

LAPORAN PROYEK AKHIR SEMESTER GASAL
JURUSAN PENGEMBANGAN PERANGKAT LUNAK DAN GIM
SMK NEGERI 1 KANDEMAN
TAHUN PELAJARAN 2025/2026



Disusun oleh:

Nama : Abimanyu Octa A.S
Kelas : X PPLG 2
NIS : 258743

Judul Proyek

“Aplikasi Pengelolaan Kasir Masakan Padang”

PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENDIDIKAN DAN KEBUDAYAAN
TAHUN 2025

DAFTAR ISI

DAFTAR ISI.....	1
BAB I.....	2
PENDAHULUAN.....	2
A. Deskripsi Proyek.....	2
B. Ringkasan Teori C# dan Materi yang Digunakan.....	3
C. Perancangan Program.....	8
BAB II.....	10
IMPLEMENTASI.....	10
A. Potongan Kode Program Utama.....	10
B. Jelaskan Fungsi Masing-Masing Bagian.....	11
C. Hasil dan Pembahasan.....	12
D. Tampilan Output Program.....	12
E. Hasil Uji Coba.....	13
BAB III.....	14
LAMPIRAN PROGRAM.....	14
A. Kode Lengkap.....	14
B. Screenshot Program.....	15

BAB I

PENDAHULUAN

A. Deskripsi Proyek

Aplikasi pengelolaan kasir masakan padang adalah sebuah aplikasi yang dirancang untuk membantu proses pencatatan transaksi pada rumah makan atau warung masakan padang. Aplikasi ini berfungsi untuk mempermudah kasir dalam melakukan input pesanan, membatalkan pesanan, dan menghitung total harga.



Gambar 1.1 Logo Aplikasi Pengelolaan Kasir Masakan Padang

Dengan adanya aplikasi ini, proses pencatatan menjadi lebih cepat, akurat, dan mengurangi kesalahan manual seperti salah hitung atau kehilangan data. Aplikasi dibuat menggunakan bahasa pemrograman *C#* dan antarmuka berbasis *Windows Form Application*, sehingga mudah digunakan oleh pemula maupun pengguna umum.

B. Ringkasan Teori C# dan Materi yang Digunakan

1. Variabel dan Tipe Data

Variabel adalah wadah atau penyimpanan data pada program yang akan digunakan selama program itu berjalan. Tipe data adalah jenis-jenis data yang akan disimpan di dalam variabel. Seperti data teks, angka, huruf dan lainnya. Pada proyek ini, tipe data yang saya gunakan antara lain:

a. String

Tipe data *string* adalah jenis data yang digunakan untuk menyimpan teks. Pada proyek ini tipe data *string* digunakan untuk menyimpan nama makanan dan nama minuman. Contoh: “*Saya hebat*”;

```
string nama_makanan = comboBox1.SelectedItem?.ToString();  
string nama_minuman = comboBox2.SelectedItem?.ToString();
```

b. Int

Int adalah tipe data yang berupa angka. Pada proyek ini tipe data *int* digunakan untuk menyimpan porsi, harga (makanan atau minuman) dan total harga. Contoh: 3,9,0;

```
int porsi_mamam = 0;
int porsi_mimik = 0;
```

c. Double

Double adalah tipe data yang berupa desimal. Pada proyek ini tipe data *double* digunakan untuk menyimpan kembalian. Contoh: 3,14;

```
double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
lblkembalian.Text = $"{kembalian.ToString("N0")}";
```

d. Float

Float adalah tipe data yang berupa angka pecahan. Pada proyek ini tipe data *float* tidak digunakan. Contoh: 1.2f, 2.4f, 5.5f (*huruff artinya float*)

e. Bool

Bool adalah tipe data boolean yang hanya berisi true dan false. Pada proyek ini tipe data *bool* tidak digunakan.

