Dokumentasi Mini Paint V2 Menggunakan Bahasa C#

Oleh :

Nama : Muhammad Husain Fadhlullah

NIM : 141524017

3A-D4 Teknik Informatika Politeknik Negeri Bandung

Mata Kuliah : Komputer Grafik

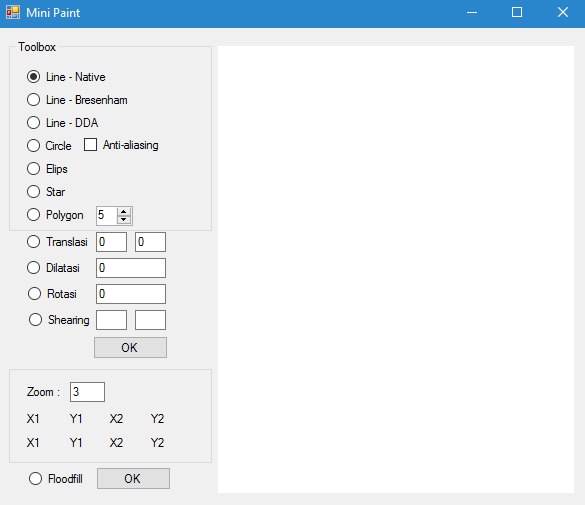
**Deskripsi Program:** Program yang dibuat adalah program seperti paint yang memiliki fitur antara lain dapat menggambar garis dengan 3 algoritma yaitu native, Bresenham, DDA; menggambar lingkaran; menggambar elips; menggambar n-bintang; menggambar n-segi, Filling menggunakan floodfill, kemudian transformasi Matrix (translasi, rotasi, dilatasi, shearing).

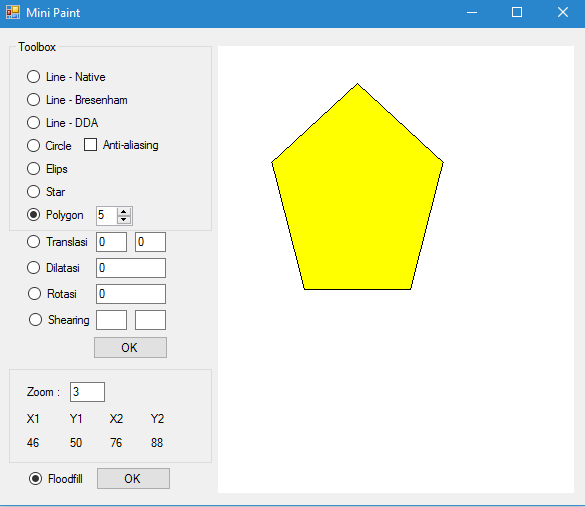
Selain fitur utama dari yang disebutkan di atas, ada juga fitur tambahan seperti zoom (semakin besar angka zoom, semakin dekat), n atau banyaknya sisi bintang dan segi.

