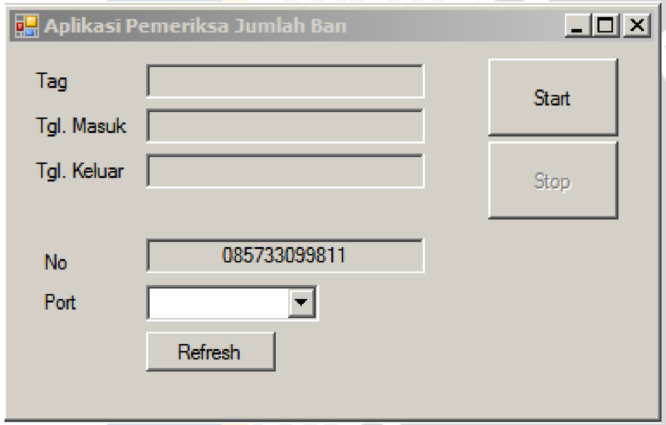
LAMPIRAN III

Program Visual

Tampilan Visual



No 085733099811

Listing Program

```
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 System.IO.Ports;
using System.Data.OleDb;
using System.Xml.Serialization;
using System.Threading;
namespace BanSkripsi
{
    public partial class Form1 : Form
    {
        SerialPort port = new SerialPort();
        delegate void setTextCallback(string Text);
        delegate void setTCallback();
        string rxString;
        string tglMsk;
        string tglKlr;

        public Form1()
        {
            InitializeComponent();
            refreshList();
            int jumlah = hitungBarang();
            buttonStop.Enabled = false;
        }
        private void buttonStart_Click(object sender, EventArgs e)
        {
            port.BaudRate = 9600;
            port.PortName = cbPort.SelectedItem.ToString();
            port.Open();
              port.DataReceived += new
              SerialDataReceivedEventHandler(serialPort1_DataReceived);
            if (port.IsOpen)
            {
                buttonStart.Enabled = false;
                buttonStop.Enabled = true;
            }
            else
            {
                MessageBox.Show("Gagal melakukan koneksi");
            }
        }
```

```
private void buttonStop_Click(object sender, EventArgs e)
{
    port.Close();
    buttonStart.Enabled = true;
    buttonStop.Enabled = false;
}
 private void refreshList()
{
    cbPort.Items.Clear();
    foreach (string name in SerialPort.GetPortNames())
    {
        cbPort.Items.Add(name);
    }
}

private void buttonRefresh_Click(object sender, EventArgs e)
{
    refreshList();
}
private void serialPort1_DataReceived
    (object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
    rxString = "";
    char[] ok = new char[12];
    Thread.Sleep(100);
    port.Read(ok, 0, 12);
    for (int i = 0; i < 12; i++)
    {
        rxString += ok[i];
    }
    port.DiscardInBuffer();
    dataHandler(rxString);
```

```
        }
        private void dataHandler(string data)
        {
            if (data[0] == 'c' && data[1] == 'h' && data[2] == 'e' && data[3] == 'c'
&& data[4] == 'k')
            {
                int jumlah = hitungBarang();
                port.Write(jumlah+" ");
                port.DiscardOutBuffer();
                port.Close();
                port.Open();
            }
            if (data[1] == '\0' || data[0] != '2')
            {
                port.Write("error");
                port.DiscardOutBuffer();
            }
            else
            {
                int check = checkTag(data);
                if (check == 0)
                {
                    char[] ok = {'0'};
                    tagKeluar(data);
                    setStatus("Keluar");
                    setTag(data);
                    setMasuk("");
                    setKeluar("");
                    port.Write(ok, 0, 1);
                    port.DiscardOutBuffer();
                }
                else if (check == 1)
        {
            char[] ok = { '1' };
            dbInsert(data);
            setStatus("Masuk");
            setTag(data);
            setMasuk("");
            setKeluar("");
            port.Write(ok,0,1);
            port.DiscardOutBuffer();
        }
        else if (check == 2)
        {
            char[] ok = { '1' };
            dbInsert(data);
            setStatus("Masuk");
            setTag(data);
            setMasuk("");
            setKeluar("");
            port.Write(ok,0,1);
            port.DiscardOutBuffer();
        }
        else
        {
            setTag(data);
            setMasuk("");
            setKeluar("");
            lblStatus.Text = "Error";
            port.DiscardOutBuffer();
        }
    }
}
private void setStatus(string text)
{
    if (this.lblStatus.InvokeRequired)
    {
        setTextCallback d = new setTextCallback(setStatus);
        this.Invoke(d, new object[] { text });
    }
    else
    {
        this.lblStatus.Text = text;
    }
}
private void setTag(string text)
{
    if (this.lblStatus.InvokeRequired)
    {
        setTextCallback k = new setTextCallback(setTag);
        this.Invoke(k, new object[] { text });
    }
    else
    {
        this.tbTag.Text = text;
    }
}
private void setMasuk(string text)
{
    text = tglMsk;
    if (this.lblStatus.InvokeRequired)
    {
        setTextCallback d = new setTextCallback(setMasuk);
        this.Invoke(d, new object[] { text });
    }
    else
    {
        this.tbMasuk.Text = text;
        }
    }
    private void setKeluar(string text)
    {
        text = tglKlr;
        if (this.lblStatus.InvokeRequired)
        {
            setTextCallback d = new setTextCallback(setKeluar);
            this.Invoke(d, new object[] { text });
