


Lampiran 1 Daftar Riwayat Hidup



Dwigita Octasya Fajry

Teknik Informatika

BRIEFLY ABOUT ME

Kelebihan saya adalah saya dapat beradaptasi dengan cepat, dapat bekerja secara team, bertanggung jawab, menyukai hal baru, disiplin, jujur, dan sopan.

PENDIDIKAN

SMPIT ALFATIH 1

2009 to 2012

SMAN 3 KAB. TANGERANG

2012 to 2015

UNIV. ESA UNGGUL

2015 to 2019

PELATIHAN SERTIFIKAT

TOEFL SCORE 490

SAP 1

SAP 2

SKILLS

HTML

75%

SQL

60%


UI DESIGNER

50%


JAVA

25%


INTEREST




PROGRAM MING




INTERNET



WRITING



email:
gitaoctasyaa@gmail.com



+62 878 8535 4214

LAMPIRAN 2 CODING*Loading splash*

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace pemantauan
{
    public partial class LoadingSplash : Form
    {
        public LoadingSplash()
        {
            InitializeComponent();
        }

        private void Timer1_Tick(object sender, EventArgs e)
        {
            try
            {
                rectangleShape2.Width += 1;
                if (rectangleShape2.Width >= 492)
                {
                    timer1.Stop();
                    FormLogin frm = new FormLogin();
                    frm.Show();
                    this.Hide();
                }
            }
            catch (Exception)
            {
                return;
            }
        }
    }
}

```

Form login

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using MySql.Data.MySqlClient;

```

```

namespace pemantauan
{
    public partial class FormLogin : Form
    {
        MySqlConnection conn = ConnectionService.GetConnection();
        public FormLogin()
        {
            InitializeComponent();
        }

        private void Button1_Click(object sender, EventArgs e)
        {
            if (formlogin(textBox1.Text, textBox2.Text))
            {
                Utama outama = new Utama();
                outama.ShowDialog();
            }
            else
            {
                MessageBox.Show("Login gagal");
            }
        }
        private Boolean formlogin(string sUsername, string sPassword)
        {
            string SQL = "SELECT username,password FROM tb_user";
            conn.Open();
            MySqlCommand cmd = new MySqlCommand(SQL, conn);
            MySqlDataReader reader = cmd.ExecuteReader();
            while (reader.Read())
            {
                if ((sUsername == reader.GetString(0)) && (sPassword ==
reader.GetString(1)))
                {
                    conn.Close();
                    return true;
                }
            }
            conn.Close();
            return false;
        }
    }
}

```

Pemantauan

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using AForge.Imaging;
using AForge.Imaging.Filters;
using AForge;

```

```

using AForge.Video.DirectShow;
using System.Drawing.Imaging;
using System.Threading;
using System.IO;

namespace pemantauan
{
    public partial class utama : Form
    {
        string d = "";
        private FilterInfoCollection videoDevices;
        EuclideanColorFiltering filter = new EuclideanColorFiltering();
        Color color = Color.Black;
        GrayscaleBT709 grayscaleFilter = new GrayscaleBT709();
        BlobCounter blobCounter = new BlobCounter();
        int range = 120;
        public f21()
        {
            InitializeComponent();

            blobCounter.MinWidth = 2;
            blobCounter.MinHeight = 2;
            blobCounter.FilterBlobs = true;
            blobCounter.ObjectsOrder = ObjectsOrder.Size;
            try
            {
                // enumerate video devices
                videoDevices = new
FilterInfoCollection(FilterCategory.VideoInputDevice);

                if (videoDevices.Count == 0)
                    throw new ApplicationException();

                // add all devices to combo
                foreach (FilterInfo device in videoDevices)
                {
                    camerasCombo.Items.Add(device.Name);
                }
                camerasCombo.SelectedIndex = 0;
            }
            catch (ApplicationException)
            {
                camerasCombo.Items.Add("No local capture devices");
                videoDevices = null;
            }

            Bitmap b = new Bitmap(320, 240);
            // Rectangle a = (Rectangle)r;
            Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
            Graphics g2 = Graphics.FromImage(b);
            pen1 = new Pen(Color.FromArgb(255, 0, 0), 3);
            g2.Clear(Color.White);
            g2.DrawLine(pen1, b.Width / 2, 0, b.Width / 2, b.Width);
            g2.DrawLine(pen1, b.Width, b.Height / 2, 0, b.Height / 2);
            pictureBox1.Image = (System.Drawing.Image)b;
        }

        private void timer1_Tick(object sender, EventArgs e)
    }
}

```

```

    {
    }
}

private void videoSourcePlayer1_NewFrame(object sender, ref Bitmap
image)
{
    Bitmap objectsImage = null;
    Bitmap mImage = null;
    mImage=(Bitmap)image.Clone();
    filter.CenterColor = Color.FromArgb(color.ToArgb());
    filter.Radius =(short)range;

    objectsImage = image;
    filter.ApplyInPlace(objectsImage);

    BitmapData objectsData = objectsImage.LockBits(new Rectangle(0,
0, image.Width, image.Height),
    ImageLockMode.ReadOnly, image.PixelFormat);
    UnmanagedImage grayImage = grayscaleFilter.Apply(new
UnmanagedImage(objectsData));
    objectsImage.UnlockBits(objectsData);

    blobCounter.ProcessImage(grayImage);
    Rectangle[] rects = blobCounter.GetObjectRectangles();

    if (rects.Length > 0)
    {
        foreach (Rectangle objectRect in rects)
        {
            Graphics g = Graphics.FromImage(mImage);
            using (Pen pen = new Pen(Color.FromArgb(160, 255, 160),
5))
            {
                g.DrawRectangle(pen, objectRect);
            }
            g.Dispose();
        }
    }

    image = mImage;
}

private void videoSourcePlayer3_NewFrame(object sender, ref Bitmap
image)
{
    Bitmap objectsImage = null;

