LAMPIRAN A

PROGRAM PADA VISUAL BASIC

PROGRAM UTAMA

```
Dim h As Integer, i As Integer, j As Integer
Dim r(16) As Integer
Dim g(16) As Integer
Dim b(16) As Integer


Private Sub Command1_Click()
CommonDialog1.ShowOpen
Picture1.Picture = LoadPicture(CommonDialog1.FileName)
Text1.Text = CommonDialog1.FileName
End Sub
'Gaussian Noise
Private Sub Command2_Click()
probnoise = Val(HScroll1) / 100
Label5.Caption = Val(HScroll1)
Picture1.AutoRedraw = True
Picture2.AutoRedraw = True
Picture5.AutoRedraw = True
‘Pindai gambar awal
For i = 1 To Picture1.Width Step 1
For j = 1 To Picture1.Height Step 1
        warna = Picture1.Point(i, j)
                rr = warna And RGB(255, 0, 0)
                gg = Int((warna And RGB(0, 255, 0)) / 256)
                bb = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)
 ' Merubah RGB ke YCrcB
                Y = 0.299 * rr + 0.587 * gg + 0.114 * bb
                cb = -0.16874 * rr - 0.33126 * gg + 0.5 * bb + 128
                cr = 0.5 * rr - 0.41869 * gg - 0.08131 * bb + 128
‘Noise Gaussian PSF
a = 0
While a = 0
        X = 2 * Rnd - 1
        yy = Rnd
        If yy < Exp(-X ^ 2) Then
         a = X
        End If
Wend
‘Pemberian Noise Gaussian PSF
Y1 = Y
'Y1 = Abs(Y - A * 127 * probnoise)
'If Y1 > 255 Then Y1 = 255
cb1 = Abs(cb - a * 127 * probnoise)
'If cb1 > 255 Then cb1 = 255
cr1 = Abs(cr - a * 127 * probnoise)
'If cr1 > 255 Then cr1 = 255
'Merubah YCrcB ke RGB
r1 = Abs(Y1 + 1.371 * (cr1 - 128))
g1 = Abs(Y1 - 0.698 * (cr1 - 128) - 0.336 * (cb1 - 128))
b1 = Abs(Y1 + 1.732 * (cb1 - 128))
'h(x + 1) = h(x + 1) + 1
```

```
Picture2.PSet (i, j), RGB(r1, g1, b1)
Picture5.PSet (i, j), RGB(r1, g1, b1)
Next j
Next i
‘ MENGHITUNG PSNR
n = 3
hitung = 1
X = Picture2.ScaleWidth
Y = Picture2.ScaleHeight
For brs = 1 To Y Step (n + 1) * 1
   For klm = 1 To X Step (n + 1) * 1
    mseR = 0
    mseB = 0
    mseG = 0
              For m = 0 To n * 1 Step 1
               For n1 = 0 To n * 1 Step 1
                  warna = Picture1.Point(klm + n1, brs + m)
                  rr = warna And RGB(255, 0, 0)
                  gg = Int((warna And RGB(0, 255, 0)) / 256)
                  bg = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

                  warna5 = Picture2.Point(klm + n1, brs + m)
                  r1 = warna5 And RGB(255, 0, 0)
                  g1 = Int((warna5 And RGB(0, 255, 0)) / 256)
                  b1 = Int(Int((warna5 And RGB(0, 0, 255)) / 256) / 256)

                  mseR = mseR + (r1 - rr) ^ 2
                  mseB = mseB + (g1 - gg) ^ 2
                  mseG = mseG + (b1 - bb) ^ 2

               Next n1
            Next m
        mse = (mseR + mseB + mseG) / (3 * (n * 1) * (n * 1))
        If mse = 0 Then mse = 1
       ' psnr = Abs(20 * (Log(255 / (mse ^ (0.5))) / Log(10)))
        psnr = psnr + Abs(10 * (Log((255 ^ 2) / mse) / Log(10)))
        hitung = hitung + 1
    Next klm
Next brs
psnr = psnr / (hitung - 1)
Label9.Caption = "PSNR =" + Str(psnr)
End Sub

Private Sub Command3_Click()
'denoising filter gaussian

Picture3.AutoRedraw = True
X = Picture2.ScaleWidth
Y = Picture2.ScaleHeight
‘MATRIKS FILTER GAUSSIAN
m1 = 0.077
m2 = 0.077
m3 = 0.077
m4 = 0.077
m5 = 0.308
```

```
m6 = 0.077
m7 = 0.077
m8 = 0.077
m9 = 0.077
mulai3= Time
For brs = 2 To Y - 1 Step 1
   For klm = 2 To X - 1 Step 1

      a1 = Picture2.Point(klm - 1, brs - 1)
      r1 = a1 And RGB(255, 0, 0)
      g1 = (a1 And RGB(0, 255, 0)) \ 256
      b1 = (a1 And RGB(0, 0, 255)) \ 256 \ 256

      a2 = Picture2.Point(klm, brs - 1)
      r2 = a2 And RGB(255, 0, 0)
      g2 = (a2 And RGB(0, 255, 0)) \ 256
      b2 = (a2 And RGB(0, 0, 255)) \ 256 \ 256

      a3 = Picture2.Point(klm + 1, brs - 1)
      r3 = a3 And RGB(255, 0, 0)
      g3 = (a3 And RGB(0, 255, 0)) \ 256
      b3 = (a3 And RGB(0, 0, 255)) \ 256 \ 256

      a4 = Picture2.Point(klm - 1, brs)
      r4 = a4 And RGB(255, 0, 0)
