

BAB V

PENUTUP

A. Kesimpulan

Sesuai dengan apa yang telah dibahas pada bab-bab terdahulu, maka penulis mengambil kesimpulan sebagai berikut:

1. Telah dibangunnya aplikasi kompresi *file* dengan menggunakan algoritma *Lempel-Ziv-Welch* (LZW).
2. Berdasarkan besarnya rasio kompresi yang dihasilkan dengan metode LZW tidak dipengaruhi oleh banyaknya huruf, angka atau tanda baca dalam file, tetapi dipengaruhi oleh sering dan variasi suatu string yang terdapat dalam file yang dikompresi. Besarnya waktu kompresi tergantung dari besarnya ukuran file yang akan dikompresi, dan keamanan datanya baik setelah dikompresi, dengan kata lain file tidak mengalami kerusakan setelah proses kompresi data dilakukan.

Saran

Sistem ini memiliki beberapa kekurangan yang dapat dikembangkan untuk memperbaiki kinerja sistem. Adapun saran-saran dari penulis adalah sebagai berikut:

1. Sistem dapat dikembangkan secara *online*.
2. *Input* pada proses kompresi/dekompresi *file* teks dapat dikembangkan lebih dari 1 masukan.

DAFTAR PUSTAKA

- Enterprise, Jubile. 2010. *Rahasia Menghemat Ruang Hard Disk*. PT. Elex Media Komputindo. Jakarta
- Madcoms. 2010. *Mahir Dalam 7 Hari: Microsoft Visual Basic 6.0 + Crystal Report 2008*. ANDI. Yogyakarta
- Muhammad, Alga & Subandi, Windi. *Studi Perbandingan Kinerja Metode LZW (Lempel-Ziv-Welch) Dan Metode Huffman Untuk Kompresi Data Video Dan Audio Pada Aplikasi Kompresi Data*. Skripsi. Program Studi Teknik Informatika STMIK MDP
- Putra, Darma. 2010. *Pengolahan Citra Digital*. C.V ANDI OFFSET. Yogyakarta
- Reza, Muhammad. 2010. *Implementasi Algoritma LZW (Lempel-Ziv-Welch) Pada Kompresi SMS (Short Message Service)*. Skripsi. Fakultas Teknik Informatika, Universitas Telkom
- Subarkah, Aan Fuad. 2010. *Rancang Bangun Aplikasi Kompresi File Menggunakan Metode LZW Berbasis Java*. Skripsi. Fakultas Sains Dan Teknologi Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang

LAMPIRAN

1. Modul Fungsi

```
Option Explicit

'Private Data Members
Private flgFirstTime As Boolean
Private Arr() As String
Public hasil, xxx As String
Private m_CharacterBytesCount As Integer

'Properties
Public Property Get CharacterBytesCount() As Integer
    CharacterBytesCount = m_CharacterBytesCount
End Property

Public Property Let CharacterBytesCount(ByVal vNewValue As Integer)
    m_CharacterBytesCount = vNewValue
End Property

'Private Methods
Private Function Add(ByVal Entry As String) As Long

    If flgFirstTime Then
        flgFirstTime = False
    Else
        ReDim Preserve Arr(UBound(Arr) + 1)
    End If

    Arr(UBound(Arr)) = Entry
    'Arr(UBound(Arr)).Code = ubund(

    Add = UBound(Arr)

End Function

Private Function IsExists(ByVal Entry As String) As Boolean
    Dim i As Long
    Dim flg As Boolean
    flg = False

    For i = LBound(Arr) To UBound(Arr)
        If Arr(i) = Entry Then
            flg = True
            Exit For
        End If
    Next i

    IsExists = flg

End Function

Private Function IsIndexExists(ByVal Index As Long) As Boolean

    Dim flg As Boolean
```

```

    flg = False

    If Index <= UBound(Arr) Then
        flg = True
    Else
        flg = False
    End If

    IsIndexExists = flg

End Function

Private Function GetValue(ByVal Index As Long) As String
    If IsIndexExists(Index) Then
        GetValue = Arr(Index)
    Else
        GetValue = ""
    End If
End Function

Private Function GetIndexOf(ByVal Entry As String) As Long

    Dim i As Long
    Dim ReturnCode As Long
    ReturnCode = -1

    For i = LBound(Arr) To UBound(Arr)
        If Arr(i) = Entry Then
            ReturnCode = i
            Exit For
        End If
    Next i

    GetIndexOf = ReturnCode

End Function

Private Sub Class_Initialize()
    ReDim Arr(0) As String
    flgFirstTime = True
    m_CharacterBytesCount = 2
End Sub

Private Function mrtLeft(ByVal InputString As String, ByVal
Length As Long) As String
    If Len(InputString) >= Length Then
        mrtLeft = Left(InputString, Length)
    Else
        mrtLeft = InputString
    End If
End Function

Private Sub InitializeDictionary()
    Dim i As Long
    For i = 0 To 127
        Add (ChrW(i))
    
```

```

    Next
End Sub

Private Function Length() As Long
    Length = UBound(Arr)
End Function

Public Function Compress(ByVal UncompressedData As String) As String

    Dim w, ff As String
    Dim c As String
    Dim wc As String
    'Dim dic As New Dictionary
    Dim Output, output1 As String
    Dim Code As Long

    Dim i As Long
    InitializeDictionary

    w = ""
    For i = 1 To Len(UncompressedData)

        c = Mid(UncompressedData, i, 1)
        frmMain.Text1(i) = c
        wc = w + c
        frmMain.Text2(i) = wc

        If IsExists(wc) Then
            w = wc
            frmMain.Text3(i) = "YES"
            frmMain.Text4(i) = w
            frmMain.Text5(i).Visible = False
        Else

            Code = Add(wc)
            frmMain.Text9(i) = Code
            Code = GetIndexOf(w)
            frmMain.Text6(i) = Code

            'frmMain.Text7(i) = Chr(Code)
            Output = Output & IIf(Output = "", "", "-") &
Code
            output1 = output1 & IIf(output1 = "", "", "-") &
Chr(Code)

            w = c
            frmMain.Text5(i) = w

            frmMain.Text4(i).Visible = False
            frmMain.Text3(i) = "NO"
        End If

    Next i
    Output = Output & IIf(Output = "", "", "-") & GetIndexOf(w)

