

PROYEK AKHIR SEMESTER GASAL



NAMA : Rizki Iqbal Tawakkal
KELAS : X PPLG 1
NIS : 258728
JUDUL PROYEK : Aplikasi Kasir warung kang IQBAL

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 Makanan Dan Minuman. 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,Makanan,Minuman,No Meja,Porsi).
2. Memasukkan jumlah Makanan Dan Minuman yang akan dibeli.
3. Menghitung total harga secara otomatis berdasarkan kategori dan jumlah makanan dan minuman yang dipilih.
4. Menghitung kembalian uang secara otomatis setelah input pembayaran.
5. Menampilkan daftar pembeli yang sudah melakukan pembelian.

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:

1. Collections - List

List digunakan untuk menyimpan kumpulan data dinamis:

csharp

```
private List<MenuItem> menuItems;
```

```
private List<CartItem> cart;
```

```
menuItems = new List<MenuItem>
```

```
{
```

```
    new MenuItem { Name = "Nasgor", Price = 15000 },
```

```
    new MenuItem { Name = "Es Teh", Price = 3500 }
```

```
};
```

- List<T> - Generic collection yang bisa bertambah/berkurang

- Add() - Menambah item

- Remove() - Menghapus item

- Clear() - Mengosongkan list

2. LINQ (Language Integrated Query)

LINQ digunakan untuk query dan manipulasi data:

csharp

```
// Filter berdasarkan kategori
```

```
var filtered = menuItems.Where(m => m.Category == "Food");
```

```
// Sorting ascending
```

```

var sorted = menuItems.OrderBy(m => m.Price);

// Sorting descending

var sorted = menuItems.OrderByDescending(m => m.Price);

// Hitung total

decimal total = cart.Sum(c => c.Subtotal);

// Cari item

var item = cart.FirstOrDefault(c => c.MenuItem.Name == "Nasgor");

```

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($" X Nahkan Uangnya kurang!\nKurang: Rp {(total - 
uangBayar):N0}");

    return;
}

```

4. Input dan Output

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

Contoh:

```
string namaPemesan = txtnamapemesan.Text;
string makanan = cmbMakanan.SelectedItem?.ToString() ?? "";
string minuman = cmbMinuman.SelectedItem?.ToString() ?? "";
string porsi = txtporsi.Text;
string noMeja = txtnomeja.Text;
string pembayaran = cmbpembayaran.SelectedItem?.ToString() ?? "";
int jumlah = 0;
// Input Uang Bayar
string inputBayar = Microsoft.VisualBasic.Interaction.InputBox ("Masukkan uang
bayar:\n\n\n Ingat jangan utang yaa!!!", "Pembayaran", "");
// Tambah ke list Pembeli / Outputnya
listPembeli.Items.Add($"{txtNama.Text} - {Makanan} - {Minuman} - {No Meja} - {Porsi}
- Rp {total:N0} - Rp {kembalian} ");
```

5. Perhitungan Multi-Kategori

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

```
int Porsi = int.Parse(porsi);
switch (makanan)
{
    case "Nasi Goreng:13.000":
        jumlah += 13000 * Porsi;
        break;
    case "Nasi Padang:15.000":
        jumlah += 15000 * Porsi;
        break;
    case "Sate Ayam:10.000":
        jumlah += 10000 * Porsi;
```

```

        break;
    case "Nasi kucing:7.000":
        jumlah += 7000 * Porsi;
        break;
    }
    switch (minuman) //minuman
    {
        case "Es Teh:4.000":
            jumlah += 4000 * Porsi;
            break;
        case "Air Mineral:3.000":
            jumlah += 3000 * Porsi;
            break;
        case "Teh Panas:3.000":
            jumlah += 3000 * Porsi;
            break;
    }
}

```

6.String Formatting

Format string untuk tampilan:

```

csharp
// Interpolation

string text = $"Total: Rp {total:N0}";

// Format currency

string price = $"Rp {15000:N0}"; // Rp 15,000

```

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

Digunakan untuk membuat tampilan aplikasi kasir yang lebih interaktif.

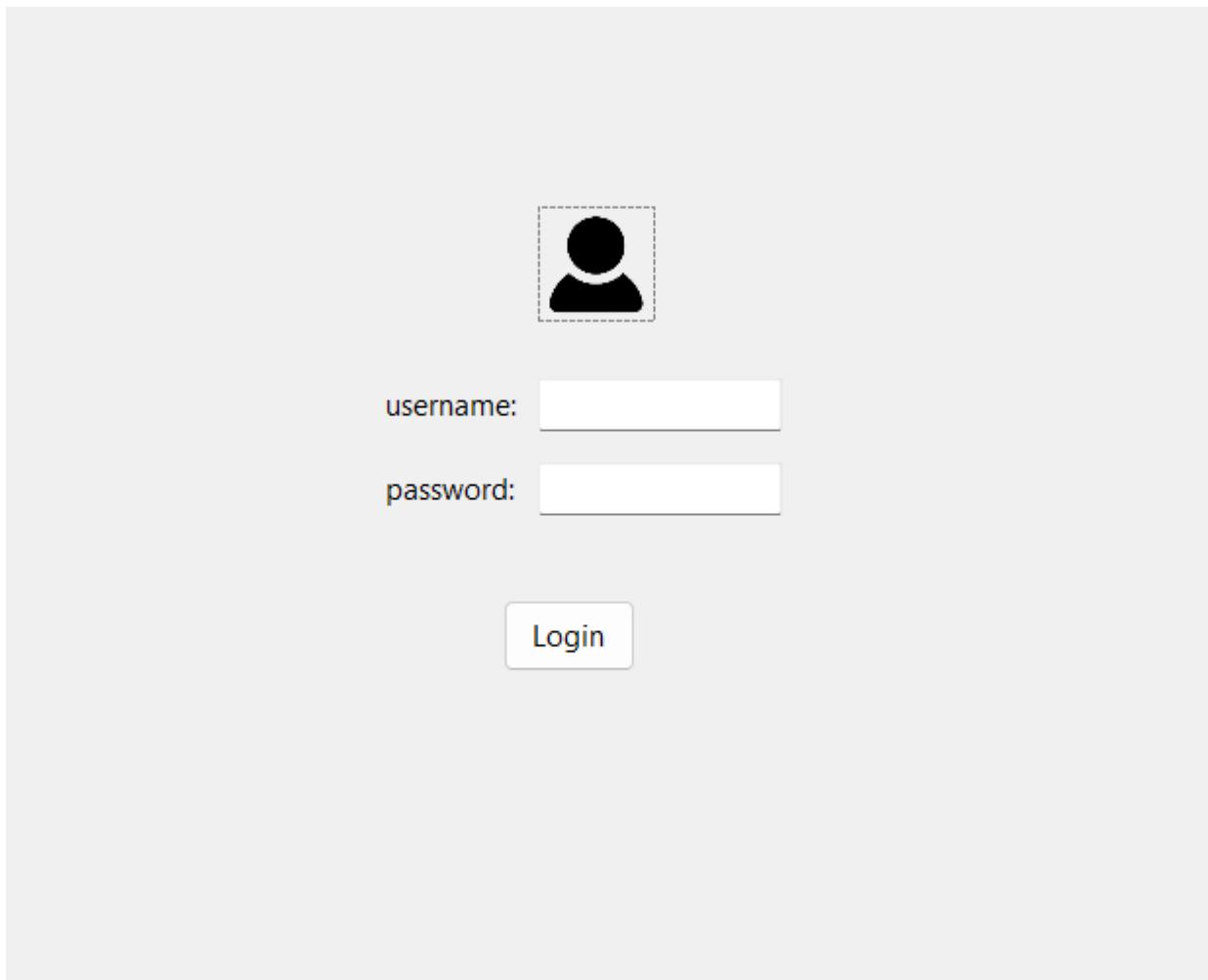
Komponen yang sering digunakan:

- **Label** → menampilkan teks seperti "Masukan Nama", "Umur", "Total Tiket".
- **TextBox** → untuk memasukkan data nama, umur, jumlah tiket.
- **DataGrid** → untuk mengumpulkan data yang dipilih.
- **Button** → untuk mengeksekusi perintah seperti *Pesan*, *Tampilkan Daftar*, dan *Keluar*.
- **Combo Box**→untuk memilih makanan dan minuman

PERANCANGAN PROGRAM

Rancangan Tampilan Pembelian Tiket Roller Coaster

Komponen GUI yang digunakan:





1. **Label** - untuk judul dan keterangan.
2. **TextBox** - untuk input Nama,Porsi,No Meja,pembayaran,Bayar.
3. **Combo Box** – Untuk memilih Makanan dan Minuman
4. **Button**- untuk Pesan Makanan dan Minuman, Tampilkan Daftar, Keluar dari program,dan login
5. **DataGridView** – untuk menampilkan pesanan.

IMPLEMENTASI PROGRAM

Kode program-Form Login:

```
public Form1()
{
    InitializeComponent();
}

private void btnlogin_Click(object sender, EventArgs e)
{
    //Username dan password yang benar
    string username = "iqbal";
```

```

string password = "12345";

//Mengambil input dari user
string usernameInput = txtusername.Text;
string passwordInput = txtpassword.Text;

//Validasi login
if (usernameInput == username && passwordInput == password)
{
    //Jika login berhasil
    MessageBox.Show("login berhasil!", "Informasi", MessageBoxButtons.OK,
    MessageBoxIcon.Information);

    //Membuka form 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();
}
}

```

Lampiran kode lengkap

```
private void btnpesan_Click(object sender, EventArgs e)
{
    // tombol pesan
    string namaPemesan = txtnamapemesan.Text;
    string makanan = cmbMakanan.SelectedItem.ToString() ?? "";
    string minuman = cmbMinuman.SelectedItem?.ToString() ?? "";
    string porsi = txtorsi.Text;
    string noMeja = txtnomeja.Text;
    string pembayaran = cmbpembayaran.SelectedItem?.ToString() ?? "";
    int jumlah = 0;
    if (string.IsNullOrWhiteSpace(namaPemesan) || string.IsNullOrWhiteSpace(makanan) ||
        string.IsNullOrWhiteSpace(minuman) || string.IsNullOrWhiteSpace(porsi) ||
        string.IsNullOrWhiteSpace(noMeja) || string.IsNullOrWhiteSpace(pembayaran))
    {
        MessageBox.Show("Harap lengkapi semua data!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }
}
```

```
//Hitung Total Harga
int Porsi = int.Parse(porsi);
switch (makanan)
{
    case "Nasi Goreng:13.000":
        jumlah += 13000 * Porsi;
        break;
    case "Nasi Padang:15.000":
        jumlah += 15000 * Porsi;
        break;
    case "Sate Ayam:10.000":
        jumlah += 10000 * Porsi;
        break;
    case "Nasi kucing:7.000":
        jumlah += 7000 * Porsi;
        break;
}
```

```
switch (minuman) //minuman
{
    case "Es Teh:4.000":
        jumlah += 4000 * Porsi;
        break;
    case "Air Mineral:3.000":
        jumlah += 3000 * Porsi;
        break;
    case "Teh Panas:3.000":
        jumlah += 3000 * Porsi;
        break;
}
```

```
//Menambahkan Baris Ke DataGridView
string[] row = new string[] { namaPemesan, makanan, minuman, porsi, noMeja, pembayaran, jumlah.ToString() };
//dgv1.Rows.Add(row);

// tambahkan data ke GridView
ordersTable = new DataTable();
ordersTable.Columns.Add("Nama Pemesan", typeof(string));
ordersTable.Columns.Add("Makanan", typeof(string));
ordersTable.Columns.Add("Minuman", typeof(string));
ordersTable.Columns.Add("Porsi", typeof(string));
ordersTable.Columns.Add("No Meja", typeof(string));
ordersTable.Columns.Add("Pembayaran", typeof(string));
ordersTable.Columns.Add("Total Harga", typeof(string));
ordersTable.Rows.Add(namaPemesan, makanan, minuman, porsi, noMeja, pembayaran, jumlah);
dgv1.DataSource = ordersTable;
```

```
1 reference
private void FormTransaksi_Load(object sender, EventArgs e)
{
    // Tambahkan data ComboBox
    cmbMakanan.Items.Add("Nasi Goreng:13.000");
    cmbMakanan.Items.Add("Nasi Padang:15.000");
    cmbMakanan.Items.Add("Sate Ayam:10.000");
    cmbMakanan.Items.Add("Nasi kucing:7.000");
    cmbMinuman.Items.Add("Es Teh:4.000");
    cmbMinuman.Items.Add("Es Jeruk:5.000");
    cmbMinuman.Items.Add("Air Mineral:3.000");
    cmbMinuman.Items.Add("Teh Panas:3.000");

    //Tambahkan data Combobox
    cmbpembayaran.Items.Add("Cash");
    cmbpembayaran.Items.Add("Qris");
    cmbpembayaran.Items.Add("Ovo");
    cmbpembayaran.Items.Add("Dana");
    cmbpembayaran.Items.Add("Gopay");

    // bersihkan selection
    dgv1.ClearSelection();
    cmbpembayaran.SelectedIndex = -1;
    dgv1.DataSource = new object[] {
        };
}
```

```
// Bersihkan input
txtnamapemesan.Clear();
cmbMakanan.SelectedIndex = -1;
cmbMinuman.SelectedIndex = -1;
txtnomeja.Clear();
txtporsi.Clear();
txtnamapemesan.Focus();
txtnomeja.Focus();
txtporsi.Focus();
cmbpembayaran.SelectedIndex = -1;

lblTotal.Text = jumlah.ToString();
```

```
1 reference
private void btnClear_Click(object sender, EventArgs e)
{
    if (dgv1.CurrentRow != null)
    {
        dgv1.Rows.RemoveAt(dgv1.CurrentRow.Index);
    }
    else
    {
        MessageBox.Show("Pilih baris yang akan dihapus!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

```
1 reference
private void btnBayar_Click(object sender, EventArgs e)
{
    //Hitung Kembalian
    if (!string.IsNullOrEmpty(txtTotalBayar.Text))
    {
        if (Convert.ToDouble(txtTotalBayar.Text) >= Convert.ToDouble(lblTotal.Text))
        {
            double kembalian = Convert.ToDouble(txtTotalBayar.Text) - Convert.ToDouble(lblTotal.Text);
            lblKembalian.Text = kembalian.ToString();
        }
        else
        {
            MessageBox.Show("Uang tidak cukup!", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

Penjelasan Kode Program

1. Deklarasi Namespace dan Class

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

```
namespace PROJECTAKHIR_RizkiIqbalTawakkal_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 PROJECTAKHIR_RizkiIqbalTawakkal_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.

Penjelasan:

- Mengelompokkan kelas-kelas yang secara fungsional terkait ke dalam satu unit.
- Mempermudah developer untuk mengidentifikasi dan mencari kode sumber yang relevan, karena struktur *namespace* seringkali mencerminkan struktur folder proyek.

2. Constructur Form

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

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

Penjelasan:

-Input Data Terstruktur: Menyediakan cara yang terorganisir bagi pengguna untuk memasukkan data yang berbeda-beda (Nama,Makanan,Minuman,dll.) dan memaketkannya menjadi satu kesatuan.

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).

Penjelasan:

-Integritas Data: Memastikan bahwa data yang disimpan di database (misalnya, nama pelanggan) bersih, terstruktur, dan hanya mengandung karakter yang diizinkan (tidak ada simbol atau angka yang tidak relevan).

