = 1; $i <= 16; $i++) {
        $shiftC = $C[$i-1];
        $shiftD = $D[$i-1];

        for($shiftIndex = 1; $shiftIndex <= $this->r[$i];
$shiftIndex++) {
            $shiftC = $this->leftShift($shiftC); // left shift Cn

```

```

        $shiftD = $this->leftShift($shiftD); // left shift Dn
    }
    $C[$i] = $shiftC;
    $D[$i] = $shiftD;
    $CC[$i] = $C[$i];
    $DD[$i] = $D[$i];
}
for ($j = 16; $j >= 1; $j--){
    $k++;
    $C[$j] = $CC[$k];
    $D[$j] = $DD[$k];
}
// step 4 : PC-2 Permutation
for($i = 1; $i <= 16; $i++) {
    $concat = $C[$i] . $D[$i];

    $this->k[$i] = ""; // Kn (permuted according to the PC-2)
    foreach($this->PC2 as $index) {
        $this->k[$i] .= $concat[$index-1];
    }
}

foreach($D as $id => $row) {
    $this->log['key']['C-' . $id] = $C[$id];
    $this->log['key']['D-' . $id] = $D[$id];
    if(isset($this->k[$id])) $this->log['key']['K-' . $id] = $this->k[$id];
}
}
}
private function encryptMessage($message){
    // Step 1: IP permutation

```



```

$this->log['msg']['message'] = $message;
$this->log['msg']['hexMsg'] = $this->stringToHex($message);
$binMsg = $this->log['msg']['binMsg'] = $this-
>stringToHexToBin($message);

$sourceIP = "";
foreach($this->IP as $IPIndex) {
    $this->log['msg']['msgIP'] = $sourceIP .=
$binMsg[$IPIndex - 1];
}

// Step 2 : IP splitting
$L = $R = array();
$L[0] = $this->log['msg']['L0'] = substr($sourceIP, 0, 32);
$R[0] = $this->log['msg']['R0'] = substr($sourceIP, 32, 32);

// Step 3 : Iterations
for($i = 1; $i <= 16; $i++) {
    // set Ln
    $L[$i] = $this->log['msg']['L' . $i] = $R[$i-1];

    // Step 3.1 : E permutation
    $SEL = $this->log['msg']['E(R' . ($i - 1) . ')'] = $this-
>permutation($this->E, $L[$i]);

    // Step 3.2 : XOR with a subkey
    $this->log['msg']['K' . $i] = $this->k[$i];
    $this->log['msg']['K' . $i . ' xor E(R' . ($i - 1) . ')'] = $SEL =
$this->dataXOR($SEL, $this->k[$i]);

    // Step 3.3 : S box transformation

```

```

    $this-
>log['msg']['S(B1)S(B2)S(B3)S(B4)S(B5)S(B6)S(B7)S(B8) ' . $i] = $s = $this-
>sbx($EL);

    // Step 3.4 : P permutation
    $this->log['msg']['f =
P(S(B1)S(B2)S(B3)S(B4)S(B5)S(B6)S(B7)S(B8)) ' . $i] = $f = $this-
>permutation($this->P, $s);

    // set Rn
    $this->log['msg']['R' . $i] = $R[$i] = $this-
>dataXOR($L[$i-1], $f);
    }

    // Step 4 : Reverse Connecting
    $this->log['msg']['L16 concat R16'] = $concat = $R[16] . $L[16];

    // Step 5 : IP-1 permutation
    $this->log['msg']['IP-1 permutation'] = $encoded = $this-
>permutation($this->IP1, $concat);
    $this->log['msg']['result'] = $result = "";
    for($start = 0; $start < strlen($encoded); $start += 4) {
        $this->log['msg']['result'] = $result .=
strtoupper(base_convert(substr($encoded, $start, 4), 2, 16));
    }

    // result with convert into hex
    return $result;
}

public function encrypt($message, $key){

```

```

        if(strlen($message) == 8 && strlen($key) == 8) {
            $this->generateKey($key);
            return $this->encryptMessage($message);
        } else {
            echo 'Message & Key lenght must be 8 characters.';
        }
    }
}

public function decrypt($message, $key){
    if(strlen($message) == 8 && strlen($key) == 8) {
        $this->generateKeyChiper($key);
        return $this->encryptMessage($message);
    } else {
        echo 'Message & Key lenght must be 8 characters.';
    }
}

public function showLog(){
    echo '<pre>';
    print_r($this->log, false);
    echo '</pre>';
}
}