2. Operator

Operator adalah simbol atau tanda khusus yang digunakan untuk melakukan operasi terhadap nilai (data) dalam pemrograman. Operator digunakan untuk melakukan operasi pada variabel dan nilai.

a. Operator Aritmatika

1. + : Penjumlahan
2. - : Pengurangan
3. * : Perkalian
4. / : Pembagian
5. % : Sisa bagi (modulus)

b. Operator Perbandingan dalam C#

Operator perbandingan digunakan untuk membandingkan dua nilai dan hasilnya berupa boolean (true atau false). Jenis-jenis Operator Perbandingan di C#:

Operator	Arti	Contoh
----------	------	--------

==	Sama dengan	5 == 5
!=	Tidak sama dengan	5 != 3
>	Lebih besar	10 > 7
<	Lebih kecil	3 < 8
>=	Lebih besar atau sama dengan	6 >= 6
<=	Lebih kecil atau sama dengan	4 <= 9

c. Operator Increment & Decrement

1. ++ : Menambah nilai 1 variabel
2. -- : Mengurangi nilai 1 variabel

d. Operator Logika

Operator logika digunakan untuk menghubungkan **dua atau lebih ekspresi kondisi**(yang menghasilkan true atau false) dan mengembalikan hasil tanpa berupa **boolean**(true atau false).

Nama Operator	Simbol
Logika AND	&&
Logika OR	
Negasi/kebalikan	!

e. Operator Penugasan

Operator penugasan(Assignment Operator) merupakan operator untuk menambahkan tugas pada variabel. Biasanya digunakan untuk **mengisi nilai**.

Nama Operator	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=

Pengisian dan Pengurangan	<code>_=</code>
Pengisian dan Perkalian	<code>*=</code>
Pengisian dan Pembagian	<code>/=</code>
Pengisian dan Sisa bagi	<code>%=</code>

3. Percabangan Kondisi

Percabangan kondisi dalam pemrograman adalah konsep yang sama. Ini memungkinkan program kita untuk membuat keputusan dan menjalankan blok kode tertentu hanya jika sebuah kondisi terpenuhi.

Di C#, percabangan kondisi utama meliputi `if`, `if-else`, `if-else if`, `nested if` (if bersarang), dan `switch` untuk mengontrol alur program berdasarkan suatu kondisi. Pernyataan Pernyataan ini memungkinkan program untuk mengeksekusi blok kode yang berbeda tergantung pada hasil evaluasi suatu kondisi, baik itu `true` (benar) atau `false` (salah).

a. Pernyataan `if`

`if` digunakan untuk menjalankan sebuah blok kode hanya jika kondisinya bernilai `true` (benar).

```
if(nama_makanan == "Rendang")
{
    harga_makanan = 20000 * porsi_mamam;
}
if(nama_minuman == "Es Teh")
{
    harga_minuman = 3000 * porsi_mimik;
}
if (nama_makanan == "Ayam Gulai")
{
    harga_makanan = 18000 * porsi_mamam;
}
if (nama_minuman == "Es Jeruk")
{
    harga_minuman = 4000 * porsi_mimik;
}
if (nama_makanan == "Telur Balado")
{
    harga_makanan = 13000 * porsi_mamam;
}
if (nama_minuman == "Teh Anget")
{
    harga_minuman = 3000 * porsi_mimik;
}
if (nama_makanan == "Gulai Tunjang")
{
    harga_makanan = 15000 * porsi_mamam;
}
if (nama_minuman == "Es Susu")
{
    harga_minuman = 4000 * porsi_mimik;
}
if (nama_makanan == "Ayam Pop")
{
    harga_makanan = 18000 * porsi_mamam;
}
if (nama_minuman == "Kopi Susu")
{
    harga_minuman = 5000 * porsi_mimik;
}
```