```

```

output1 = output1 & IIf(output1 = "", "", "-") &
Chr(GetIndexof(w))
    frmMain.Text6(i) = GetIndexof(w)

    Compress = Output
frmMain.Text12 = output1
End Function

Public Function Decompress(ByVal CompressedData As String) As
String

    If CompressedData = "" Then Exit Function

    'Dim dic As New Dictionary

    InitializeDictionary

    Dim Output As String
    Dim Entry As String
Dim Code As Long
    Dim k As String
    Dim w As String
    Dim i As Long

    Dim arrCompressedData() As String

    'FUNGSI SPLIT MEMECAHKAN KARAKTER
arrCompressedData = Split(CompressedData, "-")

    k = arrCompressedData(0)

    Output = Output & GetValue(k)
    DECODING.Text3(i) = Output
    DECODING.Text6(i) = Output

    w = k
    DECODING.Text1(i) = w

    For i = 1 To UBound(arrCompressedData)

        k = arrCompressedData(i)
        DECODING.Text1(i) = k
        If IsIndexExists(k) Then

            Entry = GetValue(k)

        ElseIf IsIndexExists(k) = False And k = Length + 1 Then

            Entry = GetValue(w) + mrtLeft(GetValue(w), 1)

        Else

            MsgBox "Bad compressed."

```

```

        End If

        Output = Output + Entry
        DECODING.Text4(i) = Entry
        DECODING.Text6(i) = Output
        DECODING.Text3(i) = GetValue(w)
        Add (GetValue(w) + mrtLeft(Entry, 1))

        DECODING.Text5(i) = (GetValue(w) + mrtLeft(Entry, 1))
        w = GetIndexOf(Entry)
    Next
    Decompress = Output

End Function

Public Function byner()
Dim Temp, Temp1 As Long
Dim i As Integer
Dim res As String
Dim Done As Boolean

Temp = Text1.Text
Temp1 = Text1.Text
Do Until Temp \ 2 = 1
    Temp = Temp \ 2
    i = i + 1
Loop
res = ""
For j = i + 1 To 0 Step -1
    If Temp1 - 2 ^ j > 0 And Done = False Then
        Temp1 = Temp1 - 2 ^ j
        res = res & "1"
    ElseIf Temp1 - 2 ^ j <> 0 And Done = False Then
        res = res & "0"
    ElseIf Temp1 - 2 ^ j = 0 And Done = False Then
        res = res & "1"
        Done = True
    ElseIf Done = True Then
        res = res & "0"
    End If
Next
Text2.Text = res
End Function

```

2. Form Utama

```

Private Sub Command1_Click()
Kompresi.Show
End Sub
Private Sub Command2_Click()
Dekompres.Show 1
End Sub
Private Sub Command3_Click()
Kompresi.Show
End Sub

```



```

Private Sub Dekompresi_Click()
DECODING.Show 1
End Sub

Private Sub Dekompresi1_Click()
DECODING.Show 1
End Sub

Private Sub Form_Unload(Cancel As Integer)
End
End Sub

Private Sub Kompresi1_Click()
frmMain.Show 1
End Sub

```

3. Form Simulasi Kompresi

```

Public xxx, x1 As String

Private Sub cmdCompress_Click()
Dim nn As String

nn = Len(frmMain.txtUncompressedData.Text)
If nn > 20 Then
MsgBox "String terlalu panjang"
txtUncompressedData = ""
txtCompressedData = ""
For o = 0 To 21
Text1(o) = ""
Text2(o) = ""
Text3(o) = ""
Text4(o) = ""
Text5(o) = ""
Text6(o) = ""
Text9(o) = ""
Text4(o).Visible = True
Text5(o).Visible = True
Next o
Exit Sub
End If

For o = 0 To 21
Text1(o) = ""
Text2(o) = ""
Text3(o) = ""
Text4(o) = ""
Text5(o) = ""
Text6(o) = ""
Text9(o) = ""
Text4(o).Visible = True
Text5(o).Visible = True
Next o
    If Me.txtUncompressedData.Text = "" Then
        MsgBox "Inputkan teks untuk dikompres"
    End If

```

```

Exit Sub
End If

Dim lzw As New lzw
Dim t1, t2
t1 = Now
Me.txtCompressedData =
lzw.Compress (Me.txtUncompressedData)

t2 = Now

Dim c As Long, i, aa As Long
c = 0
For i = 1 To Len(txtCompressedData.Text)
    If Mid(txtCompressedData.Text, i, 1) = "-" Then
        c = c + 1
        Text8 = c + 1
    End If
Next i

hasil = frmMain.txtCompressedData.Text

Text7 = hasil

Text11 = Len(Me.txtUncompressedData.Text)
MsgBox "Data Compressed." & vbNewLine & _
    "Ratio: " & Round(Text8 /
Len(Me.txtUncompressedData.Text) * 100, 2) & "%" & vbNewLine
& _
    "Compression took: " & Format$(t2 - t1, "s") & "
seconds."
xxx = txtCompressedData
x1 = Text12

DECODING.txtCompressedData = xxx
DECODING.Text12 = x1
End Sub

Public Function DecToBiner(ByVal vNewValue As Double) As
String
    Dim strTmp As String
    Dim dblBagi As Double, intSisa As Integer

    dblBagi = vNewValue
    Do While dblBagi <> 0
        intSisa = dblBagi Mod 2
        dblBagi = RoundDown(dblBagi / 2)
        strTmp = Trim(Str(intSisa) & strTmp)
    Loop
    DecToBiner = strTmp
End Function

Private Sub Form_Activate()
txtUncompressedData = "wabbawabba"

```

```
End Sub
```

4. Form Simulasi Dekompresi

```
Public xxx As String
Private Sub cmdCompress_Click()
End Sub
Private Sub cmdDecompress_Click()
For o = 0 To 18
Text1(o) = ""
Text3(o) = ""
Text4(o) = ""
Text5(o) = ""
Text6(o) = ""
Next o
    If Me.txtCompressedData.Text = "" Then
        MsgBox "Please enter some text to be decompressed."
        Exit Sub
    End If

    Dim lzw As New lzw
    Dim t1, t2
    t1 = Now
    Me.txtUncompressedData.Text =
lzw.Decompress(Me.txtCompressedData.Text)
    t2 = Now
    MsgBox "Data decompressed." & vbNewLine & _
        "Decompression took: " & Format$(t2 - t1, "s") & "
seconds."

End Sub

Private Sub Form_Activate()
txtCompressedData.SetFocus
End Sub
```