**Bahasa:** C#

**IDE:** Visual Studio 2015

**Screen Captures:**





**Detail Modul:**

|  |  |
| --- | --- |
| No | 1 |
| Nama Modul / Fitur | Floodfill menggunakan polygon |
| Status Modul (Full, Bug, Not) | Full (dengan catatan titik mulai masih static belum bisa ditentukan secara dinamis) |
| Deskripsi Modul | Modul untuk mewarnai dengan algoritma floodfill |
| Source Code:  void FloodFill(Bitmap bitmap, int x, int y, Color color)  {  BitmapData data = bitmap.LockBits(  new Rectangle(0, 0, bitmap.Width, bitmap.Height),  ImageLockMode.ReadWrite, PixelFormat.Format32bppArgb);  int[] bits = new int[data.Stride / 4 \* data.Height];  Marshal.Copy(data.Scan0, bits, 0, bits.Length);  LinkedList<Point> check = new LinkedList<Point>();  int floodTo = color.ToArgb();  int floodFrom = bits[x + y \* data.Stride / 4];  bits[x + y \* data.Stride / 4] = floodTo;  if (floodFrom != floodTo)  {  check.AddLast(new Point(x, y));  while (check.Count > 0)  {  Point cur = check.First.Value;  check.RemoveFirst();  foreach (Point off in new Point[] {  new Point(0, -1), new Point(0, 1),  new Point(-1, 0), new Point(1, 0)})  {  Point next = new Point(cur.X + off.X, cur.Y + off.Y);  if (next.X >= 0 && next.Y >= 0 &&  next.X < data.Width &&  next.Y < data.Height)  {  if (bits[next.X + next.Y \* data.Stride / 4] == floodFrom)  {  check.AddLast(next);  bits[next.X + next.Y \* data.Stride / 4] = floodTo;  }  }  }  }  }  Marshal.Copy(bits, 0, data.Scan0, bits.Length);  bitmap.UnlockBits(data);  }  Digunakan di polygon:  using (Bitmap bitmap = new Bitmap(1000, 1000))  {  using (Graphics g = Graphics.FromImage(bitmap))  {  int num\_points = Convert.ToInt16(nud\_polygon.Value);  PointF[] pts = new PointF[num\_points];  Pen myPen = new Pen(Color.Black);  Color color = Color.Black;  double rx = getRadiusX(start\_point.X \* zoom, end\_point.X \* zoom);  double ry = getRadiusY(start\_point.Y \* zoom, end\_point.Y \* zoom);  double cx = start\_point.X \* zoom;  double cy = start\_point.Y \* zoom;  double alpha = -Math.PI / 2; //starting point  double theta = 2 \* Math.PI / num\_points;  //rotasi  //Rumus SMA :  //Pi = O + R \* (d / R) ^ mod[i, 2] \* (cos(2 π / n + π / n \* i),sin(2 π / n + π / n \* i)); i = 0,1,….2n - 1  for (int i = 0; i < num\_points; i++)  {  pts[i] = new PointF(  (float)(cx + rx \* Math.Cos(alpha)),  (float)(cy + ry \* Math.Sin(alpha)));  alpha += theta;  }  g.DrawPolygon(myPen, pts);  }  FloodFill(bitmap, 200, 200, Color.Yellow);  e.Graphics.DrawImage(bitmap, 0, 0);  } | |