    // set center color and radius
    filter.CenterColor = Color.FromArgb(color.ToArgb());
    filter.Radius = (short)range;
    // apply the filter
    objectsImage = image;
    filter.ApplyInPlace(image);

    // lock image for further processing

```

```

        BitmapData objectsData = objectsImage.LockBits(new Rectangle(0,
0, image.Width, image.Height),
        ImageLockMode.ReadOnly, image.PixelFormat);

        // grayscaling
        UnmanagedImage grayImage = grayscaleFilter.Apply(new
UnmanagedImage(objectsData));

        // unlock image
        objectsImage.UnlockBits(objectsData);

        // locate blobs
        blobCounter.ProcessImage(grayImage);
        Rectangle[] rects = blobCounter.GetObjectRectangles();

        if (rects.Length > 0)
        {
            Rectangle objectRect = rects[0];

            // draw rectangle around detected object
            Graphics g = Graphics.FromImage(image);

            using (Pen pen = new Pen(Color.FromArgb(160, 255, 160), 5))
            {
                g.DrawRectangle(pen, objectRect);
            }
            g.Dispose();
            int objectX = objectRect.X + objectRect.Width / 2 -
image.Width / 2;
            int objectY = image.Height / 2 - (objectRect.Y +
objectRect.Height / 2);
            ParameterizedThreadStart t = new
ParameterizedThreadStart(p);
            Thread aa = new Thread(t);
            aa.Start(rects[0]);
        }
        Graphics g1 = Graphics.FromImage(image);
        Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
        g1.DrawLine(pen1, image.Width/2, 0, image.Width/2, image.Width);
        g1.DrawLine(pen1, image.Width, image.Height / 2, 0,
image.Height / 2);
        g1.Dispose();
    }

    void p(object r)
    {
        try
        {
            Bitmap b = new Bitmap(pictureBox1.Image);
            Rectangle a = (Rectangle)r;
            Pen pen1 = new Pen(Color.FromArgb(160, 255, 160), 3);
            Graphics g2 = Graphics.FromImage(b);
            pen1 = new Pen(color, 3);
            // Brush b5 = null;
            SolidBrush b5 = new SolidBrush(color);
            // g2.Clear(Color.Black);
        }
    }

```

```

Font f = new Font(Font, FontStyle.Bold);

g2.DrawString("o", f, b5, a.Location);
g2.Dispose();
pictureBox1.Image = (System.Drawing.Image)b;
this.Invoke((MethodInvoker)delegate
{
    richTextBox1.Text = a.Location.ToString() + "\n" +
richTextBox1.Text + "\n"; ;
});
}
catch (Exception faa)
{
    Thread.CurrentThread.Abort();
}

Thread.CurrentThread.Abort();
}

private void button1_Click(object sender, EventArgs e)
{
    videoSourcePlayer1.SignalToStop();
    videoSourcePlayer1.WaitForStop();
    videoSourcePlayer2.SignalToStop();
    videoSourcePlayer2.WaitForStop();
    videoSourcePlayer3.SignalToStop();
    videoSourcePlayer3.WaitForStop();
    // videoDevices = null;
    VideoCaptureDevice videoSource = new
VideoCaptureDevice(videoDevices[camerasCombo.SelectedIndex].MonikerString);
    videoSource.DesiredFrameSize = new Size(320, 240);
    videoSource.DesiredFrameRate = 12;

    videoSourcePlayer1.VideoSource = videoSource;
    videoSourcePlayer1.Start();
    videoSourcePlayer2.VideoSource = videoSource;
    videoSourcePlayer2.Start();
    videoSourcePlayer3.VideoSource = videoSource;
    videoSourcePlayer3.Start();
    //groupBox1.Enabled = false;
}

private void f21_FormClosing(object sender, FormClosingEventArgs e)
{
    videoSourcePlayer1.SignalToStop();
    videoSourcePlayer1.WaitForStop();
    videoSourcePlayer2.SignalToStop();
    videoSourcePlayer2.WaitForStop();
    videoSourcePlayer3.SignalToStop();
    videoSourcePlayer3.WaitForStop();
    groupBox1.Enabled = true;
}

private void button2_Click(object sender, EventArgs e)
{

```

```

        videoSourcePlayer1.SignalToStop();
        videoSourcePlayer1.WaitForStop();
        videoSourcePlayer2.SignalToStop();
        videoSourcePlayer2.WaitForStop();
        videoSourcePlayer3.SignalToStop();
        videoSourcePlayer3.WaitForStop();
    }

    private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
    {
    }

    private void button3_Click(object sender, EventArgs e)
    {
        colorDialog1.ShowDialog();
        color = colorDialog1.Color;
    }

e) private void numericUpDown1_ValueChanged(object sender, EventArgs
    {
        range = Convert.ToInt32(numericUpDown1.Value) ;
    }

e) private void numericUpDown2_ValueChanged(object sender, EventArgs
    {
        blobCounter.MaxWidth = Convert.ToInt32(numericUpDown2.Value);
    }

e) private void numericUpDown3_ValueChanged(object sender, EventArgs
    {
        blobCounter.MinWidth = Convert.ToInt32(numericUpDown3.Value);
    }
}
}

```