      g4 = (a4 And RGB(0, 255, 0)) \ 256
      b4 = (a4 And RGB(0, 0, 255)) \ 256 \ 256

      a5 = Picture2.Point(klm, brs)
      r5 = a5 And RGB(255, 0, 0)
      g5 = (a5 And RGB(0, 255, 0)) \ 256
      b5 = (a5 And RGB(0, 0, 255)) \ 256 \ 256

      a6 = Picture2.Point(klm + 1, brs)
      r6 = a6 And RGB(255, 0, 0)
      g6 = (a6 And RGB(0, 255, 0)) \ 256
      b6 = (a6 And RGB(0, 0, 255)) \ 256 \ 256

      a7 = Picture2.Point(klm - 1, brs + 1)
      r7 = a7 And RGB(255, 0, 0)
      g7 = (a7 And RGB(0, 255, 0)) \ 256
      b7 = (a7 And RGB(0, 0, 255)) \ 256 \ 256

      a8 = Picture2.Point(klm, brs + 1)
      r8 = a8 And RGB(255, 0, 0)
      g8 = (a8 And RGB(0, 255, 0)) \ 256
      b8 = (a8 And RGB(0, 0, 255)) \ 256 \ 256

      a9 = Picture2.Point(klm + 1, brs + 1)
      r9 = a9 And RGB(255, 0, 0)
      g9 = (a9 And RGB(0, 255, 0)) \ 256
      b9 = (a9 And RGB(0, 0, 255)) \ 256 \ 256

      h = Abs(r1 * m9 + r2 * m8 + r3 * m7 + r4 * m6 + r5 * m5 + r6 * m4 + r7 * m3 + r8 * m2
+ r9 * m1)
```

```
    i = Abs(g1 * m9 + g2 * m8 + g3 * m7 + g4 * m6 + g5 * m5 + g6 * m4 + g7 * m3 + g8 * m2
+ g9 * m1)
    j = Abs(b1 * m9 + b2 * m8 + b3 * m7 + b4 * m6 + b5 * m5 + b6 * m4 + b7 * m3 + b8 *
m2 + b9 * m1)
    Picture3.PSet (klm, brs), RGB(h, i, j)
    Picture4.PSet (klm, brs), RGB(h, i, j)
  Next klm
Next brs
selesai3 = Time
selisih3 = selesai3 - mulai3
Form1.Refresh
‘MENGHITUNG PSNR
n = 3
hitung = 1
X = Picture3.ScaleWidth
Y = Picture3.ScaleHeight
For brs = 1 To Y Step (n + 1) * 1
  For klm = 1 To X Step (n + 1) * 1
  mseR = 0
  mseB = 0
  mseG = 0
      For m = 0 To n * 1 Step 1
        For n1 = 0 To n * 1 Step 1
        warna = Picture1.Point(klm + n1, brs + m)
          rr = warna And RGB(255, 0, 0)
          gg = Int((warna And RGB(0, 255, 0)) / 256)
          bg = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

          warna5 = Picture3.Point(klm + n1, brs + m)
          r1 = warna5 And RGB(255, 0, 0)
          g1 = Int((warna5 And RGB(0, 255, 0)) / 256)
          b1 = Int(Int((warna5 And RGB(0, 0, 255)) / 256) / 256)

          mseR = mseR + (r1 - rr) ^ 2
          mseB = mseB + (g1 - gg) ^ 2
          mseG = mseG + (b1 - bb) ^ 2

        Next n1
      Next m
    mse = (mseR + mseB + mseG) / (3 * (n * 1) * (n * 1))
    If mse = 0 Then mse = 1
    'psnr = Abs(20 * (Log(255 / (mse ^ (0.5))) / Log(10)))
    psnr = psnr + Abs(10 * (Log((255 ^ 2) / mse) / Log(10)))
    hitung = hitung + 1
  Next klm
Next brs
psnr = psnr / (hitung - 1)
Label6.Caption = "PSNR =" + Str(psnr)
End Sub
Private Sub Command4_Click()
'Interpolasi Bicubic dengan denoising
Picture4.AutoRedraw = False
X = Picture3.ScaleWidth
Y = Picture3.ScaleHeight
List1.Clear
```

```
 mulai = Time
For brs = 2 To Y - 1 Step 1
  For klm = 2 To X - 1 Step 1


    a1 = Picture3.Point(klm - 1, brs - 1)
    r1 = a1 And RGB(255, 0, 0)
    g1 = (a1 And RGB(0, 255, 0)) \ 256
    b1 = (a1 And RGB(0, 0, 255)) \ 256 \ 256

    a2 = Picture3.Point(klm, brs - 1)
    r2 = a2 And RGB(255, 0, 0)
    g2 = (a2 And RGB(0, 255, 0)) \ 256
    b2 = (a2 And RGB(0, 0, 255)) \ 256 \ 256

    a3 = Picture3.Point(klm + 1, brs - 1)
    r3 = a3 And RGB(255, 0, 0)
    g3 = (a3 And RGB(0, 255, 0)) \ 256
    b3 = (a3 And RGB(0, 0, 255)) \ 256 \ 256

    a4 = Picture3.Point(klm + 2, brs - 1)
    r4 = a4 And RGB(255, 0, 0)
    g4 = (a4 And RGB(0, 255, 0)) \ 256
    b4 = (a4 And RGB(0, 0, 255)) \ 256 \ 256

    a5 = Picture3.Point(klm - 1, brs)
    r5 = a5 And RGB(255, 0, 0)
    g5 = (a5 And RGB(0, 255, 0)) \ 256
    b5 = (a5 And RGB(0, 0, 255)) \ 256 \ 256

    a6 = Picture3.Point(klm, brs)
    r6 = a6 And RGB(255, 0, 0)
    g6 = (a6 And RGB(0, 255, 0)) \ 256
    b6 = (a6 And RGB(0, 0, 255)) \ 256 \ 256

    a7 = Picture3.Point(klm + 1, brs)
    r7 = a7 And RGB(255, 0, 0)
    g7 = (a7 And RGB(0, 255, 0)) \ 256
    b7 = (a7 And RGB(0, 0, 255)) \ 256 \ 256