|  |  |
| --- | --- |
| No | 2 |
| Nama Modul / Fitur | Transformasi matrix |
| Status Modul (Full, Bug, Not) | Bug (baru bisa translasi, rotasi, dilatasi dan baru diimplementasi pada algoritma dda saja – belum digunakan untuk modul matrix2nya) |
| Deskripsi Modul | Modul transformasi matrix |
| Source Code:  private int[,] Translasi(int x, int y)  {  int[,] a1 = new int[3, 3];  a1[0, 0] = 1;  a1[0, 1] = 0;  a1[0, 2] = Convert.ToInt16(tb\_translasi\_x.Text);  a1[1, 0] = 0;  a1[1, 1] = 1;  a1[1, 2] = Convert.ToInt16(tb\_translasi\_y.Text);  a1[2, 0] = 0;  a1[2, 1] = 0;  a1[2, 2] = 1;  int[,] a2 = new int[3, 1];  a2[0, 0] = x;  a2[1, 0] = y;  a2[2, 0] = 1;  int[,] a3 = new int[3, 1];  a3[0, 0] = a1[0, 0] \* a2[0, 0] + a1[0, 1] \* a2[1, 0] + a1[0, 2] \* a2[2, 0];  a3[1, 0] = a1[1, 0] \* a2[0, 0] + a1[1, 1] \* a2[1, 0] + a1[1, 2] \* a2[2, 0];  a3[2, 0] = 1;    return a3;  }  private int[,] Dilatasi(int x, int y)  {  int[,] a1 = new int[3, 3];  a1[0, 0] = Convert.ToInt16(tb\_dilatasi.Text);  a1[0, 1] = 0;  a1[0, 2] = 0;  a1[1, 0] = Convert.ToInt16(tb\_dilatasi.Text);  a1[1, 1] = 0;  a1[1, 2] = 0;  a1[2, 0] = 0;  a1[2, 1] = 0;  a1[2, 2] = 1;  int[,] a2 = new int[3, 1];  a2[0, 0] = x;  a2[1, 0] = y;  a2[2, 0] = 1;  int[,] a3 = new int[3, 1];  a3[0, 0] = a1[0, 0] \* a2[0, 0] + a1[0, 1] \* a2[1, 0] + a1[0, 2] \* a2[2, 0];  a3[1, 0] = a1[1, 0] \* a2[0, 0] + a1[1, 1] \* a2[1, 0] + a1[1, 2] \* a2[2, 0];  a3[2, 0] = 1;  return a3;  }  private int[,] Rotasi(int x, int y)  {  int[,] a1 = new int[3, 3];  a1[0, 0] = (int)Math.Cos(Math.PI/Convert.ToDouble(tb\_rotasi.Text));  a1[0, 1] = (int)Math.Sin(Math.PI / Convert.ToDouble(tb\_rotasi.Text));  a1[0, 2] = 0;  a1[1, 0] = -(int)Math.Sin(Math.PI / Convert.ToDouble(tb\_rotasi.Text));  a1[1, 1] = (int)Math.Cos(Math.PI / Convert.ToDouble(tb\_rotasi.Text));  a1[1, 2] = 0;  a1[2, 0] = 0;  a1[2, 1] = 0;  a1[2, 2] = 1;  int[,] a2 = new int[3, 1];  a2[0, 0] = x;  a2[1, 0] = y;  a2[2, 0] = 1;  int[,] a3 = new int[3, 1];  a3[0, 0] = a1[0, 0] \* a2[0, 0] + a1[0, 1] \* a2[1, 0] + a1[0, 2] \* a2[2, 0];  a3[1, 0] = a1[1, 0] \* a2[0, 0] + a1[1, 1] \* a2[1, 0] + a1[1, 2] \* a2[2, 0];  a3[2, 0] = 1;  return a3;  }  private int[,] Shearing(int x, int y)  {  int[,] a1 = new int[3, 3];  a1[0, 0] = 1;  a1[0, 1] = Convert.ToInt16(tb\_shearing\_x.Text);  a1[0, 2] = 0;  a1[1, 0] = Convert.ToInt16(tb\_shearing\_y.Text);  a1[1, 1] = 1;  a1[1, 2] = 0;  a1[2, 0] = 0;  a1[2, 1] = 0;  a1[2, 2] = 1;  int[,] a2 = new int[3, 1];  a2[0, 0] = x;  a2[1, 0] = y;  a2[2, 0] = 1;  int[,] a3 = new int[3, 1];  a3[0, 0] = a1[0, 0] \* a2[0, 0] + a1[0, 1] \* a2[1, 0] + a1[0, 2] \* a2[2, 0];  a3[1, 0] = a1[1, 0] \* a2[0, 0] + a1[1, 1] \* a2[1, 0] + a1[1, 2] \* a2[2, 0];  a3[2, 0] = 1;  return a3;  }  Berikut implementasinya:  if (rb\_rotasi.Checked)  {  xI = Dx / r;  yI = Dy / r;  double rotasi = -Convert.ToDouble(tb\_rotasi.Text) / 180 \* Math.PI;  for (k = 0; k < r; k++)  {  x += xI;  y += yI;  x1 += xI \* Math.Cos(rotasi) - yI \* Math.Sin(rotasi);  y1 += xI \* Math.Sin(rotasi) + yI \* Math.Cos(rotasi);  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x \* zoom), (int)(y \* zoom), 4, 4));  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x1 \* zoom), (int)(y1 \* zoom), 4, 4));  }  }  if (rb\_translasi.Checked)  {  xI = Dx / r;  yI = Dy / r;  int new\_x = Convert.ToInt16(tb\_translasi\_x.Text);  int new\_y = Convert.ToInt16(tb\_translasi\_y.Text);  x1 = new\_x + start\_point.X;  y1 = new\_x + start\_point.Y;  for (k = 0; k < r; k++)  {  x += xI;  y += yI;  x1 += xI;  y1 += yI;  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x \* zoom), (int)(y \* zoom), 4, 4));  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x1 \* zoom), (int)(y1 \* zoom), 4, 4));  }  }  if (rb\_dilatasi.Checked)  {  xI = Dx / r;  yI = Dy / r;  for (k = 0; k < r \* Convert.ToInt16(tb\_dilatasi.Text); k++)  {  x += xI;  y += yI;  x1 += xI;  y1 += yI;  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x \* zoom), (int)(y \* zoom), 4, 4));  e.Graphics.FillRectangle(new SolidBrush(color), new Rectangle((int)(x1 \* zoom), (int)(y1 \* zoom), 4, 4));  }  } | |