

## LAMPIRAN

### Lampiran 1. Perhitungan Patchwork

<b>Jumlah Byte Gambar</b>				
X	Y	Kedalaman	Bit	
360	640	3	24	
Resolusi	230400	pixel		
jumlah byte	691200		202554	
jumlah bit	16588800			

<b>Password</b>				
A	B	C		
65	66	67		
01000001	01000010	01000011		

<b>Pesan</b>				
Text			Panjang Biner text	
P	80	01010000	8	64
O	79	01001111		10
L	76	01001100		2
I	73	01001001		
N	78	01001110		
E	69	01000101		
M	77	01001101		
A	65	01000001		

<b>Pixel</b>		
No.	Alamat	Byte

1	4357	4357
2	287629	287629
3	18983581	321181
4	21198013	462013
5	30492925	80125
6	5288317	449917
7	29694589	664189
8	43836541	290941
9	19202173	539773
10	35625085	373885
11	24676477	484477
12	31975549	180349
13	11903101	152701
14	10078333	401533
15	26501245	235645
16	15552637	346237
17	22851709	42109
18	2779261	14461
19	954493	263293
20	17377405	97405
21	6428797	207997
22	13727869	595069
23	39274621	567421
24	37449853	125053
25	8253565	650365
26	42924157	69757
27	4604029	456829
28	30150781	429181
29	28326013	678013
30	44748925	512125
31	33800317	622717

32	41099389	318589
----	----------	--------

## Lampiran 2. Script Sistem

<b>Generate Alamat Password</b>	
<pre> class Address {     public static int[] generateAlamat(int size, string pass)     {         int[] alamat = new int[] { };         char[] passArr = pass.ToCharArray();         int firstKey = (int)passArr[0];         int secondtKey = (int)passArr[1];         int thirdKey = (int)passArr[2];         int x0 = firstKey;         while (true)         {             int lamatnya = (x0 * secondtKey + thirdKey) % size;              x0 = lamatnya;             if (alamat.Length == 0)             {                 var temp = alamat.ToList();                 temp.Add(lamatnya);                 alamat = temp.ToArray();             }             else             {                 if (alamat.Contains(lamatnya))                 {                     break;                 }                 else                 {                     var temp = alamat.ToList();                     temp.Add(lamatnya);                     alamat = temp.ToArray();                 }             }         }         Array.Sort(alamat);         return alamat;     } } </pre>	
<b>Convert Kata ke Binary (String to Bin)</b>	
<pre> public static string toBin(string data, bool formatBits = false) { </pre>	

```

        char[] buffer = new char[(((data.Length * 8) +
(formatBits ? (data.Length - 1) : 0)))]];
        int index = 0;
        for (int i = 0; i < data.Length; i++)
        {
            string binary = Convert.ToString(data[i],
2).PadLeft(8, '0');
            for (int j = 0; j < 8; j++)
            {
                buffer[index] = binary[j];
                index++;
            }
            if (formatBits && i < (data.Length - 1))
            {
                buffer[index] = ' ';
                index++;
            }
        }
        return new string(buffer);
    }

```

### Extract Text

```

public static string extractText(string pass, Bitmap bmp)
    {
        string extractedText = String.Empty;

        int channel = Image.GetPixelFormatSize(bmp.PixelFormat) / 8;
        int size = bmp.Width * bmp.Height * channel;
        int[] alamat = Address.generateAlamat(size, pass);

        int indexAlamat = 0;

        int R = 0, G = 0, B = 0, S = 0;

        System.Diagnostics.Debug.WriteLine("Panjang Gambar "
+ size);
        System.Diagnostics.Debug.WriteLine("Panjang Alamat "
+ alamat.Length);
        CreateLog("Proses Extraksi");
        CreateLog("Panjang Gambar " + size);
        CreateLog("Panjang Alamat " + alamat.Length);
        for (int i = 0; i < bmp.Height; i++)
        {
            for (int j = 0; j < bmp.Width; j++)
            {
                Color pixel = bmp.GetPixel(j, i);

                R = pixel.R;
                G = pixel.G;
                B = pixel.B;

                for (int n = 0; n < 3; n++)
                {

```

```

        if (S == alamat[indexLamat])
        {
            switch (n)
            {
                case 0:
                {
                    string biner =
Convert.ToString(R, 2).PadLeft(8, '0');
                    extractedText +=
biner[7];

System.Diagnostics.Debug.Write("Alamat Ke " + indexLamat + ",
Array ke" + alamat[indexLamat] + " extra " +
extractedText[indexLamat]);

System.Diagnostics.Debug.WriteLine(" di N " + n + " Color " + R
+ "," + G + "," + B);

                    CreateLog("Ke " +
indexLamat + "," + alamat[indexLamat] + " extract " +
extractedText[indexLamat] + " di N " + n + " Color " + R + "," +
G + "," + B);
                }
                break;
                case 1:
                {
                    string biner =
Convert.ToString(G, 2).PadLeft(8, '0');
                    extractedText +=
biner[7];

System.Diagnostics.Debug.Write("Alamat Ke " + indexLamat + ",
Array ke" + alamat[indexLamat] + " extra " +
extractedText[indexLamat]);

System.Diagnostics.Debug.WriteLine(" di N " + n + " Color " + R
+ "," + G + "," + B);

                    CreateLog("Ke " +
indexLamat + "," + alamat[indexLamat] + " extract " +
extractedText[indexLamat] + " di N " + n + " Color " + R + "," +
G + "," + B);
                }
                break;
                case 2:
                {
                    string biner =
Convert.ToString(B, 2).PadLeft(8, '0');
                    extractedText +=
biner[7];

System.Diagnostics.Debug.Write("Alamat Ke " + indexLamat + "," +
alamat[indexLamat] + " extra " + extractedText[indexLamat]);

System.Diagnostics.Debug.WriteLine(" di N " + n + " Color " + R
+ "," + G + "," + B);

                    CreateLog("Ke " +
indexLamat + "," + alamat[indexLamat] + " extract " +
extractedText[indexLamat] + " di N " + n + " Color " + R + "," +
G + "," + B);
                }
            }
        }
    }
}

```

```
        }
        break;
    }

    if (indexLamat < lamat.Length-1)
    {
        indexLamat++;
    }

    }
    S++;
}
}
}
CreateLog("Kata Biner "+ extractedText);
extractedText = toStr(extractedText);
return extractedText;
}
```