<html>

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

<head>
    <title>Enkripsi DES-CBC(XOR)</title>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">

    <link rel="stylesheet" type="text/css" href="css/bootstrap.min.css">
    <link rel="stylesheet" type="text/css" href="css/kedipjam.css">

```

```

<link rel="stylesheet" type="text/css" href="css/kedipterminal.css">
<link rel="stylesheet" type="text/css" href="css/bgandterminal.css">
<link rel="stylesheet" type="text/css" href="css/menu.css">
<style type="text/css">
body {
    background: url(image/1.jpg);
    margin: 0;
    padding: 0;
    background-size: 100%
}
</style>
<script src="js/terminal.js"></script>
</head>

```



```

<body>
<!-- Dibawah Ini Tabel Menu -->
<font color="#00fafa" font face="ubuntu" size="2">
<nav>
    <a href="index.php">Cryptography</a>
    <a href="about.php">About</a>
    <div class="animation start-home"></div>
</nav>
</font>
<!-- Finish -->

```

```

<!--text terminal-->
<div id="wrapper">
    <div class="box">
        <span class="prefix">
            <div id="console">
                <div id="message">

```

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

```

<h1>
  <center>Cryptography PDF(DES)<span
class="a">_</span></center>
  </h1>
</div>
Mode CBC(Cipher Block Chaining)<br>
Rumus  $(Y_{i-1} = PlainText_{i-1} XOR$ 
Ciphertext $-1)$ <br>
<br>
<form method="POST" action="" enctype="multipart/form-data">
  <font color="#00ff9d">
    <div class="form-group row">
      <div class="col-xs-2">
        Pilih File Pdf
        <input type="file" class="form-control-file" name="pdf"
accept="application/pdf"></br>
        Kunci :<input type="text" name="kunci" maxlength="8"
class="form-control"
placeholder="12345678">
      </div>
    </div>
    <input type="submit" name="enkripsi" class="btn btn-danger"
value="Encrypt">
    <input type="submit" name="dekripsi" class="btn btn-success"
value="Decrypt">
  </form><?php
error_reporting(0);

```

```

function enkripsiDES($data, $key) {
    $initialPermutasi = $key;
    //Enkripsi dengan Mode DES CBC/XOR 64bit

    //Encode dengan base64 karena pdf bukan plaintext melainkan data object
    //agar saat perubahan/enkripsi data tidak rusak
    return base64_encode($enkripsi);
}

function dekripsiDES($data, $key) {
    $initialPermutasi = $key;
    $raw = base64_decode($data);
    $dekripsi = openssl_decrypt($raw, 'DES-CBC', $key,
    OPENSSL_RAW_DATA, $initialPermutasi);
    return $dekripsi;
}

if(isset($_POST['enkripsi'])){
    $kunci = $_POST['kunci'];
    $pdf = $_FILES['pdf']['name']; //Mengambil nama dari sebuah file pdf
    $pdfbaru = "enkripsi".date('is'); //Rename file pdf
    $tmp = $_FILES['pdf']['tmp_name'];
    $ekstensi = explode(".", $pdf);
    $ekstensi1 = ".$ekstensi[1];

    mcrypt_ecb(cipher, key, data, mode) if($ekstensi1 == ".pdf"){

        $pdfobj = file_get_contents($tmp);
        $msg_encrypted = enkripsiDES($pdfobj, $kunci);

        $file = fopen("./file/$pdfbaru", 'wb');
        fwrite($file, $msg_encrypted);
    }
}

```

