

PROJEK AKHIR SEMESTER GASAL



Disusun oleh:

Nama : Ummi Salamah
Kelas : X PPLG 2
No.Absen : 33
NIS : 258775
Judul Projek : Aplikasi Pemesanan Kebab

**PENGEMBANGAN PERANGKAT LUNAK DAN GIM
SMK NEGERI 1 KANDEMAN
TAHUN PELAJARAN 2025/2026**

DAFTAR ISI

1. DESKRIPSI PROYEK.....	2
Fitur Utama:.....	2
2. RINGKASAN TEORI C# YANG DIGUNAKAN.....	3
A. Variabel dan Tipe Data.....	3
B. Operator.....	3
C. Percabangan.....	3
D. Perulangan.....	4
E. Array atau List.....	4
F. Fungsi atau Method.....	4
G. Input dan Output.....	5
3. PERANCANGAN PROGRAM.....	5
A. Struktur Alur Program.....	5
B. Daftar Menu dan Harga.....	6
4. IMPLEMENTASI PROGRAM.....	6
Komponen Form.....	6
5. PENJELASAN KODE PROGRAM.....	7
A. Event Handler Pemilihan Menu (comboBox1_SelectedIndexChanged).....	7
B. Fungsi Perhitungan (btnHitung_Click).....	7
C. Fungsi Reset Form (btnHapus_Click).....	9
D. Fungsi Keluar Aplikasi (btnKeluar_Click).....	9
6. KONSEP PEMROGRAMAN YANG DIIMPLEMENTASIKAN.....	10
A. Exception Handling (Try-Catch).....	10
B. Type Conversion.....	10
C. Event-Driven Programming.....	10
D. User Input Validation.....	10
7. KELEBIHAN DAN KEKURANGAN PROGRAM.....	10
A. Kelebihan:.....	10
B. Kekurangan:.....	10
8. SARAN PENGEMBANGAN.....	10
9. KESIMPULAN.....	11

1. DESKRIPSI PROYEK

Proyek ini merupakan aplikasi desktop pemesanan kebab yang dikembangkan menggunakan bahasa pemrograman C# dengan Windows Forms. Aplikasi ini dirancang untuk memudahkan proses transaksi pembelian kebab dengan berbagai varian menu. Sistem ini menyediakan fitur pemilihan menu, perhitungan otomatis harga total, validasi pembayaran, dan perhitungan kembalian.

Fitur Utama:

- a. Pemilihan menu kebab melalui ComboBox
- b. Tampilan harga otomatis berdasarkan pilihan menu
- c. Input jumlah pembelian
- d. Perhitungan total harga otomatis
- e. Validasi pembayaran (cek kecukupan uang)
- f. Perhitungan kembalian
- g. Tombol reset untuk transaksi baru
- h. Tombol keluar aplikasi

2. RINGKASAN TEORI C# YANG DIGUNAKAN

A. Variabel dan Tipe Data

Program ini menggunakan beberapa tipe data dasar:

- 1) Integer (bilangan bulat):

```
```csharp
int harga = int.Parse(txtHarga.Text);
int jumlah = int.Parse(txtJumlahBeli.Text);
int total = harga * jumlah;
int bayar = int.Parse(txtBayar.Text);
int kembalian = bayar - total;
```
```

Variabel integer digunakan untuk menyimpan nilai numerik seperti harga, jumlah beli, total pembayaran, dan kembalian.

- 2) String (teks):

```
```csharp
cmbKebab.Text // Mengambil teks dari ComboBox
txtHarga.Text = "15000"; // Menyimpan nilai sebagai string
```
```

B. Operator

- 1) Operator Aritmatika:

- a) Perkalian: `total = harga * jumlah`
- b) Pengurangan: `kembalian = bayar - total`

- 2) Operator Relasional:

Kurang dari: `if (kembalian < 0)` untuk validasi kecukupan pembayaran

- 3) Operator Assignment:

Sama dengan (=): Untuk memberikan nilai ke variabel.

C. Percabangan

- 1) SWITCH-CASE:

```
```csharp
switch (cmbKebab.Text)
{
 case "Kebab Ayam":
 txtHarga.Text = "15000";
 break;
 case "Kebab Sapi":
 txtHarga.Text = "20000";
 break;
 case "Kebab Spesial":
 txtHarga.Text = "25000";
 break;
 case "Kebab Sayur":
 txtHarga.Text = "13000";
 break;
 case "Kebab Keju Mozarella":
 txtHarga.Text = "22000";
 break;
}
````
```

Switch-case digunakan untuk menentukan harga berdasarkan pilihan menu kebab.

- 2) IF-ELSE:

```
```csharp
if (kembalian < 0)
{
 MessageBox.Show("UANG BAYAR KURANG!!", "PERINGATAN");
}
else
{
 MessageBox.Show("Terima Kasih Sudah Memesan", "Sukses");
}
````
```

Percabangan IF-ELSE digunakan untuk validasi apakah uang pembayaran mencukupi.

D. Perulangan

Program ini tidak menggunakan perulangan secara eksplisit karena fokus pada transaksi per item. Namun, konsep perulangan implisit terjadi ketika user dapat melakukan transaksi berulang kali dengan menggunakan tombol reset.

E. Array atau List

Program ini tidak mengimplementasikan array atau list secara langsung. Data menu dan harga disimpan dalam struktur switch-case. Untuk pengembangan lebih lanjut, bisa digunakan array atau dictionary untuk menyimpan data menu.

Contoh pengembangan dengan array:

```
```csharp
// Tidak ada dalam kode saat ini, tapi bisa dikembangkan:
// string[] menuKebab = {"Kebab Ayam", "Kebab Sapi", "Kebab Spesial"};
// int[] hargaKebab = {15000, 20000, 25000};
````
```

F. Fungsi atau Method

Program memiliki beberapa method event handler:

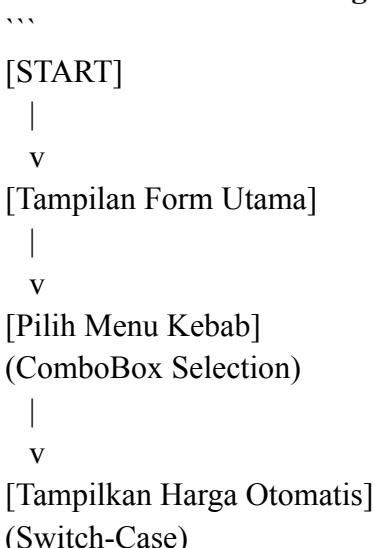
- 1) btnHitung_Click() : Method untuk menghitung total dan kembalian
- 2) btnHapus_Click() : Method untuk mereset form
- 3) btnKeluar_Click() : Method untuk keluar aplikasi
- 4) comboBox1_SelectedIndexChanged() : Method untuk menampilkan harga saat menu dipilih

G. Input dan Output

- 1) Input
 - a) ComboBox: `cmbKebab` (pemilihan menu)
 - b) TextBox: `txtJumlahBeli` (jumlah pesanan)
 - c) TextBox: `txtBayar` (jumlah uang pembayaran)
- 2) Output
 - a) TextBox: `txtHarga` (harga per item)
 - b) TextBox: `txtTotal` (total pembayaran)
 - c) TextBox: `txtKembalian` (kembalian)
 - d) MessageBox: Notifikasi sukses atau error

3. PERANCANGAN PROGRAM

A. Struktur Alur Program



```

|  

v  

[Input Jumlah Beli]  

|  

v  

[Input Jumlah Bayar]  

|  

v  

[Klik Tombol Hitung]  

|  

v  

[Proses Perhitungan]  

- Total = Harga × Jumlah  

- Kembalian = Bayar - Total  

|  

v  

[Validasi Pembayaran]  

|  

+---> [Kembalian < 0] --> [Peringatan "Uang Kurang"]  

|  

+---> [Kembalian >= 0] --> [Sukses "Terima Kasih"]  

|  

v  

[Tombol Hapus] --> [Reset Form]  

|  

v  

[Tombol Keluar] --> [Exit Program]  

|  

v  

[END]
...