b. Pernyataan if-else

if-else digunakan ketika kita memiliki dua pilihan. Jika kondisi if benar, jalankan blok if. Jika tidak (else), jalankan blok else.

```
//Menghitung kembalian
if (!string.IsNullOrEmpty(txtBayar.Text))
{
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
    {
        MessageBox.Show("Uangnya kurang kak");
        return;
    }

    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
    lblkembalian.Text = $"{kembalian.ToString("N0")}";
}
else MessageBox.Show("Total bayar tidak dapat terdefiniskan!");
return;
```

c. Pernyataan if-else if-else if-else

if-else digunakan ketika kita memiliki lebih dari dua pilihan atau kondisi yang perlu diperiksa secara berurutan. Program akan memeriksa kondisi satu per satu dari atas ke bawah, dan menjalankan blok kode pertama yang kondisinya benar.

d. Pernyataan switch-case

switch-case adalah alternatif dari if-else if-else yang lebih bersih dan mudah dibaca, terutama ketika kita ingin membandingkan satu variabel dengan banyak nilai yang pasti.

4. Array C#

Array adalah tipe data yang berisi kumpulan data dengan tipe serupa. Tipe data ini sangat berguna dalam pengelolaan data karena mengelompokkan data-data berdasarkan kesamaannya. Dengan menggunakan array, kita bisa menyimpan banyak data dalam satu variabel yang sama dan dalam satu tipe data yang sama juga. Array mempermudah penghitungan dan pengolahan data dalam berbagai situasi.

Array digambarkan dalam sebuah tabel terdiri dari : value/nilai dan index/indeks

Value / nilai: isi dari array

Index : urutan array (dimulai dari 0)

30	3	22	9	21	26	→ Value / Nilai
0	1	2	3	4	5	→ Indeks

5. Perulangan

Perulangan (Looping) adalah salah satu struktur kontrol yang memungkinkan serangkaian instruksi diulang secara terus-menerus selama kondisi tertentu terpenuhi.

a. Perulangan for

Digunakan ketika jumlah perulangan sudah diketahui di awal.

b. Perulangan while

Digunakan ketika perulangan harus berlanjut selama suatu kondisi benar. Kondisi diperiksa sebelum blok kode dieksekusi.

c. Perulangan do-while

Mirip dengan while, tetapi blok kode dieksekusi minimal satu kali sebelum kondisi diperiksa.

d. Perulangan foreach

Digunakan untuk mengiterasi elemen dalam suatu koleksi (seperti array atau list) tanpa perlu menggunakan indeks.

e. Nested Loop (perulangan bersarang)

Nested Loop (atau perulangan bersarang) adalah perulangan di dalam perulangan lain. Artinya, satu loop berada di dalam loop lainnya, dan loop bagian dalam akan dijalankan sepenuhnya untuk setiap iterasi loop bagian luar.

6. Fungsi

Fungsi (method) dalam C# adalah blok kode yang memiliki nama dan dapat digunakan untuk menjalankan perintah tertentu secara berulang tanpa menulis ulang kode.

Jenis Fungsi	Ciri-Ciri	Contoh
Tanpa parameter dan tanpa return	Hanya menampilkan pesan atau melakukan tugas tanpa hasil balik	void Sapa()

Dengan parameter	Menerima data input dari luar	void SapaNama(string nama)
Dengan return	Menghasilkan nilai dari perhitungan	int Tambah(int a, int b).
Dengan parameter & return	Lengkap, menerima input dan menghasilkan output	double Hitung Luas(double p, double l)

C. Perancangan Program

1. Struktur Program

a. Form Menu Utama

- Tombol “Input Transaksi”
- Tombol “Daftar Menu”
- Tombol “Riwayat Transaksi”

b. Form Transaksi

- ComboBox pilihan menu masakan padang
- TextBox jumlah harga
- Button tambah pesanan
- DataGridView untuk menampilkan daftar pesanan
- Button hitung total harga