```

fclose($file);

echo "<br><br>". "Kunci = ".$kunci."<br>";
echo "Initial Permutation". "<br>";
$k = strlen($kunci);
$hasil = "";
$hasil2 = "";
$hasil3 = "";
while ($k-- > 0) {
    $hasil = str_pad(dechex(ord($kunci[$k])), 2, "0", STR_PAD_LEFT) . ' ';
$hasil;
    $hasil2 = str_pad(decbin(ord($kunci[$k])), 8, "0", STR_PAD_LEFT) . ' ';
$hasil2;
}
echo "Hex = ".$hasil."<br>";
echo "Binary = ".$hasil2."<br>";
echo "<a href='\$pdfbaru'>Download </a>";
}
else{
    echo '<script language="javascript">';
    echo 'alert("Harus File PDF")';
    echo '</script>';
}
}

if(isset($_POST['dekripsi'])){
    $kunci = $_POST['kunci'];
    $pdf = $_FILES['pdf']['name']; //Mengambil nama dari sebuah file pdf
    echo $pdfbaru = "dekripsi".date('is')."pdf"; //Rename file pdf
    $tmp = $_FILES['pdf']['tmp_name'];

```

```

$pdfobj = file_get_contents($tmp);
$msg_encrypted = dekripsiDES($pdfobj, $kunci);

$file = fopen("./file/$pdfbaru", 'wb');
fwrite($file, $msg_encrypted);
fclose($file);

echo "<br>". "Kunci = ".$kunci."<br>";
echo "<br>". "Initial Permutation". "<br>";
$k = strlen($kunci);
$hasil = "";
$hasil2 = "";
$hasil3 = "";
while ($k-->0) {
    $hasil = str_pad(dehex(ord($kunci[$k])), 2, "0", STR_PAD_LEFT) . ' ';
$hasil;
    $hasil2 = str_pad(decbin(ord($kunci[$k])), 8, "0", STR_PAD_LEFT) . ' ';
$hasil2;
}
echo "Hex = ".$hasil."<br>";
echo "Binary = ".$hasil2."<br>";
echo "<a href='$pdfbaru'>Download </a>";
}
?>
</div>
</span>
</div>
</div>
</body>

</html>

```


LAMPIRAN 2

DAFTAR RIWAYAT HIDUP

1. DATA PRIBADI

Nama : Arif Wijaya Panjaitan
Nim : 0701163093
Tempat. Tanggallahir : Lubuk Pakam, 25 September 1998
Alamat : jalan bakti II Gg rahayu 47 Kecamatan
Lubuk Pakam Kabupaten Deli Serdang
Sumatera Utara
Alamat Email : arifpanjaitan70@gmail.com
No-Hp : 0895600135703
Agama : Islam
Status : Single/ Mahasiswa
Tinggi/ berat badan : 163 cm/90 kg
Kemampuan : Komputerisasi (Microsoft Office)



2. DATA PENDIDIKAN

PerguruanTinggi : Universitas IslamNegeri Sumatera Utara
Medan (2016-2022)

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN
Medan, 24 Maret 2022

Arif Wijaya Panjaitan






LAMPIRAN 3

KARTU BIMBINGAN SKRIPSI

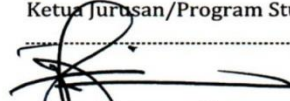
Semester Gasal/Genap Tahun Akademik /

Nama : ARIF Wipya Pangkajene	Pembimbing I : HZ WKA ILka Zulfria, M.Kom
NIM : 070162003	Pembimbing II : Yusuf Ramadhan Nasution, M.Kom
Prog. Studi : Ilmu Komputer	SK Pembimbing :
Judul Skripsi : Implementasi algoritma DES untuk pengamanan data pada Dokument PDF	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	30 Juni 2021	Konsultasi bab I-III		31 Maret 2021	Konsultasi bab I	
II	30 Juni 2021	Revisi bab I-III		31 Maret 2021	Revisi bab I	
III	8 Juli 2021	Revisi bab I-III		16 April 2021	Revisi bab I	
IV	10 Juli 2021	Acc Seminar proposal		4 Juni 2021	Konsultasi bab I & II Revisi bab III	Paring
V				26 Juni 2021	Acc Sempro	PARING

VI	2/3	Revisi bab IV	7	4/3	Perbaikan Bab IV	
VII		Sembatkan Bab IV x Program		2/3	Acc. Selay	
VIII		perbaikan Bab IV x Program				
IX		Acc Selay Menanggapi				
X						

Medan, 21 April 2022
 An. Dekan
 Ketua Jurusan/Program Studi


 Wita Zulfira, M.kom
 NIP. 19506092015310006

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing