

## **PROYEK AKHIR SEMESTER GASAL**



NAMA : Fajri Riqza M  
KELAS : X PPLG 1  
NIS : 258713  
JUDUL PROYEK : APLIKASI PEMBELIAN TIKET

PENGEMBANGAN PERANGKAT LUNAK DAN GIM

**SMK NEGERI 1 KANDEMAN**

TAHUN PELEJARAN 2025/2026

## DESKRIPSI PROYEK

Proyek ini bertujuan untuk membuat **aplikasi kasir sederhana** yang digunakan pada pembelian tiket roller coaster. Aplikasi ini membantu proses transaksi penjualan agar lebih cepat, akurat, dan efisien dibandingkan pencatatan manual.

Melalui aplikasi ini, pengguna (kasir) dapat:

1. Memasukkan data pesanan pelanggan (Nama, Umur , dan Tinggi Badan).
2. Memilih kategori tiket berdasarkan kelompok usia (Anak, Remaja, atau Dewasa) dengan sistem checkbox yang memungkinkan pembelian multi-kategori dalam satu transaksi..
3. Memasukkan jumlah tiket yang akan dibeli.
4. Menghitung total harga secara otomatis berdasarkan kategori dan jumlah tiket yang dipilih.
5. Menghitung kembalian uang secara otomatis setelah input pembayaran.
6. Memberikan diskon otomatis 10% untuk pembelian 5 tiket atau lebih.
7. Menampilkan daftar pembeli yang sudah melakukan pembelian.
8. Melakukan validasi otomatis untuk persyaratan keselamatan seperti tinggi badan minimal 155 cm, maksimal 200 cm, dan umur minimal 14 tahun.

Proyek ini mengimplementasikan **konsep dasar pemrograman** seperti:

1. **Variabel dan Tipe Data** untuk menyimpan informasi pembeli (string untuk nama, int untuk umur dan jumlah tiket, decimal untuk perhitungan harga).
2. **Operator dan Percabangan (if, else)** untuk validasi input (nama, umur, tinggi badan), menentukan harga berdasarkan kategori checkbox, menghitung diskon, dan menentukan kembalian.
3. **Perulangan (for loop)** untuk menampilkan daftar pembeli yang tersimpan di ListBox dengan nomor urut otomatis.
4. **Fungsi/Method** untuk memisahkan logika validasi, perhitungan harga, dan menampilkan data pembeli.
5. **GUI (Graphical User Interface)** untuk antarmuka kasir yang mudah digunakan dengan komponen seperti TextBox, CheckBox, Button, dan ListBox.
6. **Validasi Input Lengkap** untuk memastikan data yang dimasukkan benar, lengkap, dan memenuhi persyaratan keselamatan wahana (validasi nama tidak boleh kosong, umur minimal 14 tahun, tinggi badan minimal 155 cm,dan maksimal 200 cm).
7. **Form Login** untuk login ke form transaksi dan harus masukan username,password.

## RINGKASAN TEORI C# YANG DIGUNAKAN

### 1. Variabel dan Tipe Data

Variabel digunakan untuk menyimpan data yang akan digunakan dalam program.

Contoh tipe data yang sering digunakan dalam aplikasi kasir:

```
string nama = "BUDI SANTOSO";
```

```
int umur = 29;
```

```
int tinggi = 170;
```

```
int jumlah = 6;
```

```
int harga = 100000;
```

```
int total = jumlah * harga;
```

❖ string → menyimpan teks (nama pembeli, kategori tiket).

❖ int → menyimpan bilangan bulat (umur, tinggi badan, jumlah tiket, harga).

❖ bool → menyimpan nilai true/false (status checkbox).

### 2. Operator

Operator digunakan untuk melakukan perhitungan atau operasi logika.

Contoh:

```
total = harga * jumlah;
```

```
diskon = total * 10 / 100;
```

```
kembalian = uangBayar - total;
```

Jenis operator yang digunakan:

❖ **Aritmatika:** +, -, \*, /, %

❖ **Perbandingan:** ==, !=, >, <, >=, <=

❖ **Logika:** &&, ||, !

❖ **Assignment:** =, +=,

### 3. Percabangan (Kondisi)

Percabangan digunakan untuk menentukan alur program berdasarkan kondisi tertentu.

Contoh:

```
// Validasi uang bayar
```

```
if (!int.TryParse(inputBayar, out int uangBayar))
```

```
{
```

```

    MessageBox.Show("Nominal tidak valid!", "Pembayaran");
    return;
}

// Cek cukup atau tidak
if (uangBayar < total)
{
    MessageBox.Show($" ✕ Nahkan Uangnya kurang!\nKurang: Rp {(total -
uangBayar):N0}");
    return;
}

```

#### 4. Perulangan (Looping)

Digunakan untuk mengulang proses tertentu seperti menampilkan daftar pembeli.

Contoh:

```

// FOR LOOP untuk tampilkan daftar
for (int i = 0; i < listPembeli.Items.Count; i++)
{
    daftar += $"{i + 1}. {listPembeli.Items[i]}\n";
}

daftar += $"
=====
";

daftar += $"
Total Pembeli: {listPembeli.Items.Count}";
MessageBox.Show(daftar, "Rincian Pembeli");
}

```

#### 5. Perhitungan Multi-Kategori

Aplikasi ini memungkinkan pembelian beberapa kategori tiket sekaligus dengan perhitungan otomatis.