2. Rancangan Antarmuka (Mockup)

a. Tampilan Menu Utama

b. Form Transaksi

```
//Menghitung kembalian
if (!string.IsNullOrEmpty(txtBayar.Text))
{
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
    {
        MessageBox.Show("Uangnya kurang kak");
        return;
    }

    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
    lblkembalian.Text = $"{kembalian.ToString("N0")}";
}
else MessageBox.Show("Total bayar tidak dapat terdefiniskan!");
return;
```

BAB II

IMPLEMENTASI

A. Potongan Kode Program Utama

1. Menampilkan Daftar Menu

```
InitializeComponent();
dataGridView1.Rows.Add("Rendang", "20.000", "Es Teh", "3.000");
dataGridView1.Rows.Add("Ayam Gulai", "18.000", "Es Jeruk", "4.000");
dataGridView1.Rows.Add("Telur Balado", "13.000", "Teh Anget", "3.000");
dataGridView1.Rows.Add("Gulai Tunjang", "15.000", "Es Susu", "4.000");
dataGridView1.Rows.Add("Ayam Pop", "18.000", "Kopi Susu", "5.000");
```

2. Menghapus Riwayat Transaksi

```
if (dataGridView2.CurrentRow != null)
{
    dataGridView2.Rows.RemoveAt(dataGridView2.CurrentRow.Index);
}
else
    MessageBox.Show("Pilih baris pesanan yang ingin dihapus");
```

3. Pemilihan Pesanan

```
string nama_makanan = comboBox1.SelectedItem?.ToString();
string nama_minuman = comboBox2.SelectedItem?.ToString();
int porsi_maman = 0;
int porsi_mimik = 0;
if (!string.IsNullOrEmpty(comboBox4.Text))
{
    porsi_maman = Convert.ToInt32(comboBox4.Text);
}
if (!string.IsNullOrEmpty(comboBox3.Text))
{
    porsi_mimik = Convert.ToInt32(comboBox3.Text);
}
int harga_makanan = 0;
int harga_minuman = 0;
if (nama_makanan == "Rendang")
{
    harga_makanan = 20000 * porsi_maman;
}
if (nama_minuman == "Es Teh")
{
    harga_minuman = 3000 * porsi_mimik;
}
if (nama_makanan == "Ayam Gulai")
{
    harga_makanan = 18000 * porsi_maman;
}
if (nama_minuman == "Es Jeruk")
{
    harga_minuman = 4000 * porsi_mimik;
}
if (nama_makanan == "Telur Balado")
{
    harga_makanan = 13000 * porsi_maman;
}
if (nama_minuman == "Teh Anget")
{
    harga_minuman = 3000 * porsi_mimik;
}
if (nama_makanan == "Gulai Tunjang")
{
    harga_makanan = 15000 * porsi_maman;
}
if (nama_minuman == "Es Susu")
{
    harga_minuman = 4000 * porsi_mimik;
}
if (nama_makanan == "Ayam Pop")
{
    harga_makanan = 18000 * porsi_maman;
}
if (nama_minuman == "Kopi Susu")
{
    harga_minuman = 5000 * porsi_mimik;
}
```

4. Menghitung Total Harga

```
int total_harga = harga_minuman + harga_makanan;  
dataGridView2.Rows.Add(nama_makanan, porsi_naman, nama_minuman, porsi_mimik, total_harga);  
txtTotal.Text = $"{Convert.ToInt32(txtTotal.Text) + total_harga}";
```

5. Menghitung Kembalian

```
//Menghitung kembalian  
if (!string.IsNullOrEmpty(txtBayar.Text))  
{  
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))  
    {  
        MessageBox.Show("Uangnya kurang kak");  
        return;  
    }  
  
    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);  
    lblkembalian.Text = $"{kembalian.ToString("N0")}";  
}  
else MessageBox.Show("Total bayar tidak dapat terdefiniskan!");  
return;
```

6. Menghapus Pesanan

```
comboBox1.SelectedIndex = -1;  
comboBox2.SelectedIndex = -1;  
comboBox3.SelectedIndex = -1;  
comboBox4.SelectedIndex = -1;
```

B. Jelaskan Fungsi Masing-Masing Bagian

1. Fungsi Halaman Menu

Halaman menu pada aplikasi kasir berfungsi sebagai halaman utama yang menjadi pusat akses ke semua fitur yang dibutuhkan dalam proses operasional toko. Menu ini mempermudah kasir, admin, atau pemilik toko untuk berpindah ke fitur lain dengan cepat tanpa kebingungan.

2. Fungsi Menghapus Riwayat Transaksi

Fitur Menghapus Riwayat Transaksi berfungsi untuk menghapus data transaksi tertentu dari sistem kasir. Biasanya fitur ini hanya dapat diakses oleh admin atau pemilik toko, bukan kasir biasa, karena berkaitan dengan data penjualan.

3. Fungsi Pemilihan Pesanan

Pemilihan Pesanan adalah fitur yang memungkinkan kasir atau pengguna untuk memilih item/barang/jasa yang ingin dibeli pelanggan saat proses transaksi. Fitur ini biasanya muncul di halaman penjualan atau pemesanan.

4. Fungsi Menghitung Total Harga

Fitur Menghitung Total Harga digunakan untuk menjumlahkan seluruh biaya yang harus dibayar pelanggan berdasarkan barang atau jasa yang dipilih dalam transaksi. Fitur ini sangat penting karena menentukan jumlah akhir yang harus dibayar.

5. Fungsi Menghitung Kembalian

Fitur Menghitung Kembalian digunakan untuk menentukan berapa uang kembalian yang harus diberikan kepada pelanggan setelah mereka membayar transaksi. Fitur ini penting untuk memastikan pembayaran dilakukan dengan akurat, cepat, dan tanpa kesalahan.

C. Hasil dan Pembahasan

Aplikasi “Kasir Masakan Padang” berhasil berjalan dengan baik. Fitur yang dapat digunakan:

1. Memilih menu dari daftar yang disediakan
2. Menambahkan pesanan lebih dari satu menu
3. Menampilkan riwayat transaksi
4. Menghitung total pembayaran secara otomatis
5. Menghitung kembalian

D. Tampilan Output Program

MENU MASAKAN PADANG

Masakan

Porsi

Minuman

Porsi

PESAN **BATALKAN** **BERSIHKAN**

Makanan	Porsi	Minuman	Porsi	Total Harga
---------	-------	---------	-------	-------------

Pembayaran

Total Harga

Jumlah Uang Pelanggan

Bayar **Uang Kembalian :**

Makanan	Harga Makanan	Minuman	Harga Minuman
---------	---------------	---------	---------------

E. Hasil Uji Coba

Uji coba dilakukan untuk memastikan semua fitur utama aplikasi kasir berjalan dengan baik. Hasilnya menunjukkan bahwa aplikasi stabil dan sesuai fungsinya:

- Halaman Menu: Berfungsi sebagai pusat navigasi, semua tombol mengarah ke halaman yang tepat tanpa error.
- Pemilihan Pesanan: Kasir dapat memilih produk, mengatur jumlah barang, dan melihat detail produk dengan akurat.
- Menghitung Total Harga: Sistem otomatis menjumlahkan semua harga, termasuk diskon dan pajak, sehingga perhitungan tepat dan cepat.
- Menghitung Kembalian: Kembalian dihitung otomatis berdasarkan uang bayar, mengurangi kesalahan kasir dan mempermudah transaksi.
- Menghapus Riwayat Transaksi: Admin dapat menghapus transaksi yang salah, data riwayat dan laporan penjualan menyesuaikan otomatis.