    a8 = Picture3.Point(klm + 2, brs)
    r8 = a8 And RGB(255, 0, 0)
    g8 = (a8 And RGB(0, 255, 0)) \ 256
    b8 = (a8 And RGB(0, 0, 255)) \ 256 \ 256

    a9 = Picture3.Point(klm - 1, brs + 1)
    r9 = a9 And RGB(255, 0, 0)
    g9 = (a9 And RGB(0, 255, 0)) \ 256
    b9 = (a9 And RGB(0, 0, 255)) \ 256 \ 256

    a10 = Picture3.Point(klm, brs + 1)
    r10 = a10 And RGB(255, 0, 0)
    g10 = (a10 And RGB(0, 255, 0)) \ 256
    b10 = (a10 And RGB(0, 0, 255)) \ 256 \ 256
    a11 = Picture3.Point(klm + 1, brs + 1)
    r11 = a11 And RGB(255, 0, 0)
    g11 = (a11 And RGB(0, 255, 0)) \ 256
    b11 = (a11 And RGB(0, 0, 255)) \ 256 \ 256

    a12 = Picture3.Point(klm + 2, brs + 1)
    r12 = a12 And RGB(255, 0, 0)
    g12 = (a12 And RGB(0, 255, 0)) \ 256
    b12 = (a12 And RGB(0, 0, 255)) \ 256 \ 256

    a13 = Picture3.Point(klm - 1, brs + 2)
    r13 = a13 And RGB(255, 0, 0)
    g13 = (a13 And RGB(0, 255, 0)) \ 256
    b13 = (a13 And RGB(0, 0, 255)) \ 256 \ 256

    a14 = Picture3.Point(klm, brs + 2)
    r14 = a14 And RGB(255, 0, 0)
    g14 = (a14 And RGB(0, 255, 0)) \ 256
    b14 = (a14 And RGB(0, 0, 255)) \ 256 \ 256

    a15 = Picture3.Point(klm + 1, brs + 2)
    r15 = a15 And RGB(255, 0, 0)
    g15 = (a15 And RGB(0, 255, 0)) \ 256
    b15 = (a15 And RGB(0, 0, 255)) \ 256 \ 256

    a16 = Picture3.Point(klm + 2, brs + 2)
    r16 = a16 And RGB(255, 0, 0)
    g16 = (a16 And RGB(0, 255, 0)) \ 256
    b16 = (a16 And RGB(0, 0, 255)) \ 256 \ 256
  'Pengisian matrix W(sy) dan W(sx)

  sy = 1 / 3
   e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
   e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
   e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
   e3 = ((sy ^ 3) - (sy ^ 2)) / 2
   sx = 1 / 3
   f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
   f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
   f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
   f3 = ((sx ^ 3) - (sx ^ 2)) / 2
   'Perkalian W(sy) dengan matrix 4x4 = matrix h
   h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
   h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
   h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
   h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
     'Perkalian matrix h dengan W(sx) = matrix m
 m00 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
        'Perkalian W(sy) dengan matrix 4x4 = matrix p
       p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
       p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
       p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
       p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
'Perkalian matrix p dengan W(sx) = matrix q
```

```
    q00 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
       'Perkalian W(sy) dengan matrix 4x4 = matrix s
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        'Perkalian matrix s dengan W(sx) = matrix t

      t00 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)




sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m01 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q01 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t01 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
```

```
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m02 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q02 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t02 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m03 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q03 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t03 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
```

```
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m10 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q10 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t10 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)

sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m11 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q11 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t11 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
```

```
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m12 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q12 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t12 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m13 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q13 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t13 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
```

```
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m20 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
      p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
      p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
      p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
      p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
      q20 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
             s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
             s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
             s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
             s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
             t20 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m21 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
      p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
      p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
      p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
      p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
      q21 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
             s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
             s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
             s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
             s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
             t21 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
```

```
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m22 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q22 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
         s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
         s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
         s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
         s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
         t22 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m23 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q23 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
         s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
         s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
         s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
         s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
         t23 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
```

```
sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m30 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q30 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t30 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m31 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q31 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t31 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m32 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q32 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t32 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m33 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q33 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
```

```
          s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
          s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
          s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
          t33 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)


  zr = (q00 + q01 + q02 + q03 + q10 + q11 + q12 + q13 + q20 + q21 + q22 + q23 + q30 + q31
+ q32 + q33) / 16
  zg = (m00 + m01 + m02 + m03 + m10 + m11 + m12 + m13 + m20 + m21 + m22 + m23 +
m30 + m31 + m32 + m33) / 16
  zb = (t00 + t01 + t02 + t03 + t10 + t11 + t12 + t13 + t20 + t21 + t22 + t23 + t30 + t31 + t32 +
t33) / 16
  zr = Int(zr)
  zg = Int(zg)
  zb = Int(zb)
  If zr < 0 Then zr = r6
  If zg < 0 Then zg = g6
  If zb < 0 Then zb = b6
  'List1.AddItem (Str(klm) + "," + Str(brs) + " = " + Str(zr) + "," + Str(zg) + "," + Str(zb))
  Picture4.PSet (klm, brs), RGB(zr, zg, zb)
  Next klm
  Next brs
  selesai = Time
  beda = (selesai - mulai)
  total = selisih3 + beda
  Text2.Text = Format(total, "h:m:s")
  Form1.Refresh
 ‘menghitung PSNR
n = 3
hitung = 1
X = Picture4.ScaleWidth
Y = Picture4.ScaleHeight
For brs = 1 To Y Step (n + 1) * 1
  For klm = 1 To X Step (n + 1) * 1
  mseR = 0
  mseB = 0
  mseG = 0
      For m = 0 To n * 1 Step 1
        For n1 = 0 To n * 1 Step 1
        warna = Picture1.Point(klm + n1, brs + m)
          rr = warna And RGB(255, 0, 0)
          gg = Int((warna And RGB(0, 255, 0)) / 256)
          bg = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

          warna5 = Picture4.Point(klm + n1, brs + m)
          r1 = warna5 And RGB(255, 0, 0)
          g1 = Int((warna5 And RGB(0, 255, 0)) / 256)
          b1 = Int(Int((warna5 And RGB(0, 0, 255)) / 256) / 256)

          mseR = mseR + (r1 - rr) ^ 2
          mseB = mseB + (g1 - gg) ^ 2
          mseG = mseG + (b1 - bb) ^ 2

       Next n1
      Next m
```

```
        mse = (mseR + mseB + mseG) / (3 * (n * 1) * (n * 1))
        If mse = 0 Then mse = 1
        'psnr = Abs(20 * (Log(255 / (mse ^ (0.5))) / Log(10)))
        psnr = psnr + Abs(10 * (Log((255 ^ 2) / mse) / Log(10)))
        hitung = hitung + 1
    Next klm
Next brs

psnr = psnr / (hitung - 1)
Label7.Caption = "PSNR =" + Str(psnr)
End Sub
‘ SAVE
Private Sub Command5_Click()
CommonDialog2.Filter = "Bitmap (*.BMP)|*.bmp"
CommonDialog2.ShowSave
CommonDialog3.Filter = "Bitmap (*.BMP)|*.bmp"
CommonDialog3.ShowSave
CommonDialog4.Filter = "Bitmap (*.BMP)|*.bmp"
CommonDialog4.ShowSave
CommonDialog5.Filter = "Bitmap (*.BMP)|*.bmp"
CommonDialog5.ShowSave
SavePicture Picture2.Image, CommonDialog2.FileName
SavePicture Picture3.Image, CommonDialog3.FileName
SavePicture Picture4.Image, CommonDialog4.FileName
SavePicture Picture5.Image, CommonDialog5.FileName
End Sub
'Interpolasi Bicubic tanpa denoising
Private Sub Command6_Click()
Picture5.AutoRedraw = False
X = Picture2.ScaleWidth
Y = Picture2.ScaleHeight
List1.Clear
 mulai = Time
For brs = 2 To Y - 1 Step 1
    For klm = 2 To X - 1 Step 1


        a1 = Picture2.Point(klm - 1, brs - 1)
        r1 = a1 And RGB(255, 0, 0)
        g1 = (a1 And RGB(0, 255, 0)) \ 256
        b1 = (a1 And RGB(0, 0, 255)) \ 256 \ 256

        a2 = Picture2.Point(klm, brs - 1)
        r2 = a2 And RGB(255, 0, 0)
        g2 = (a2 And RGB(0, 255, 0)) \ 256
        b2 = (a2 And RGB(0, 0, 255)) \ 256 \ 256

        a3 = Picture2.Point(klm + 1, brs - 1)
        r3 = a3 And RGB(255, 0, 0)
        g3 = (a3 And RGB(0, 255, 0)) \ 256
        b3 = (a3 And RGB(0, 0, 255)) \ 256 \ 256

        a4 = Picture2.Point(klm + 2, brs - 1)
        r4 = a4 And RGB(255, 0, 0)
        g4 = (a4 And RGB(0, 255, 0)) \ 256
```

```
b4 = (a4 And RGB(0, 0, 255)) \ 256 \ 256

a5 = Picture2.Point(klm - 1, brs)
r5 = a5 And RGB(255, 0, 0)
g5 = (a5 And RGB(0, 255, 0)) \ 256
b5 = (a5 And RGB(0, 0, 255)) \ 256 \ 256

a6 = Picture2.Point(klm, brs)
r6 = a6 And RGB(255, 0, 0)
g6 = (a6 And RGB(0, 255, 0)) \ 256
b6 = (a6 And RGB(0, 0, 255)) \ 256 \ 256

a7 = Picture2.Point(klm + 1, brs)
r7 = a7 And RGB(255, 0, 0)
g7 = (a7 And RGB(0, 255, 0)) \ 256