Contoh:

```

// Hitung semua kategori yang dicentang
if (chkAnak.Checked) // Checked itu property bool (true/false)
{
    int hargaAnak = 50000 * jumlah;
}

```

```

        totalSemua += hargaAnak;
        kategoriDipilih += "Anak ";
        jumlahKategori++;
    }
    if (chkRemaja.Checked)
    {
        int hargaRemaja = 75000 * jumlah;
        totalSemua += hargaRemaja;
        kategoriDipilih += "Remaja ";
        jumlahKategori++;
    }
    if (chkDewasa.Checked)
    {
        int hargaDewasa = 100000 * jumlah;
        totalSemua += hargaDewasa;
        kategoriDipilih += "Dewasa ";
        jumlahKategori++;
    }

    int totalTiket = jumlah * jumlahKategori;
    int total = totalSemua;

```

## 6. Input dan Output

Input digunakan untuk menerima data dari pengguna, output menampilkan hasil.

Contoh:

```

string nama = txtNama.Text;
int umur = int.Parse(txtUmur.Text);
int tinggi = (int)numTinggiBadan.Value;
int jumlah = int.Parse(txtJumlahTiket.Text);
string kategoriDipilih = "";
// Input Uang Bayar
string inputBayar = Microsoft.VisualBasic.Interaction.InputBox (\nMasukkan uang
bayar:\n\n\n ✖ Ingat jangan utang yaa!!!", "Pembayaran", "");
// Tambah ke list Pembeli / Outputnya

```

```
listPembeli.Items.Add($"{txtNama.Text} - {umur} th - {tinggi} cm - {kategoriDipilih} - {totalTiket} tiket - Rp {total:N0} - Rp {kembalian} ");
```

## 7. GUI (Graphical User Interface) *(Jika menggunakan Visual Studio Windows App)*

Digunakan untuk membuat tampilan aplikasi kasir yang lebih interaktif.

Komponen yang sering digunakan:

- **Label** → menampilkan teks seperti "Masukan Nama", "Umur", "Total Tiket".
- **TextBox** → untuk memasukkan data nama, umur, jumlah tiket.
- **NumericUpDown** → untuk input tinggi badan dengan kontrol up/down.
- **CheckBox** → untuk memilih kategori tiket (bisa lebih dari satu).
- **Button** → untuk mengeksekusi perintah seperti *Pesan*, *Tampilkan Daftar*, dan *Keluar*.
- **ListBox** → menampilkan daftar pembeli dengan detail transaksi.

## PERANCANGAN PROGRAM

### Rancangan Tampilan Pembelian Tiket Roller Coaster

Komponen GUI yang digunakan:

1. **Label** - untuk judul dan keterangan.
2. **TextBox** - untuk input Nama, Umur, Jumlah Tiket, Username, Password.
3. **NumericUpDown** - untuk input Tinggi Badan (155-200 cm).
4. **CheckBox** - untuk pilihan kategori tiket (Anak, Remaja, Dewasa).
5. **Button** – untuk Pesan Tiket, Tampilkan Daftar, Keluar dari program, dan login
6. **ListBox** - untuk menampilkan keterangan pembeli.
7. **PopUp MessageBox** – untuk menampilkan pesan.

## IMPLEMENTASI PROGRAM

Kode program-Form Login

```
using System;
```

```
using System.Windows.Forms;
```

```
namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
{
    public partial class FormLogin : Form
    {
        public FormLogin()
        {
            InitializeComponent();
        }

        private void btnlogin_Click(object sender, EventArgs e)
        {
            //username dan password yang benar
            string usernameTrue = "RIXXXZA";
            string passwordtrue = "Warbam12345";

            //Mengambil Input dari user
            string usernameInput = txtUsername.Text;
            string passwordInput = txtPassword.Text;
```

```

// validasi Login

if (usernameInput == usernameTrue && passwordInput == passwordtrue)

{

// jika login Berhasil

MessageBox.Show("Login Berhasil", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information);


// Membuka Transaksi

FormTransaksi formTransaksi = new FormTransaksi();

formTransaksi.Show();


// Menyembunyikan Form Login (bukan menutup, agar aplikasi tidak berhenti)

this.Hide();

}

else

{

// Jika Login gagal

MessageBox.Show("Username atau Password salah!", "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error);

// Membersihkan field password

txtPassword.Clear();

txtPassword.Focus();

}

}

```



## Penjelasan Kode Program

### 1. Deklarasi Namespace Dan Class

```
using System;
using System.Windows.Forms;
namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
{
    Public partial class FormLogin:Form
    {
        .....
    }
}
```

- ❖ `using System;` → digunakan agar kita bisa memakai perintah dasar C# seperti `DateTime`.
- ❖ `using System.Windows.Forms;` → untuk menggunakan komponen GUI seperti `TextBox`, `Button`, `MessageBox`.
- ❖ `namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG1` → ruang lingkup (wadah) program agar tidak bentrok dengan program lain.
- ❖ `public partial class FormLogin : Form` → kelas utama yang mewarisi (inherit) dari `Form` untuk membuat aplikasi GUI.

### 2. Constructor Form

```
public FormLogin()
{
    InitializeComponent();
}
```

- ❖ `InitializeComponent()` → method yang otomatis dipanggil untuk menginisialisasi semua komponen GUI yang sudah dibuat di designer.

### 3. Validasi Login

```
if (usernameInput == usernameTrue && passwordInput == passwordtrue)
{
    // jika login Berhasil

    MessageBox.Show("Login Berhasil", "Informasi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);

    //Membuka Transaksi
    FormTransaksi formTransaksi = new FormTransaksi();
    formTransaksi.Show();
    // Menyembunyikan Form Login (bukan menutup, agar aplikasi tidak berhenti)
    this.Hide();

}
else
{
    // Jika Login gagal
    MessageBox.Show("Username atau Password salah!", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error);
    //Membersihkan field password
    txtPassword.Clear();
    txtPassword.Focus();
}
```

- ❖ Mengecek Username dan Password benar atau salah.
- ❖ Jika benar akan muncul notif login berhasil ,setelah itu langsung ke form transaksi dan jika salah akan error.

## IMPLEMENTASI PROGRAM

Kode Program Form-Transaksi

```
using System;
```

```
using System.Windows.Forms;
```

```
namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
```

```
{
```

```
    public partial class FormTransaksi : Form
```

```
    {
```

```
        public FormTransaksi()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        private void lblJudul_Load(object sender, EventArgs e)
```

```
        {
```

```
        }
```

```
        private void btnPesan_Click(object sender, EventArgs e)
```

```
        {
```

```
            // Validasi Nama
```

```
            if (txtNama.Text == "")
```

```
            {
```

```
                MessageBox.Show("Nama tidak boleh kosong!", "Validasi Nama");
```

```
                return;
```

```

    }

    // Validasi Umur & Jumlah

    if (txtUmur.Text == "" || txtJumlahTiket.Text == "")

    {

        MessageBox.Show("Isi semua data!", "Validasi Umur & Jumlah Tiket");

        return;

    }

    // Validasi Umur Minimal

    int umur = int.Parse(txtUmur.Text);

    if (umur < 14)

    {

        MessageBox.Show($" ✖ Maaf, umur minimal 14 tahun!\nUmur Anda : {umur} tahun", "Validasi Umur");

        return;

    }

    // Validasi Tinggi Badan Minimal

    int tinggi = (int)numTinggiBadan.Value;

    if (tinggi < 155)

    {

        MessageBox.Show($" ✖ Maaf, tinggi minimal 155 cm!\nTinggi Anda: {tinggi} cm", "Validasi Tinggi Badan");

        return;

```

```
}
```

```
// Validasi CheckBox (harus pilih minimal 1)
```

```
if (!chkAnak.Checked && !chkRemaja.Checked && !chkDewasa.Checked)
```

```
{
```

```
    MessageBox.Show("Pilih minimal 1 kategori tiket!", "Kategori Tiket");
```

```
    return;
```

```
}
```

```
int jumlah = int.Parse(txtJumlahTiket.Text);
```

```
int totalSemua = 0;
```

```
string kategoriDipilih = "";
```

```
int jumlahKategori = 0;
```

```
// Hitung semua kategori yang dicentang
```

```
if (chkAnak.Checked) // Checked itu property bool (true/false)
```

```
{
```

```
    int hargaAnak = 50000 * jumlah;
```

```
    totalSemua += hargaAnak;
```

```
    kategoriDipilih += "Anak ";
```

```
    jumlahKategori++;
```

```
}
```

```
if (chkRemaja.Checked)
```

```

{
    int hargaRemaja = 75000 * jumlah;

    totalSemua += hargaRemaja;

    kategoriDipilih += "Remaja ";

    jumlahKategori++;
}

if (chkDewasa.Checked)
{
    int hargaDewasa = 100000 * jumlah;

    totalSemua += hargaDewasa;

    kategoriDipilih += "Dewasa ";

    jumlahKategori++;
}

int totalTiket = jumlah * jumlahKategori;

int total = totalSemua;

// Diskon 10% kalau total tiket 5+

string diskonText = "";

if (totalTiket >= 5)
{
    total = total - (total * 10 / 100);

    diskonText = " (Diskon 10%)";
}

```

```

// Input Uang Bayar

string inputBayar = Microsoft.VisualBasic.Interaction.InputBox($"Total Bayar: Rp
{total:N0}{diskonText}\nMasukkan uang bayar:\n\n❌ Ingat jangan utang yaa!!!",
"Pembayaran", "");

if (inputBayar == "") return;

// Validasi uang bayar

if (!int.TryParse(inputBayar, out int uangBayar))

{
    MessageBox.Show("Nominal tidak valid!", "Pembayaran");

    return;
}

// Cek cukup atau tidak

if (uangBayar < total)

{
    MessageBox.Show($"❌ Nahkan Uangnya kurang!\nKurang: Rp {(total -
uangBayar):N0}");

    return;
}

int kembalian = uangBayar - total;

// Konfirmasi

string pesan = $"Nama : {txtNama.Text}\n";

```

```

pesan += $"Umur    : {umur} tahun\n";

pesan += $"Tinggi  : {tinggi} cm ✓\n";

pesan += $"Kategori : {kategoriDipilih}\n";

pesan += $"Jumlah   : {totalTiket} tiket ({jumlah} x {jumlahKategori} kategori)\n";

pesan += $"Total    : Rp {total:N0} {diskonText}\n";

pesan += $"Bayar    : Rp {uangBayar:N0}\n";

pesan += $"Kembalian: Rp {kembalian:N0}\n\n";

pesan += "Konfirmasi pemesanan?";

if (MessageBox.Show(pesan, "Konfirmasi", MessageBoxButtons.YesNo) ==
DialogResult.Yes)

{

    // Tambah ke list Pembeli

    listPembeli.Items.Add($" {txtNama.Text} - {umur} th - {tinggi} cm -
{kategoriDipilih} - {totalTiket} tiket - Rp {total:N0} - Rp {kembalian} ");

    MessageBox.Show($"✓ Pemesanan berhasil!\nDan jangan lupa kembali lagi
yaaa😍😍😍 \nKembalian: Rp {kembalian:N0}", "Sukses");

    // Reset form

    txtNama.Clear();

    txtUmur.Clear();

    txtJumlahTiket.Clear();

    numTinggiBadan.Value = 120;

    chkAnak.Checked = false;

    chkRemaja.Checked = false;

```



```

        chkDewasa.Checked = false;

    }

}

```

## Penjelasan Kode Program

### 1. Deklarasi Namespace dan Class

```

using System;
using System.Windows.Forms;

namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
{
    public partial class FormTransaksi : Form
    {
        public FormTransaksi()
        {
            InitializeComponent();
        }
    }
}

```

- ❖ `using System;` → digunakan agar kita bisa memakai perintah dasar C# seperti `DateTime`.
- ❖ `using System.Windows.Forms;` → untuk menggunakan komponen GUI seperti `TextBox`, `Button`, `MessageBox`.
- ❖ `namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1` → ruang lingkup (wadah) program agar tidak bentrok dengan program lain.
- ❖ `public partial class FormTransaksi : Form` → kelas utama yang mewarisi (inherit) dari `Form` untuk membuat aplikasi GUI.

### 2. Constructur Form

```

public FormTransaksi()
{
    InitializeComponent();
}

```

- ❖ `InitializeComponent()` → method yang otomatis dipanggil untuk menginisialisasi semua komponen GUI yang sudah dibuat di designer.

### 3. Validasi Input Nama

```
if (txtNama.Text == "")
{
    MessageBox.Show("Nama tidak boleh kosong!", "Validasi Nama");
    return;
}
```

- ❖ Mengecek apakah TextBox nama kosong atau tidak.
- ❖ Jika kosong, tampilkan pesan error dan return (keluar dari method, tidak lanjut ke kode berikutnya).

### 4. Validasi Umur dan Jumlah Tiket

```
if (txtUmur.Text == "" || txtJumlahTiket.Text == "")
{
    MessageBox.Show("Isi semua data!", "Validasi Umur & Jumlah Tiket");
    return;
}
int umur = int.Parse(txtUmur.Text);
if (umur < 14)
{
    MessageBox.Show($" ✖ Maaf, umur minimal 14 tahun!\nUmur Anda : {umur} tahun",
"Validasi Umur");
    return;
}
```

- ❖ || (OR) → jika salah satu kosong, tampilkan error.
- ❖ int.Parse() → mengubah teks dari TextBox menjadi angka integer.
- ❖ Validasi umur minimal 14 tahun untuk keselamatan.

### 5. Validasi Tinggi Badan

```
int tinggi = (int)numTinggiBadan.Value;
if (tinggi < 155)
{
    MessageBox.Show($" ✖ Maaf, tinggi minimal 155 cm!\nTinggi Anda: {tinggi} cm",
"Validasi Tinggi Badan");
    return;
}
```

- ❖ numTinggiBadan.Value → mengambil nilai dari NumericUpDown.
- ❖ (int) → casting (mengubah) dari decimal ke integer.
- ❖ Validasi tinggi minimal 155 cm sesuai standar keselamatan roller coaster.

## 6. Validasi CheckBox

```
if (!chkAnak.Checked && !chkRemaja.Checked && !chkDewasa.Checked)
{
    MessageBox.Show("Pilih minimal 1 kategori tiket!", "Kategori Tiket");
    return;
}
```

- ❖ ! (NOT) → artinya "tidak".
- ❖ && (AND) → semua kondisi harus benar.
- ❖ Jika semua checkbox TIDAK dicentang, tampilkan error.

## 7. Perhitungan Harga Mulai-Kategori

```
if (chkAnak.Checked) // Checked itu property bool (true/false)
{
    int hargaAnak = 50000 * jumlah;
    totalSemua += hargaAnak;
    kategoriDipilih += "Anak ";
    jumlahKategori++;
}
if (chkRemaja.Checked)
{
    int hargaRemaja = 75000 * jumlah;
    totalSemua += hargaRemaja;
    kategoriDipilih += "Remaja ";
    jumlahKategori++;
}
if (chkDewasa.Checked)
{
    int hargaDewasa = 100000 * jumlah;
    totalSemua += hargaDewasa;
    kategoriDipilih += "Dewasa ";
    jumlahKategori++;
}
```

```
int totalTiket = jumlah * jumlahKategori;
int total = totalSemua;
```

```
// Diskon 10% kalau total tiket 5+
string diskonText = "";
if (totalTiket >= 5)
{
    total = total - (total * 10 / 100);
    diskonText = "(Diskon 10%)";
}
```

- ❖ Menggunakan IF biasa (bukan IF-ELSE) karena bisa memilih lebih dari satu kategori.
- ❖ += → operator penugasan untuk menambahkan nilai.
- ❖ ++ → increment, menambah 1.
- ❖ Menghitung total untuk setiap kategori yang dicentang, lalu dijumlahkan semua.
- ❖ Ada diskon ketika membeli 5 tiket atau diatas 5.

## 8. Input Uang Bayar menggunakan Input Box

```
string inputBayar = Microsoft.VisualBasic.Interaction.InputBox($"Total Bayar: Rp
{total:N0}{diskonText}\nMasukkan uang bayar:\n\n\n❌ Ingat jangan utang yaa!!!",
"Pembayaran", "");
```

```
if (inputBayar == "") return;
```

- ❖ InputBox → menampilkan dialog untuk input.
- ❖ \n → newline (baris baru).
- ❖ Jika tekan cancel akan keluar.

## 9. Validasi Uang Bayar

```
if (!int.TryParse(inputBayar, out int uangBayar))
{
    MessageBox.Show("Nominal tidak valid!", "Pembayaran");
    return;
}
// Cek cukup atau tidak
if (uangBayar < total)
{
```

```

        MessageBox.Show($" ✕ Nahkan Uangnya kurang!\nKurang: Rp {(total -
uangBayar):N0}");

```

```

        return;
    }

```

```

    int kembalian = uangBayar - total;

```

- ❖ Validasi apakah uang cukup atau tidak.
- ❖ Kembalian = Uang Bayar – Total.

## 10. Konfirmasi Pesanan

```

string pesan = $"Nama : {txtNama.Text}\n";

```

```

pesan += $"Umur : {umur} tahun\n";

```

```

pesan += $"Tinggi : {tinggi} cm ✓\n";

```

```

pesan += $"Kategori : {kategoriDipilih}\n";

```

```

pesan += $"Jumlah : {totalTiket} tiket ({jumlah} x {jumlahKategori} kategori)\n";

```

```

pesan += $"Total : Rp {total:N0} {diskonText}\n";

```

```

pesan += $"Bayar : Rp {uangBayar:N0}\n";

```

```

pesan += $"Kembalian: Rp {kembalian:N0}\n\n";

```

```

pesan += "Konfirmasi pemesanan?";

```

```

if (MessageBox.Show(pesan, "Konfirmasi", MessageBoxButtons.YesNo) ==
DialogResult.Yes)

```

- ❖ MessageBox.Show dengan MessageBoxButtons.YesNo → menampilkan tombol Yes dan No.

## 11. Simpan ke ListBox

```

listPembeli.Items.Add($" {txtNama.Text} - {umur} th - {tinggi} cm - {kategoriDipilih} -
{totalTiket} tiket - Rp {total:N0} - Rp {kembalian} ");

```

- ❖ listPembeli.Items.Add() → menambahkan item baru ke ListBox.
- ❖ Data ditampilkan dalam listbox.

## 12. Reset Form

```

// Reset form

```

```

txtNama.Clear();

```

```

txtUmur.Clear();

```

```

txtJumlahTiket.Clear();

```

```

numTinggiBadan.Value = 120;

```

```
chkAnak.Checked = false;
```

```
chkRemaja.Checked = false;
```

```
chkDewasa.Checked = false;
```

- ❖ Clear() → mengosongkan TextBox.

- ❖ Value = 0 → reset NumericUpDown ke nilai default.

- ❖ Checked = false → agar checkbox Kembali kosong/tidak dicentang

### 13. Daftar Pembeli

```
if (listPembeli.Items.Count == 0)
```

```
{
```

```
    MessageBox.Show("Belum ada data pembeli");
```

```
    return;
```

```
}
```

```
string daftar =
```

```
"=====
=====\\n";
```

```
daftar += "        DAFTAR PEMBELI TIKET\\n";
```

```
daftar +=
```

```
"=====
=====\\n\\n";
```

```
// FOR LOOP untuk tampilkan daftar
```

```
for (int i = 0; i < listPembeli.Items.Count; i++)
```

```
{
```

```
    daftar += $"{i + 1}. {listPembeli.Items[i]}\\n";
```

```
}
```

```
daftar +=
```

```
 $"\\n=====
=====";
```

```
daftar += $"\\nTotal Pembeli: {listPembeli.Items.Count}";
```

```
MessageBox.Show(daftar, "Rincian Pembeli");
```

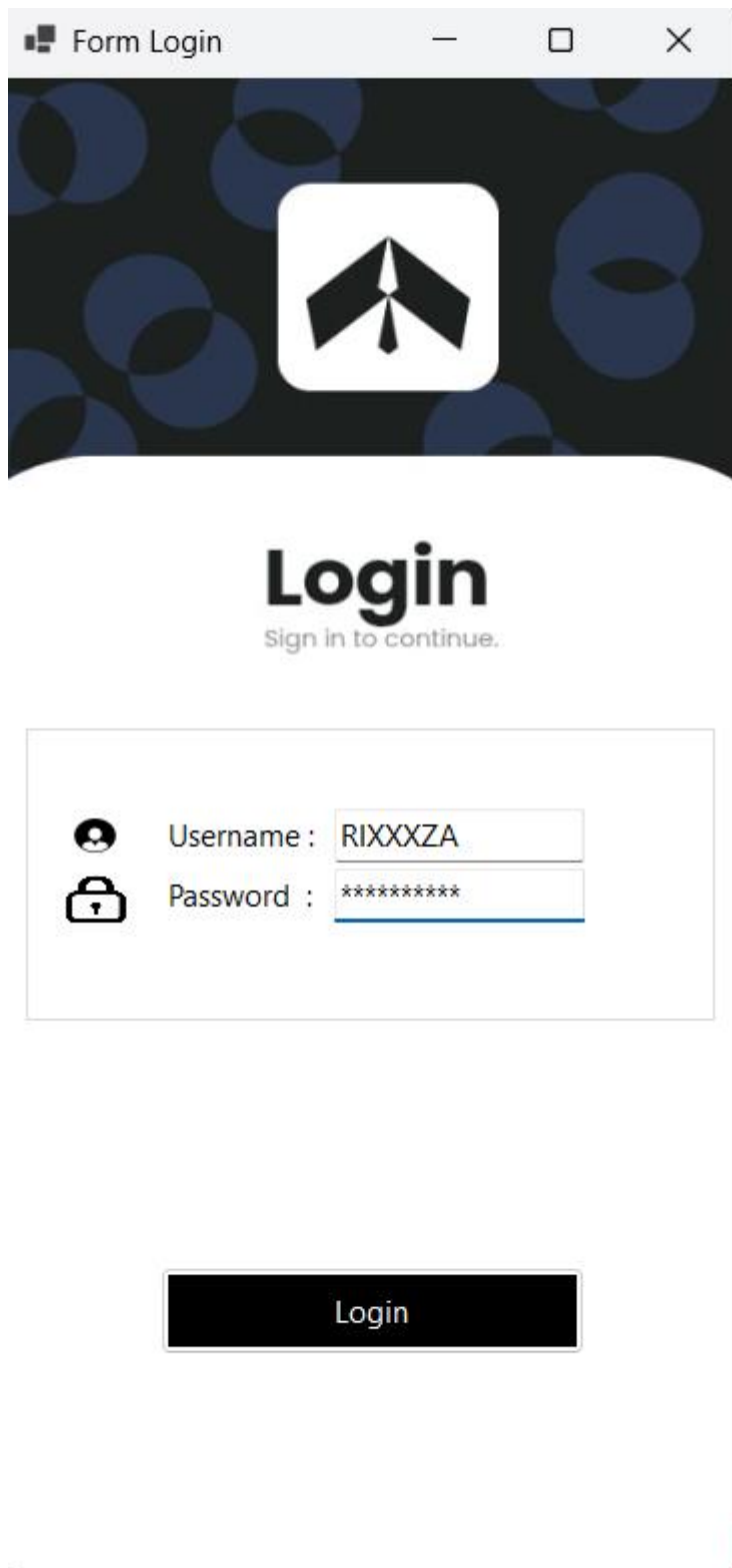
```
}
```

- ❖ for loop → perulangan dari 0 sampai jumlah item - 1.


- ❖ i + 1 → nomor urut dimulai dari 1 (bukan 0).

- ❖ listPembeli.Items[i] → mengambil item ke-i dari ListBox.

### Tampilan Input Data Form Login





Form Login



# Login

Sign in to continue.

 Username :

 Password :

Login

## Lampiran Form Login

### Kode Program Lengkap

```
using System;
using System.Windows.Forms;

namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
{
    4 references
    public partial class FormLogin : Form
    {
        1 reference
        public FormLogin()
        {
            InitializeComponent();
        }

        1 reference
        private void btnlogin_Click(object sender, EventArgs e)
        {
            //username dan password yang benar
            string usernameTrue = "RXXXXZA";
            string passwordtrue = "Warbam12345";

            //Mengambil Input dari user
            string usernameInput = txtUsername.Text;
            string passwordInput = txtPassword.Text;

            //validasi Login
            if (usernameInput == usernameTrue && passwordInput == passwordtrue)
            {
                // jika login Berhasil
                MessageBox.Show("Login Berhasil", "Informasi", MessageBoxButtons.OK, MessageBoxIcon.Information);

                //Membuka Transaksi
                FormTransaksi formTransaksi = new FormTransaksi();
                formTransaksi.Show();

                // Menyembunyikan Form Login (bukan menutup, agar aplikasi tidak berhenti)
                this.Hide();
            }
            else
            {
                // Jika Login gagal
                MessageBox.Show("Username atau Password salah!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                //Membersihkan field password
                txtPassword.Clear();
                txtPassword.Focus();
            }
        }
    }
}
```