BAB III

LAMPIRAN PROGRAM

A. Kode Lengkap

```
InitializeComponent();
dataGridView1.Rows.Add("Rendang", "20.000", "Es Teh", "3.000");
dataGridView1.Rows.Add("Ayam Gulai", "18.000", "Es Jeruk", "4.000");
dataGridView1.Rows.Add("Telur Balado", "13.000", "Teh Anget", "3.000");
dataGridView1.Rows.Add("Gulai Tunjang", "15.000", "Es Susu", "4.000");
dataGridView1.Rows.Add("Ayam Pop", "18.000", "Kopi Susu", "5.000");
```

```
if (dataGridView2.CurrentRow != null)
{
    dataGridView2.Rows.RemoveAt(dataGridView2.CurrentRow.Index);
}
else
    MessageBox.Show("Pilih baris pesanan yang ingin dihapus");
```

```
string nama_makanan = comboBox1.SelectedItem?.ToString();
string nama_minuman = comboBox2.SelectedItem?.ToString();
int porsi_maman = 0;
int porsi_misik = 0;
if (!string.IsNullOrEmpty(comboBox4.Text))
{
    porsi_maman = Convert.ToInt32(comboBox4.Text);
}
if (!string.IsNullOrEmpty(comboBox3.Text))
{
    porsi_misik = Convert.ToInt32(comboBox3.Text);
}
int harga_makanan = 0;
int harga_minuman = 0;
if (nama_makanan == "Rendang")
{
    harga_makanan = 20000 * porsi_maman;
}
if (nama_minuman == "Es Teh")
{
    harga_minuman = 3000 * porsi_misik;
}
if (nama_makanan == "Ayam Gulai")
{
    harga_makanan = 18000 * porsi_maman;
}
if (nama_minuman == "Es Jeruk")
{
    harga_minuman = 4000 * porsi_misik;
}
if (nama_makanan == "Telur Balado")
{
    harga_makanan = 13000 * porsi_maman;
}
if (nama_minuman == "Teh Anget")
{
    harga_minuman = 3000 * porsi_misik;
}
if (nama_makanan == "Gulai Tunjang")
{
    harga_makanan = 15000 * porsi_maman;
}
if (nama_minuman == "Es Susu")
{
    harga_minuman = 4000 * porsi_misik;
}
if (nama_makanan == "Ayam Pop")
{
    harga_makanan = 18000 * porsi_maman;
}
if (nama_minuman == "Kopi Susu")
{
    harga_minuman = 5000 * porsi_misik;
}
```

```
int total_harga = harga_minuman + harga_makanan;
dataGridView2.Rows.Add(nama_makanan, porsi_maman, nama_minuman, porsi_misik, total_harga);
txtTotal.Text = $"{Convert.ToInt32(txtTotal.Text) + total_harga}";
```

```
//Menghitung kembalian
if (!string.IsNullOrEmpty(txtBayar.Text))
{
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
    {
        MessageBox.Show("Uangnya kurang kak");
        return;
    }

    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
    lblkembalian.Text = $"{kembalian.ToString("N0")}";
}
else MessageBox.Show("Total bayar tidak dapat terdefiniskan!");
return;
```

```
comboBox1.SelectedIndex = -1;
comboBox2.SelectedIndex = -1;
comboBox3.SelectedIndex = -1;
comboBox4.SelectedIndex = -1;
```

B. Screenshot Program

The screenshot shows a Windows application window titled "PESAN" with a subtitle "MENU MASAKAN PADANG". The interface is divided into several sections:

- Order Form:** Located on the left, it includes four dropdown menus labeled "Masakan", "Porsi", "Minuman", and "Porsi", each followed by a small input field. Below these are three buttons: "PESAN" (green), "BATALKAN" (red), and "BERSIHKAN" (red).
- Menu Table:** A table with four columns: "Makanan", "Harga Makanan", "Minuman", and "Harga Minuman". The table body is currently empty.
- Payment Section:** Located on the right, it includes a "Pembayaran" header, a "Total Harga" label with a text box containing "0", a "Jumlah Uang Pelanggan" label with an empty text box, a "Bayar" button (green), and a label "Uang Kembalian :".