b7 = (a7 And RGB(0, 0, 255)) \ 256 \ 256

a8 = Picture2.Point(klm + 2, brs)
r8 = a8 And RGB(255, 0, 0)
g8 = (a8 And RGB(0, 255, 0)) \ 256
b8 = (a8 And RGB(0, 0, 255)) \ 256 \ 256

a9 = Picture2.Point(klm - 1, brs + 1)
r9 = a9 And RGB(255, 0, 0)
g9 = (a9 And RGB(0, 255, 0)) \ 256
b9 = (a9 And RGB(0, 0, 255)) \ 256 \ 256

a10 = Picture2.Point(klm, brs + 1)
r10 = a10 And RGB(255, 0, 0)
g10 = (a10 And RGB(0, 255, 0)) \ 256
b10 = (a10 And RGB(0, 0, 255)) \ 256 \ 256

a11 = Picture2.Point(klm + 1, brs + 1)
r11 = a11 And RGB(255, 0, 0)
g11 = (a11 And RGB(0, 255, 0)) \ 256
b11 = (a11 And RGB(0, 0, 255)) \ 256 \ 256

a12 = Picture2.Point(klm + 2, brs + 1)
r12 = a12 And RGB(255, 0, 0)
g12 = (a12 And RGB(0, 255, 0)) \ 256
b12 = (a12 And RGB(0, 0, 255)) \ 256 \ 256

a13 = Picture2.Point(klm - 1, brs + 2)
r13 = a13 And RGB(255, 0, 0)
g13 = (a13 And RGB(0, 255, 0)) \ 256
b13 = (a13 And RGB(0, 0, 255)) \ 256 \ 256

a14 = Picture2.Point(klm, brs + 2)
r14 = a14 And RGB(255, 0, 0)
g14 = (a14 And RGB(0, 255, 0)) \ 256
b14 = (a14 And RGB(0, 0, 255)) \ 256 \ 256

a15 = Picture2.Point(klm + 1, brs + 2)
r15 = a15 And RGB(255, 0, 0)
g15 = (a15 And RGB(0, 255, 0)) \ 256
```

```
  b15 = (a15 And RGB(0, 0, 255)) \ 256 \ 256

  a16 = Picture2.Point(klm + 2, brs + 2)
  r16 = a16 And RGB(255, 0, 0)
  g16 = (a16 And RGB(0, 255, 0)) \ 256
  b16 = (a16 And RGB(0, 0, 255)) \ 256 \ 256

‘Pengisian matrix W(sy) dan W(sx)
sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2
  'Perkalian W(sy) dengan matrix 4x4 = matrix h
h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
      'Perkalian matrix h dengan W(sx) = matrix m
      m00 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    'Perkalian W(sy) dengan matrix 4x4 = matrix p
     p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
     p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
     p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
     p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
      'Perkalian matrix p dengan W(sx) = matrix q
     q00 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
    'Perkalian W(sy) dengan matrix 4x4 = matrix s
          s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
          s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
          s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
          s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
         'Perkalian matrix s dengan W(sx) = matrix t
      t00 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
```

```
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m01 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q01 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t01 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m02 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q02 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t02 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
```

```
m03 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q03 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t03 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m10 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q10 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t10 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m11 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q11 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t11 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)
sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m12 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q12 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t12 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)

sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = 0