        }
        else
        {
            this.tbKeluar.Text = text;
        }
    }
    private static OleDbConnection GetConnection()
    {
        OleDbConnection conn = new OleDbConnection();
        string db = "";
        try
        {
            db = @"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=D:\Database\db_tag.mdb;Mode=ReadWrite|Share Deny None;Persist
Security Info=False";
            conn = new OleDbConnection(db);
            conn.Open();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
        return conn;
    }
```

```
    private void closeConn(OleDbConnection conn)
    {
        try
        {
            conn.Close();
        }
        catch (Exception ex)
        {
            Console.WriteLine(ex.Message);
        }
    }
    private List<tag> dbFind(string tagNumber)
    {
        List<tag> tagList = new List<tag>();
        DataSet ds = new DataSet();
        DataTable dt = new DataTable();
        OleDbConnection conn = GetConnection();
        OleDbDataAdapter da = new OleDbDataAdapter("SELECT * FROM
Tb_Tag where tag = '" + tagNumber + "'", conn);
        da.Fill(ds);
        conn.Close();
        dt = ds.Tables[0];
        foreach (DataRow item in dt.Rows)
        {
            tag ok = new tag();
            ok.Tag = item[1].ToString();
            ok.tglMasuk = item[2].ToString();
            ok.Tglkeluar = item[3].ToString();

            tagList.Add(ok);
        }
        return tagList;
    }
```

```
        private int dbInsert(string tagNumber)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand dc = new OleDbCommand();
            DateTime date = new DateTime();
            date = DateTime.Now;
            string tglNow = date.Day + "-" + date.Month + "-" + date.Year;
            dc.Connection = conn;
            dc.CommandText = "insert into Tb_Tag (Tag, tglMasuk) values ('" +
tagNumber + "','" + tglNow + "')";
            dc.ExecuteNonQuery();
            conn.Close();

            return 0; //success
        }

        private int tagKeluar(string tagNumber)
        {
            OleDbConnection conn = GetConnection();
            OleDbCommand dc = new OleDbCommand();
            DateTime date = new DateTime();
            date = DateTime.Now;
            string tglNow = date.Day + "-" + date.Month + "-" + date.Year;
            dc.Connection = conn;
            dc.CommandText = "Update tb_tag set Tglkeluar = '" + tglNow + "'
where tag = '" + tagNumber + "'";
            dc.ExecuteNonQuery();
            conn.Close();
            return 0;
        }
        private int checkTag(string tagNumber)
        {
```

```
            DateTime date = new DateTime();
            date = DateTime.Now;
            string tglNow = date.Day + "-" + date.Month + "-" + date.Year;
            // Tag ada belum keluar             = 0
            // Tag ada sudah keluar             = 1
            // Tidak ada                        = 2
            List<tag> tagList = dbFind(tagNumber);
            List<tag> tmpList = new List<tag>();
            var check = from i in tagList where i.Tglkeluar == "" select i;
            foreach (var item in check)
            {
                tmpList.Add(item);
            }
            if (tmpList.Count > 0)
            {
                tglMsk = tmpList[tmpList.Count-1].tglMasuk;
                tglKlr = tglNow;
                return 0;
            }
            check = from i in tagList select i;
            tmpList.Clear();
            foreach (var item in check)
            {
                tmpList.Add(item);
            }
            if (tmpList.Count != 0)
            {
                tglMsk = tglNow;
                tglKlr = "-";
                return 1;
            }
            tglMsk = "";
            tglKlr = "";
            return 2;
        }
        private int hitungBarang()
        {
            List<tag> tagList = new List<tag>();

            DataSet ds = new DataSet();
            DataTable dt = new DataTable();
            OleDbConnection conn = GetConnection();
            OleDbDataAdapter da = new OleDbDataAdapter("SELECT * FROM
Tb_Tag", conn);

            da.Fill(ds);
            conn.Close();
            dt = ds.Tables[0];
            foreach (DataRow item in dt.Rows)
            {
                tag ok = new tag();
                ok.Tag = item[1].ToString();
                ok.tglMasuk = item[2].ToString();
                ok.Tglkeluar = item[3].ToString();

                tagList.Add(ok);
            }

            List<tag> tmpList = new List<tag>();
            var check = from i in tagList where i.Tglkeluar == "" select i;
            foreach (var item in check)
            {
                tmpList.Add(item);
            }

            return tmpList.Count;
```

```
        }
    }
}
```