5. Form Kompresi

```
Private Dict(0 To 255) As String
Private count1 As Byte
Private Sub Init()
For i = 0 To 255
    Dict(i) = Chr(i)
Next
End Sub
Private Function Search(inp As String) As Integer
For i = 0 To 255
    If Dict(i) = inp Then Search = i: Exit Function
Next
Search = 256
End Function
Private Sub Add(inp As String)
If count1 = 256 Then Wipe
```

```

Dict(count1) = inp
On Error Resume Next

count1 = count1 + 1
End Sub
Private Sub Wipe()
For i = 128 To 255
    Dict(i) = ""
Next
count1 = 128
End Sub
Public Function Deflate(inp As String) As String
'Begin Error Checking
For i = 1 To Len(inp)
    If Asc(Mid(inp, i, 1)) > 127 Then Exit Function
Next
'End Error Checking
Init
Wipe
p = ""
i = 1
Do Until i > Len(inp)
    c = Mid(inp, i, 1)
    i = i + 1
    Temp = p & c
    If Not Search(CStr(Temp)) = 256 Then
        p = Temp

    Else
        o = o & Chr(Search(CStr(p)))
        Add CStr(Temp)
        p = c
    End If
Loop
o = o & Chr(Search(CStr(p)))
Deflate = o
End Function
Public Function Inflate(inp As String) As String
Init
Wipe

cw = Asc(Mid(inp, 1, 1))
o = Dict(cw)
i = 2
Do Until i > Len(inp)
    pw = cw
    cw = Asc(Mid(inp, i, 1))
    i = i + 1
    If Not Dict(cw) = "" Then
        o = o & Dict(cw)
        p = Dict(pw)
        c = Mid(Dict(cw), 1, 1)
        Add (CStr(p) & CStr(c))
    ElseIf Dict(cw) = "" Then
        p = Dict(pw)
        c = Mid(Dict(pw), 1, 1)

```

```

        o = o & p & c
        Add (CStr(p) & CStr(c))
    End If
Loop
Inflate = o
End Function
Public Sub main()
inp = "MAMA MASAK SAGU"
d = Deflate(CStr(inp)) 'Compress
q = Inflate(CStr(d)) 'Uncompress
MsgBox "Uncompressed: " & q & vbCrLf & vbCrLf & _
    "Compressed: " & d & vbCrLf & vbCrLf & _
    "Compressed Size: " & Len(d) & vbCrLf & vbCrLf & _
    "Uncompressed Size: " & Len(q) & vbCrLf & vbCrLf & _
    "Compression Ratio: " & (100 - ((Len(d) / Len(q)) *
100) \ 1) & _
    "%", vbOKOnly, "Results:"
End Sub
Private Sub Command1_Click()
    Dim t1, t2
    t1 = Now
    txtCompressedData.Text =
Deflate(CStr(txtUncompressedData.Text))
    t2 = Now
    Text4 = Format$(t2 - t1, "s") & " seconds."

'd = Deflate(CStr(txtUncompressedData.Text)) 'Compress
'q = Inflate(CStr(d)) 'Uncompress

Text1 = Len(txtUncompressedData)
Text2 = Len(txtCompressedData)

Label12 = Len(txtUncompressedData)
Label11 = Len(txtCompressedData)
Label15 = Round(Len(txtCompressedData) /
Len(txtUncompressedData) * 100, 2) & "%"
' MsgBox "Uncompressed: " & q & vbCrLf & vbCrLf & _
'     "Compressed: " & d & vbCrLf & vbCrLf & _
'     "Compressed Size: " & Len(d) & vbCrLf & vbCrLf & _
'     "Uncompressed Size: " & Len(q) & vbCrLf & vbCrLf & _
'     "Compression Ratio: " & Round(Len(d) / Len(q) * 100,
2) & "%" & vbCrLf & _
'     "Compression took: " & Format$(t2 - t1, "s") & "
seconds."
Frame1.Visible = True
End Sub

Private Sub Command2_Click()
txtCompressedData = Asc(txtUncompressedData)
txtUncompressedData = Inflate(CStr(txtCompressedData))
Unload Me

End Sub

Private Sub Command3_Click()
    Dim cFileSave As String

```

```

Dim nBaris As Integer

On Error GoTo errSaveCancel

If txtCompressedData.Text = "" Then Exit Sub

'Save File
With oCD
    .InitDir = App.Path & "\"
    .Flags = cdLOFNPathMustExist Or cdLOFNOverwritePrompt
    .ShowSave
    cFileSave = .FileName
End With

If cFileSave = "" Then Exit Sub

'Save File
Open cFileSave For Output As #1

    Print #1, txtCompressedData.Text

Close #1

errSaveCancel:

'filenum = FreeFile
'Open App.Path & "\" & "file.txt" For Output As filenum
'Print #filenum, Trim(txtCompressedData.Text)
'Close filenum
End Sub

Private Sub Command4_Click()
Frame1.Visible = False
txtCompressedData = ""
txtUncompressedData = ""
Text3.Text = ""

Dim cLoadFile, n As Long

On Error Resume Next
    With oCD
        .Filter = "File Gambar (*.txt,*.tst)|*.txt|,|*.tst|"
        .InitDir = App.Path & "\"
        .Flags = cdLOFNPathMustExist Or cdLOFNFileMustExist
        .ShowOpen
        cLoadFile = .FileName
    End With

If oCD.FileName = "" Then Exit Sub
If oCD.FileName <> "" Then
    Text3.Text = oCD.FileName
End If

Dim m As Integer
m = FreeFile
Open Text3.Text For Input As #m