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m13 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q13 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t13 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m20 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q20 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t20 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m21 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q21 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t21 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m22 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q22 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t22 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -1
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
```

```
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m23 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q23 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t23 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m30 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q30 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t30 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = 0
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2
```

```
h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m31 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q31 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t31 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -1 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
f3 = ((sx ^ 3) - (sx ^ 2)) / 2

h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
m32 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
    p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
    p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
    p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
    p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
    q32 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
        s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
        s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
        s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
        s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
        t32 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



sy = -2 / 3
e0 = ((-(sy ^ 3)) + (2 * (sy ^ 2)) - sy) / 2
e1 = ((3 * (sy ^ 3)) - (5 * (sy ^ 2)) + 2) / 2
e2 = ((-3 * (sy ^ 3)) + (4 * (sy ^ 2)) + sy) / 2
e3 = ((sy ^ 3) - (sy ^ 2)) / 2
sx = -2 / 3
f0 = ((-(sx ^ 3)) + (2 * (sx ^ 2)) - sx) / 2
```

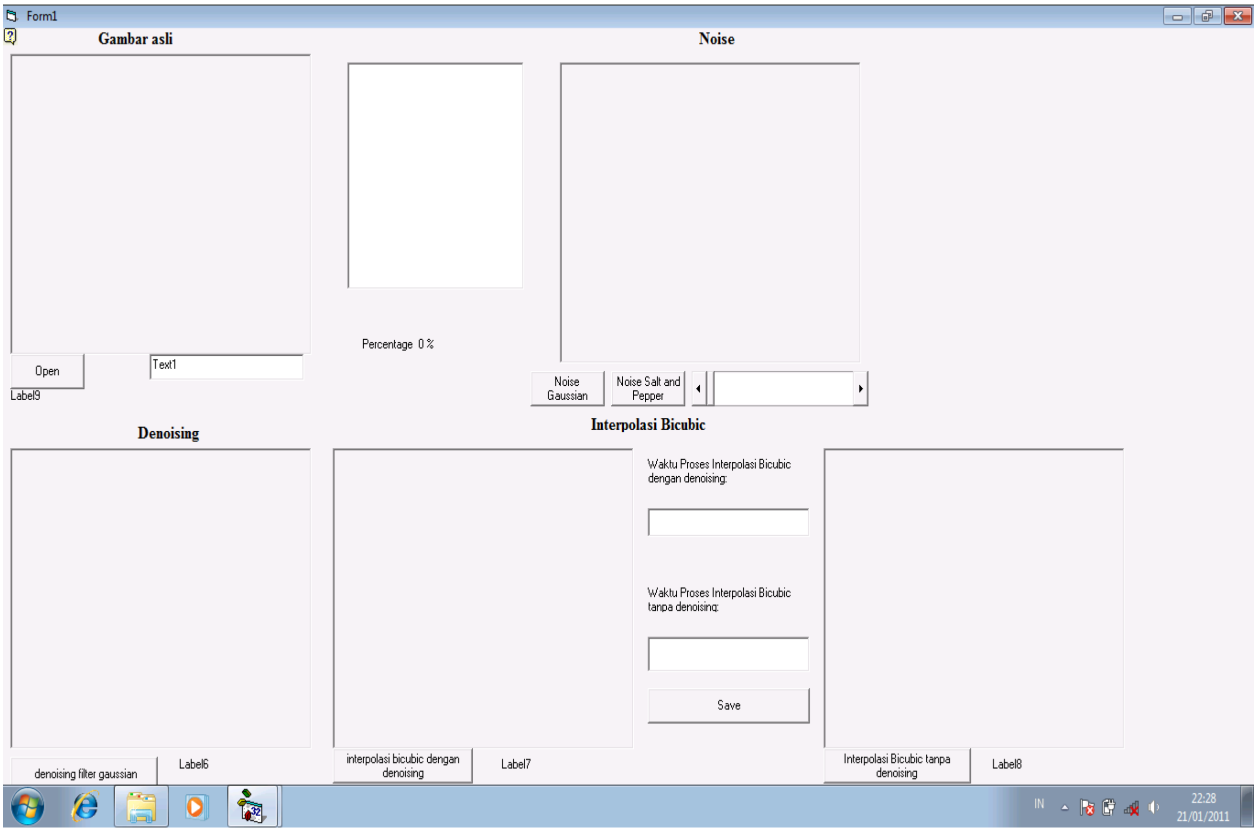
```
    f1 = ((3 * (sx ^ 3)) - (5 * (sx ^ 2)) + 2) / 2
    f2 = ((-3 * (sx ^ 3)) + (4 * (sx ^ 2)) + sx) / 2
    f3 = ((sx ^ 3) - (sx ^ 2)) / 2

    h0 = (e0 * g1 + e1 * g5 + e2 * g9 + e3 * g13)
    h1 = (e0 * g2 + e1 * g6 + e2 * g10 + e3 * g14)
    h2 = (e0 * g3 + e1 * g7 + e2 * g11 + e3 * g15)
    h3 = (e0 * g4 + e1 * g8 + e2 * g12 + e3 * g16)
    m33 = (h0 * f0 + h1 * f1 + h2 * f2 + h3 * f3)
         p0 = (e0 * r1 + e1 * r5 + e2 * r9 + e3 * r13)
         p1 = (e0 * r2 + e1 * r6 + e2 * r10 + e3 * r14)
         p2 = (e0 * r3 + e1 * r7 + e2 * r11 + e3 * r15)
         p3 = (e0 * r4 + e1 * r8 + e2 * r12 + e3 * r16)
         q33 = (p0 * f0 + p1 * f1 + p2 * f2 + p3 * f3)
               s0 = (e0 * b1 + e1 * b5 + e2 * b9 + e3 * b13)
               s1 = (e0 * b2 + e1 * b6 + e2 * b10 + e3 * b14)
               s2 = (e0 * b3 + e1 * b7 + e2 * b11 + e3 * b15)
               s3 = (e0 * b4 + e1 * b8 + e2 * b12 + e3 * b16)
               t33 = (s0 * f0 + s1 * f1 + s2 * f2 + s3 * f3)



   zr = (q00 + q01 + q02 + q03 + q10 + q11 + q12 + q13 + q20 + q21 + q22 + q23 + q30 + q31
+ q32 + q33) / 16
   zg = (m00 + m01 + m02 + m03 + m10 + m11 + m12 + m13 + m20 + m21 + m22 + m23 +
m30 + m31 + m32 + m33) / 16
   zb = (t00 + t01 + t02 + t03 + t10 + t11 + t12 + t13 + t20 + t21 + t22 + t23 + t30 + t31 + t32 +
t33) / 16

   zr = Int(zr)
   zg = Int(zg)
   zb = Int(zb)

   If zr < 0 Then zr = r6
   If zg < 0 Then zg = g6