-Pencegahan Kesalahan: Mencegah *error* pada sistem di masa depan yang disebabkan oleh format data yang salah atau *input* yang kosong.

-Keamanan: Mencegah serangan *injection* sederhana dengan membatasi jenis karakter yang boleh dimasukkan.

4. Perhitungan Harga Mulai-Kategori

```
int Porsi = int.Parse(porsi);
switch (makanan)
{
    case "Nasi Goreng:13.000":
        jumlah += 13000 * Porsi;
        break;
    case "Nasi Padang:15.000":
        jumlah += 15000 * Porsi;
        break;
    case "Sate Ayam:10.000":
        jumlah += 10000 * Porsi;
        break;
    case "Nasi kucing:7.000":
        jumlah += 7000 * Porsi;
        break;
}
switch (minuman) //minuman
{
    case "Es Teh:4.000":
        jumlah += 4000 * Porsi;
        break;
    case "Air Mineral:3.000":
        jumlah += 3000 * Porsi;
        break;
```

```
case "Teh Panas:3.000":  
    jumlah += 3000 * Porsi;  
    break;
```

Penjelasan:

- **Standarisasi Harga:** Memastikan bahwa produk dalam kategori tertentu, terlepas dari varian spesifiknya, memiliki harga jual minimum yang telah ditetapkan.
- **Awal Negosiasi:** Dalam bisnis B2B (Business-to-Business) atau layanan, harga ini bisa menjadi titik awal penawaran sebelum ada penyesuaian berdasarkan kuantitas, kustomisasi, atau diskon.
- **Informasi Pembeli:** Memberikan gambaran cepat kepada calon pembeli mengenai kisaran harga yang harus mereka siapkan untuk produk di kategori tersebut.

5. Input Uang Bayar menggunakan Input Box

```
//Hitung Kembalian  
if (!string.IsNullOrEmpty(txtotalbayar.Text))  
  
    if (Convert.ToDouble(txtotalbayar.Text) >= Convert.ToDouble(lblTotal.Text))  
    {  
        double kembalian = Convert.ToDouble(txtotalbayar.Text) -  
        Convert.ToDouble(lblTotal.Text);  
        lblkembalian.Text = kembalian.ToString();  
    }  
    else  
    {  
        MessageBox.Show("Uang tidak cukup!", "Error", MessageBoxButtons.OK,  
        MessageBoxIcon.Error);  
    }  
}
```

Penjelasan:

- Untuk Memudahkan saat membayar

- Sistem menghitung dan menampilkan total biaya yang harus dibayar.
- Sistem menampilkan hasil kembalian (misalnya, Rp 15.300).

6.Menu Form Login

```

private void btnlogin_Click(object sender, EventArgs e)
{
    //Username dan password yang benar
    string username = "iqbal";
    string password = "12345";

    //Mengambil input dari user
    string usernameInput = txtusername.Text;
    string passwordInput = txtpassword.Text;

    //Validasi login
    if (usernameInput == username && passwordInput == password)
    {
        //Jika login berhasil
        MessageBox.Show("login berhasil!", "Informasi", MessageBoxButtons.OK,
MessageBoxIcon.Information);

        //Membuka form 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:

- Validasi input nama sebelum login
- Menggunakan Trim() untuk menghapus spasi
- Menyimpan nama ke property UserName
- Return DialogResult.OK untuk indikasi sukses

7. Model Classes

csharp

```

// Model untuk menu item

public class MenuItem

{
    public string Name { get; set; }

    public string Category { get; set; }

    public string Description { get; set; }

    public decimal Price { get; set; }

    public string ImagePath { get; set; }

```

```
}
```

```
// Model untuk cart item

public class CartItem

{
    public MenuItem MenuItem { get; set; }

    public int Quantity { get; set; }

    public decimal Subtotal => MenuItem.Price * Quantity;
}
```

Penjelasan:

- MenuItem menyimpan data menu (nama, kategori, harga)
- CartItem menyimpan item pesanan dengan quantity
- Subtotal dihitung otomatis menggunakan property expression