```

```

txtUncompressedData = Input(LOF(m), m)
Command1.Enabled = True
Close #m
End Sub

```

6. Form Dekompresi

```

Private Dict(0 To 255) As String
Private count1 As Byte
Private Sub Init()
For i = 0 To 255
    Dict(i) = Chr(i)
Next
End Sub
Private Function Search(inp As String) As Integer
For i = 0 To 255
    If Dict(i) = inp Then Search = i: Exit Function
Next
Search = 256
End Function
Private Sub Add(inp As String)
If count1 = 256 Then Wipe
Dict(count1) = inp

On Error Resume Next
count1 = count1 + 1

End Sub
Private Sub Wipe()
For i = 128 To 255
    Dict(i) = ""
Next
count1 = 128
End Sub
Public Function Deflate(inp As String) As String
'Begin Error Checking
For i = 1 To Len(inp)
    If Asc(Mid(inp, i, 1)) > 127 Then Exit Function
Next
'End Error Checking
Init
Wipe
p = ""
i = 1
Do Until i > Len(inp)
    c = Mid(inp, i, 1)
    i = i + 1
    Temp = p & c
    If Not Search(CStr(Temp)) = 256 Then
        p = Temp

    Else
        o = o & Chr(Search(CStr(p)))
        Add CStr(Temp)
    End If

```

```

        p = c
    End If
Loop
o = o & Chr(Search(CStr(p)))
Deflate = o
End Function
Public Function Inflate(inp As String) As String
Init
Wipe

cw = Asc(Mid(inp, 1, 1))
o = Dict(cw)
i = 2
Do Until i > Len(inp)
    pw = cw
    cw = Asc(Mid(inp, i, 1))
    i = i + 1
    If Not Dict(cw) = "" Then
        o = o & Dict(cw)
        p = Dict(pw)
        c = Mid(Dict(cw), 1, 1)
        Add (CStr(p) & CStr(c))
    ElseIf Dict(cw) = "" Then
        p = Dict(pw)
        c = Mid(Dict(pw), 1, 1)
        o = o & p & c
        Add (CStr(p) & CStr(c))
    End If
Loop
Inflate = o
End Function
Public Sub main()
inp = "MAMA MASAK SAGU"
d = Deflate(CStr(inp)) 'Compress
q = Inflate(CStr(d)) 'Uncompress
MsgBox "Uncompressed: " & q & vbCrLf & vbCrLf & _
    "Compressed: " & d & vbCrLf & vbCrLf & _
    "Compressed Size: " & Len(d) & vbCrLf & vbCrLf & _
    "Uncompressed Size: " & Len(q) & vbCrLf & vbCrLf & _
    "Compression Ratio: " & (100 - ((Len(d) / Len(q)) *
100) \ 1) & _
    "%", vbOKOnly, "Results:"
End Sub
Frame1.Visible = True
End Sub

Private Sub Command1_Click()
'txtCompressedData = Asc(txtUncompressedData)
txtUncompressedData = Inflate(CStr(txtCompressedData))
Text1 = Len(txtUncompressedData)
    Text2 = Len(txtCompressedData)
End Sub

Private Sub Command4_Click()
txtCompressedData = ""

```



```
txtUncompressedData = ""
Text3.Text = ""

Dim cLoadFile, n As Long

On Error Resume Next
    With oCD
        .Filter = "File Gambar (*.txt,*.txt)|*.txt|,|*.txt|"
        .InitDir = App.Path & "\"
        .Flags = cdlOFNPathMustExist Or cdlOFNFileMustExist
        .ShowOpen
        cLoadFile = .FileName
    End With

If oCD.FileName = "" Then Exit Sub
If oCD.FileName <> "" Then
    Text3.Text = oCD.FileName
End If

Dim m As Integer
m = FreeFile
Open Text3.Text For Input As #m

txtCompressedData = Input(LOF(m), m)
Command1.Enabled = True
Close #m
End Sub
```