   If zb < 0 Then zb = b6
  ' List1.AddItem (Str(klm) + "," + Str(brs) + " = " + Str(zr) + "," + Str(zg) + "," + Str(zb))
   Picture5.PSet (klm, brs), RGB(zr, zg, zb)
   Next klm
   Next brs
   selesai = Time
   beda = selesai - mulai
   Text3.Text = Format(beda, "h:m:s")
   Form1.Refresh
‘Menghitung PSNR
n = 3
hitung = 1
X = Picture5.ScaleWidth
Y = Picture5.ScaleHeight
For brs = 1 To Y Step (n + 1) * 1
   For klm = 1 To X Step (n + 1) * 1
   mseR = 0
   mseB = 0
   mseG = 0
         For m = 0 To n * 1 Step 1
```

```
        For n1 = 0 To n * 1 Step 1
        warna = Picture1.Point(klm + n1, brs + m)
         rr = warna And RGB(255, 0, 0)
         gg = Int((warna And RGB(0, 255, 0)) / 256)
         bg = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

         warna5 = Picture5.Point(klm + n1, brs + m)
         r1 = warna5 And RGB(255, 0, 0)
         g1 = Int((warna5 And RGB(0, 255, 0)) / 256)
         b1 = Int(Int((warna5 And RGB(0, 0, 255)) / 256) / 256)

         mseR = mseR + (r1 - rr) ^ 2
         mseB = mseB + (g1 - gg) ^ 2
         mseG = mseG + (b1 - bb) ^ 2

       Next n1
      Next m
    mse = (mseR + mseB + mseG) / (3 * (n * 1) * (n * 1))
    If mse = 0 Then mse = 1
    'psnr = Abs(20 * (Log(255 / (mse ^ (0.5))) / Log(10)))
    psnr = psnr + Abs(10 * (Log((255 ^ 2) / mse) / Log(10)))
    hitung = hitung + 1
  Next klm
Next brs

psnr = psnr / (hitung - 1)

Label8.Caption = "PSNR =" + Str(psnr)
'Picture5.Refresh
End Sub
'Noise Salt and Pepper
Private Sub Command7_Click()
probnoise = Val(HScroll1) / 100
For i = 1 To Picture1.ScaleWidth Step 1
 For j = 1 To Picture1.ScaleHeight Step 1
  warna = Picture1.Point(i, j)
  rr = warna And RGB(255, 0, 0)
  gg = Int((warna And RGB(0, 255, 0)) / 256)
  bb = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)
'pembangkitan noise dengan metode Salt dan Papper
  sw = Rnd
    If sw < probnoise Then
    rr = 255
    gg = 255
    bb = 255
    End If
    Picture2.PSet (i, j), RGB(rr, gg, bb)
    Picture5.PSet (i, j), RGB(rr, gg, bb)
  Next j
Next i
‘menghitung PSNR
n = 3
hitung = 1
X = Picture2.ScaleWidth
Y = Picture2.ScaleHeight
```

```
For brs = 1 To Y Step (n + 1) * 1
  For klm = 1 To X Step (n + 1) * 1
   mseR = 0
   mseB = 0
   mseG = 0
        For m = 0 To n * 1 Step 1
           For n1 = 0 To n * 1 Step 1
             warna = Picture1.Point(klm + n1, brs + m)
             rr = warna And RGB(255, 0, 0)
             gg = Int((warna And RGB(0, 255, 0)) / 256)
             bg = Int(Int((warna And RGB(0, 0, 255)) / 256) / 256)

             warna5 = Picture2.Point(klm + n1, brs + m)
             r1 = warna5 And RGB(255, 0, 0)
             g1 = Int((warna5 And RGB(0, 255, 0)) / 256)
             b1 = Int(Int((warna5 And RGB(0, 0, 255)) / 256) / 256)

             mseR = mseR + (r1 - rr) ^ 2
             mseB = mseB + (g1 - gg) ^ 2
             mseG = mseG + (b1 - bb) ^ 2

          Next n1
        Next m
     mse = (mseR + mseB + mseG) / (3 * (n * 1) * (n * 1))
     If mse = 0 Then mse = 1
     'psnr = Abs(20 * (Log(255 / (mse ^ (0.5))) / Log(10)))
     psnr = psnr + Abs(10 * (Log((255 ^ 2) / mse) / Log(10)))
     hitung = hitung + 1
   Next klm
Next brs
psnr = psnr / (hitung - 1)
Label9.Caption = "PSNR =" + Str(psnr)
End Sub
‘Menampilkan Persentase noise
Private Sub Timer1_Timer()
Label5.Caption = "Percentage " + Str(Val(HScroll1)) + " %"
End Sub
```

LAMPIRAN B

TAMPILAN VISUAL BASIC


Form1
Gambar asli
Noise
Percentage 0 %
Open
Text1
Label9
Noise Gaussian
Noise Salt and Pepper
Denoising
Interpolasi Bicubic
Waktu Proses Interpolasi Bicubic dengan denoising:
Waktu Proses Interpolasi Bicubic tanpa denoising:
Save
denoising filter gaussian
Label6
interpolasi bicubic dengan denoising
Label7
Interpolasi Bicubic tanpa denoising
Label8
IN
22:28
21/01/2011

B-1 Tampilan Pada Visual Basic

LAMPIRAN C
HASIL DATA PENGAMATAN

LAMPIRAN D

KUESIONER