```

B. Daftar Menu dan Harga

| Menu Kebab | Harga |
|----------------------|-----------|
| Kebab sayur | Rp 13.000 |
| Kebab ayam | Rp 15.000 |
| Kebab sapi | Rp 20.000 |
| Kebab keju mozarella | Rp 22.000 |
| Kebab spesial | Rp 25.000 |

4. IMPLEMENTASI PROGRAM

Komponen Form

A. Input Controls

- a) ComboBox: cmbKebab (untuk memilih jenis kebab)
- b) TextBox: txtJumlahBeli (input jumlah pesanan)
- c) TextBox: txtBayar (input jumlah uang pembayaran)

B. Output Controls

- a) TextBox: txtHarga (menampilkan harga satuan, read-only)
- b) TextBox: txtTotal (menampilkan total harga, read-only)
- c) TextBox: txtKembalian (menampilkan kembalian, read-only)

C. Button Controls

- a) btnHitung: Tombol untuk menghitung total dan kembalian
- b) btnHapus: Tombol untuk mereset/membersihkan form
- c) btnKeluar: Tombol untuk keluar dari aplikasi

D. Visual Elements

- a) PictureBox: pictureBox1 (untuk menampilkan gambar/logo)
- b) Label: Untuk keterangan field input/output

5. PENJELASAN KODE PROGRAM

A. Event Handler Pemilihan Menu (comboBox1_SelectedIndexChanged)

```
```csharp
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
 switch (cmbKebab.Text)
 {
 case "Kebab Ayam":
 txtHarga.Text = "15000";
 break;
 // ... kasus lainnya
 }
}
````
```

a) Penjelasan:

Method ini dipanggil otomatis ketika user memilih item dari ComboBox. Menggunakan struktur ****SWITCH-CASE**** untuk menentukan harga berdasarkan menu yang dipilih. Setiap case mewakili satu jenis kebab dengan harga yang berbeda. Statement “break” digunakan untuk keluar dari switch setelah menemukan kecocokan, mencegah eksekusi case berikutnya.

b) Proses

1. Membaca teks yang dipilih dari ComboBox (cmbKebab.Text)
2. Membandingkan dengan setiap case
3. Jika cocok, set harga ke TextBox harga
4. Break untuk menghentikan pengecekan lebih lanjut

B. Fungsi Perhitungan (btnHitung_Click)

```
```csharp
private void btnHitung_Click(object sender, EventArgs e)
{
 try
 {
 int harga = int.Parse(txtHarga.Text);
 int jumlah = int.Parse(txtJumlahBeli.Text);
 int total = harga * jumlah;
 txtTotal.Text = total.ToString();

 int bayar = int.Parse(txtBayar.Text);
 int kembalian = bayar - total;
 txtKembalian.Text = kembalian.ToString();

 if (kembalian < 0)
 {
 MessageBox.Show("UANG BAYAR KURANG!!", "PERINGATAN");
 }
 else
 {
 MessageBox.Show("Terima Kasih Sudah Memesan", "Sukses");
 }
 }
 catch
 {
 MessageBox.Show("Pastikan semua data sudah terisi!!", "Error");
 }
}
```
```

```

a) Penjelasan:

Method ini merupakan inti dari proses perhitungan transaksi yang menggunakan blok “TRY-CATCH” untuk menangani error.

b) Langkah-langkah:

1. Parsing Data: Mengkonversi string dari TextBox menjadi integer menggunakan `int.Parse()`
2. Perhitungan Total: Menggunakan operator perkalian (\*) untuk menghitung total harga.  
Rumus: `Total = Harga × Jumlah Beli`
3. Perhitungan Kembalian: Menggunakan operator pengurangan (-).  
Rumus: `Kembalian = Bayar - Total`
  
4. Validasi Pembayaran: Menggunakan percabangan IF-ELSE
  - 1) Jika kembalian < 0: Pembayaran kurang, tampilkan peringatan
  - 2) Jika kembalian >= 0: Pembayaran cukup, tampilkan terima kasih
5. Error Handling: Blok catch menangkap exception jika ada field kosong atau format data salah

### C. Fungsi Reset Form (btnHapus\_Click)

```
```csharp
private void btnHapus_Click(object sender, EventArgs e)
{
    txtHarga.Clear();
    txtJumlahBeli.Clear();
    txtBayar.Clear();
    txtTotal.Clear();
    txtKembalian.Clear();
    cmbKebab.SelectedIndex = -1;
    cmbKebab.Focus();
}
```

```

- a) Penjelasan:  
Method ini berfungsi untuk membersihkan form dan mempersiapkan transaksi baru.
- b) Proses:
  1. Clear() Method: Membersihkan semua TextBox dengan method Clear()
  2. Reset ComboBox: Mengatur SelectedIndex = -1 agar tidak ada item yang terpilih
  3. Set Focus: Mengarahkan kursor ke ComboBox menggunakan Focus() untuk kemudahan user memulai transaksi baru

### D. Fungsi Keluar Aplikasi (btnKeluar\_Click)