## Tampilan Input Data Form Transaksi

The screenshot shows a web application window titled "Form Transaksi". The background is a vibrant orange and yellow illustration of a roller coaster park with various rides like a roller coaster, Ferris wheel, and hot air balloons. The form contains the following fields and options:

- Masukan Nama :** BUDI SANTOSO
- Umur :** 29
- Tinggi Badan :** 176
- Jumlah Tiket :** 3
- Kategori Tiket :** ☒ Anak 3 - 12 Tahun, ☒ Remaja 14 - 21 Tahun, ☒ Dewasa 22 - 50
- Buttons:** Pesan (green), Tampilkan Daftar (orange), Kewar (red)
- Daftar Pembeli:** A large empty text area for listing buyers.
- Ketentuan Harga Tiket:** A small box showing prices: Anak - Anak : Rp 50.000, Remaja : Rp 75.000, Dewasa : Rp 100.000. Below it, it says "JANGAN LUPA DIBELI YAAA".

Tampilan

Pop

Up

Bayar

The screenshot shows a "Pembayaran" (Payment) dialog box with a close button (X) in the top right corner. It contains the following information:

- Total Bayar:** Rp 607.500 (Diskon 10%)
- Masukkan uang bayar:** A label for the payment input field.
- Payment Input:** A text box containing the value "700000".
- Buttons:** OK and Cancel.
- Warning:** A red "X" icon followed by the text "Ingat jangan utang yaa!!!".

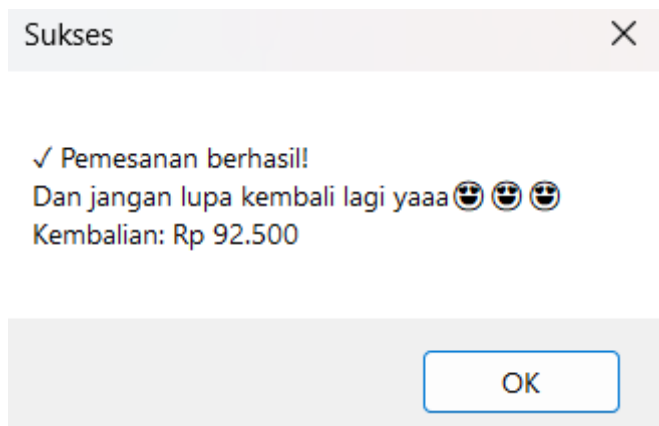
## Konfirmasi Bayar

The screenshot shows a "Konfirmasi" (Confirmation) dialog box with a close button (X) in the top right corner. It displays a summary of the transaction:

- Nama :** BUDI SANTOSO
- Umur :** 29 tahun
- Tinggi :** 176 cm ✓
- Kategori :** Anak Remaja Dewasa
- Jumlah :** 9 tiket (3 x 3 kategori)
- Total :** Rp 607.500 (Diskon 10%)
- Bayar :** Rp 700.000
- Kembalian:** Rp 92.500

Below the summary, it asks "Konfirmasi pemesanan?" (Confirm order?). At the bottom, there are two buttons: "Yes" and "No".

### Tampilan Sukses Pembayaran



### Tampilan List Box



## LAMPIRAN Form Transaksi

### Kode Program Lengkap

```
using System;
using System.Windows.Forms;

namespace PROJECT_AKHIR_FAJRI_RIQZA_M_XPPLG_1
{
    4 references
    public partial class FormTransaksi : Form
    {
        1 reference
        public FormTransaksi()
        {
            InitializeComponent();
        }
        1 reference
        private void lblJudul_Load(object sender, EventArgs e)
        {
        }
        1 reference
        private void btnPesan_Click(object sender, EventArgs e)
        {
            // Validasi Nama
            if (txtNama.Text == "")
            {
                MessageBox.Show("Nama tidak boleh kosong!", "Validasi Nama");
                return;
            }

            // Validasi Umur & Jumlah Tiket
            if (txtUmur.Text == "" || txtJumlahTiket.Text == "")
            {
                MessageBox.Show("Isi semua data!", "Validasi Umur & Jumlah Tiket");
                return;
            }

            // Validasi Umur Minimal
            int umur = int.Parse(txtUmur.Text);
            if (umur < 14)
            {
                MessageBox.Show($"❌ Maaf, umur minimal 14 tahun!\nUmur Anda : {umur} tahun", "Validasi Umur");
                return;
            }

            // Validasi Tinggi Badan Minimal
            int tinggi = (int)numTinggiBadan.Value;
            if (tinggi < 155)
            {
                MessageBox.Show($"❌ Maaf, tinggi minimal 155 cm!\nTinggi Anda: {tinggi} cm", "Validasi Tinggi Badan");
                return;
            }

            // Validasi CheckBox (harus pilih minimal 1)
            if (!chkAnak.Checked && !chkRemaja.Checked && !chkDewasa.Checked)
            {
                MessageBox.Show("Pilih minimal 1 kategori tiket!", "Kategori Tiket");
                return;
            }

            int jumlah = int.Parse(txtJumlahTiket.Text);
            int totalSemua = 0;
            string kategoriDipilih = "";
            int jumlahKategori = 0;

            // Hitung semua kategori yang dipilih
        }
    }
}
```

```

// Mitung semua kategori yang dicentang
if (chkAnak.Checked) // Checked itu property bool (true/false)
{
    int hargaAnak = 8000 * jumlah;
    totalSemua += hargaAnak;
    kategoriDipilih += "Anak ";
    jumlahKategori++;
}

if (chkRemaja.Checked)
{
    int hargaRemaja = 7500 * jumlah;
    totalSemua += hargaRemaja;
    kategoriDipilih += "Remaja ";
    jumlahKategori++;
}

if (chkDewasa.Checked)
{
    int hargaDewasa = 10000 * jumlah;
    totalSemua += hargaDewasa;
    kategoriDipilih += "Dewasa ";
    jumlahKategori++;
}

int totalTiket = jumlah * jumlahKategori;
int total = totalSemua;

// Diskon 10% kalau total tiket 5+
string diskonText = "";
if (totalTiket >= 5)
{
    total = total - (total * 10 / 100);
    diskonText = " (Diskon 10%)";
}

// Input Uang Bayar
string inputBayar = Microsoft.VisualBasic.Interaction.InputBox($"Total Bayar: Rp {total:N0}{diskonText}\nMasukkan uang bayar:\n\nXIngat jangan utang ya!!!", "Pembayaran", "");
if (inputBayar == "") return;

// Validasi uang bayar
if (!int.TryParse(inputBayar, out int uangBayar))
{
    MessageBox.Show("Nominal tidak valid!", "Pembayaran");
    return;
}

// Cek cukup atau tidak
if (uangBayar < total)
{
    MessageBox.Show($"X Nahkan Uangnya kurang!\nMurang: Rp {(total - uangBayar):N0}");
    return;
}

int kembalian = uangBayar - total;

// Konfirmasi
string pesan = $"Nama      : {txtNama.Text}\n";
pesan += $"Umur      : {umur} tahun\n";
pesan += $"Tinggi    : {tinggi} cm\n";
pesan += $"Kategori  : {kategoriDipilih}\n";
pesan += $"Jumlah    : {totalTiket} tiket ({jumlah} x {jumlahKategori} kategori)\n";
pesan += $"Total     : Rp {total:N0}{diskonText}\n";
pesan += $"Bayar     : Rp {uangBayar:N0}\n";
pesan += $"Kembalian: Rp {kembalian:N0}\n\n";
pesan += "Konfirmasi pemesanan?";

if (MessageBox.Show(pesan, "Konfirmasi", MessageBoxButtons.YesNo) == DialogResult.Yes)
{
    // Tambah ke List Pembeli
    listPembeli.Items.Add($"({txtNama.Text}) - {umur} th - {tinggi} cm - {kategoriDipilih} - {totalTiket} tiket - Rp {total:N0} - Rp {kembalian} ");

    MessageBox.Show($"√ Pemesanan berhasil!\nDan jangan lupa kembali lagi yaaa😊😊😊 \nKembalian: Rp {kembalian:N0}", "Sukses");

    // Reset form
    txtNama.Clear();
    txtUmur.Clear();
    txtTinggi.Clear();
    numTinggiBadan.Value = 0;
    chkAnak.Checked = false;
    chkRemaja.Checked = false;
    chkDewasa.Checked = false;
}

reference
private void txtNama_TextChanged(object sender, EventArgs e)

reference
private void numTinggiBadan_ValueChanged(object sender, EventArgs e)
{
    numTinggiBadan.Maximum = 200; // isi 200 cm
}

reference
private void lblJudul_Click(object sender, EventArgs e)

reference
private void btnTampilkanDaftar_Click_1(object sender, EventArgs e)
{
    if (listPembeli.Items.Count == 0)
    {
        MessageBox.Show("Belum ada data pembeli");
        return;
    }

    string daftar = "-----\n";
    daftar += "                        DAFTAR PEMBELI TIKET\n";
    daftar += "-----\n\n";

    // FOR LOOP untuk tampilkan daftar
    for (int i = 0; i < listPembeli.Items.Count; i++)
    {
        daftar += $"{i + 1}. {listPembeli.Items[i]}\n";
    }

    daftar += $"-----\n";
    daftar += $"Total Pembeli: {listPembeli.Items.Count}";

    MessageBox.Show(daftar, "Rincian Pembeli");
}

```

```
private void btnkeluar_Click(object sender, EventArgs e)
{
    DialogResult result = MessageBox.Show("Apakah Anda yakin ingin keluar?", "Konfirmasi Keluar", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
    if (result == DialogResult.Yes)
    {
        Application.Exit();
    }
}
```