```
```csharp
private void btnKeluar_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

```

```
}
```

```
...
```

Penjelasan:

Method sederhana yang menutup aplikasi menggunakan `Application.Exit()`. Method ini menghentikan semua thread yang berjalan dan menutup form aplikasi dengan aman.

## 6. KONSEP PEMROGRAMAN YANG DIIMPLEMENTASIKAN

### A. Exception Handling (Try-Catch)

Program menggunakan try-catch untuk menangani kemungkinan error saat parsing data. Ini mencegah aplikasi crash ketika user memasukkan data yang tidak valid atau membiarkan field kosong.

### B. Type Conversion

Konversi tipe data dari string ke integer menggunakan `int.Parse()` dan sebaliknya menggunakan `ToString()`.

### C. Event-Driven Programming

Program menggunakan paradigma event-driven dimana aksi user (klik tombol, pilih item) memicu eksekusi method tertentu.

### D. User Input Validation

Validasi dilakukan pada saat perhitungan untuk memastikan pembayaran mencukupi dengan membandingkan nilai kembalian dengan 0.

## 7. KELEBIHAN DAN KEKURANGAN PROGRAM

### A. Kelebihan:

- |                        |   |                                                  |
|------------------------|---|--------------------------------------------------|
| a) Interface Sederhana | : | Mudah dipahami dan digunakan                     |
| b) Validasi Error      | : | Try-catch mencegah crash aplikasi                |
| c) User Friendly       | : | Harga otomatis muncul saat memilih menu          |
| d) Feedback Jelas      | : | MessageBox memberikan informasi status transaksi |
| e) Fitur Reset         | : | Memudahkan transaksi berulang                    |

### B. Kekurangan:

- |                         |   |                                                                 |
|-------------------------|---|-----------------------------------------------------------------|
| a) Tidak Ada Array/List | : | Data menu hardcoded dalam switch-case                           |
| b) Tidak Ada Pencatatan | : | Tidak menyimpan riwayat transaksi                               |
| c) Tidak Ada Struk      | : | Tidak ada output struk pembelian                                |
| d) Validasi Terbatas    | : | Tidak ada validasi untuk input negatif atau non-numerik di awal |
| e) Event Handler Kosong | : | Banyak method kosong yang tidak terpakai                        |

## 8. SARAN PENGEMBANGAN

- a. Implementasi Array/Dictionary: Gunakan struktur data untuk menyimpan menu dan harga

```csharp

```
Dictionary<string, int> menuKebab = new Dictionary<string, int>()
{
    {"Kebab Ayam", 15000},
    {"Kebab Sapi", 20000},
    {"Kebab Spesial", 25000}
};
```

```

- b. Database Integration : Simpan data transaksi ke database
- c. Cetak Struk : Tambahkan fitur untuk mencetak atau menyimpan struk
- d. Validasi Input Real-time : Cegah input selain angka pada TextBox jumlah dan bayar
- e. Sistem Diskon : Tambahkan fitur diskon untuk pembelian dalam jumlah tertentu
- f. Laporan Penjualan : Tambahkan fitur untuk melihat total penjualan harian

## KESIMPULAN

Program aplikasi pemesanan kebab ini berhasil mengimplementasikan konsep-konsep dasar pemrograman C# meliputi variabel dan tipe data integer dan string, operator aritmatika untuk perhitungan, percabangan switch-case dan if-else, input/output melalui Windows Forms controls, exception handling dengan try-catch, dan event-driven programming.

Aplikasi ini memberikan solusi sederhana namun efektif untuk proses transaksi pemesanan kebab dengan fitur perhitungan otomatis dan validasi pembayaran. Meskipun masih memiliki ruang untuk pengembangan seperti implementasi array, database, dan fitur tambahan lainnya, program ini sudah memenuhi kebutuhan dasar transaksi penjualan dengan interface yang user-friendly dan logika yang jelas.