

**PROYEK AKHIR SEMESTER GASAL**



NAMA : Syifa Azalia  
KELAS : X PPLG 1  
NIS : 258735  
JUDUL PROYEK : BAKERY CASHIER

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

## **1. Deskripsi Program**

Program **Bakery Cashier** adalah aplikasi desktop berbasis Windows Forms yang dikembangkan menggunakan bahasa pemrograman C#. Aplikasi ini berfungsi sebagai sistem pemesanan Cake and Drink yang memudahkan pelanggan untuk melakukan pemesanan dengan perhitungan otomatis termasuk discount.

### **Fitur Utama :**

- Pemisah kategori menu (Cake atau Drink)
- Pemilihan item menu berdasarkan kategori
- Perhitungan otomatis total harga
- Sistem diskon (Rp 10.000 untuk pembelian  $\geq$  Rp 50.000)
- Validasi input data
- Tampilan detail pesanan
- Fungsi reset/bersihkan form

### **Tujuan :**

Mempermudah proses pemesanan di BAKERY dengan antarmuka yang user-friendly dan perhitungan yang akurat, serta mengurangi kesalahan manual dalam pencatatan pesanan.

## **2. Ringkasan Teori C# yang digunakan**

### **a. Variabel dan Tipe Data**

Digunakan untuk menyimpan data transaksi seperti Types, Detail Pesanan, dan Harga. Contoh:Desimal,Int, string.

- **Decimal (Tipe data untuk menyimpan angka desimal)**

#### ***CONTOH PENGGUNAAN***

```
decimal hargaSatuan = 0;  
decimal.TryParse(harga.Text, out hargaSatuan);
```

- **Int (Variabel untuk menyimpan bilangan bulat)**

***CONTOH PENGGUNAAN***

```
int jml = 0;
int.TryParse(jumlah.Text, out jml);
```

- **String (Variabel untuk menyimpan teks)**

***CONTOH PENGGUNAAN 1***

```
if (string.IsNullOrEmpty(nm.Text))
{
    MessageBox.Show("Please fill the Name!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
```

```
if (string.IsNullOrEmpty(jm.Text))
{
    MessageBox.Show("Please select Types!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
```

***CONTOH PENGGUNAAN 2***

```
string pesan = "===== DETAIL ORDER =====\n\n" +
$"NAME: {nno.Text}\n" +
$"TYPES: {jm.Text}\n" +
$"MENU: {nm.Text}\n" +
$"PRICE: Rp {harga.Text}\n" +
$"AMOUNT: {jumlah.Text}\n\n" +
"-----\n" +
$"TOTAL PRICE: Rp {totalHarga:N0}\n" +
$"DISCOUNT: Rp {potonganHarga:N0}\n" +
"-----\n" +
$"TOTAL PAYMENT: Rp {totalBayar:N0}";
```

## b. Operator

- **Operator Aritmetika (Menghitung Total dengan - dan \* )**

***CONTOH PENGGUNAAN - dan \****

```
totalBayar = totalHarga - potonganHarga;
```

```
totalHarga = hargaSatuan * jml;
```

**Fungsi:**

\* (perkalian): Menghitung total harga dengan mengalikan harga satuan dengan jumlah

- (pengurangan): Menghitung total bayar setelah dikurangi diskon

- **Operator Perbandingan (Membandingkan dua nilai dengan >= dan ==)**

***CONTOH PENGGUNAAN >= dan ==***

```
(totalHarga >= 50000)
```

```
(jm.Text == "Drink")
```

**Fungsi:**

>= (lebih besar sama dengan): Memeriksa apakah total harga memenuhi syarat untuk mendapat diskon

== (sama dengan): Memeriksa apakah kategori yang dipilih user adalah “Drink”, Jika true maka tampilkan daftar menu minuman.

## c. Percabangan

- **Percabangan If Else if**

***CONTOH PENGGUNAAN***

```
if (jm.Text == "Cake")
```

```
{
```

```
    nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
```

```
    nm.Items.Clear();
```

```
    nm.Text = "";
```

```
    nm.Items.Add("Brulee Cake");
```

```
    nm.Items.Add("Cromboloni");
```

```
    nm.Items.Add("Korean Cream Cake");
```

```
    nm.Items.Add("Mochi Crunch Cake");
```

```

        nm.SelectedIndexChanged += nm_SelectedIndexChanged;
    }

else if (jm.Text == "Drink")
{
    nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
    nm.Items.Clear();
    nm.Text = "";
    nm.Items.Add("Dalgona Coffe");
    nm.Items.Add("Squash Yakult");
    nm.Items.Add("Magic Water");
    nm.Items.Add("Matcha Cream Cloud");
    nm.SelectedIndexChanged += nm_SelectedIndexChanged;
}

```

**Fungsi:**

Memilih menu mana yang akan ditampilkan berdasarkan kategori yang dipilih user

Jika user memilih "Cake", maka daftar menu kue yang akan muncul

Jika user memilih "Drink", maka daftar menu minuman yang akan muncul

- **Percabangan If Else**

***CONTOH PENGGUNAAN***

```

if (totalHarga >= 50000)
    potonganHarga = 10000;
else
    potonganHarga = 0;

```

**Fungsi:**

Menentukan apakah customer dapat diskon Rp 10.000 atau tidak

Syarat: total pembelian minimal Rp 50.000

#### **d. Perulangan**

Tidak Ada Perulangan Eksplisit

Program ini tidak menggunakan perulangan seperti for, While, Do While, atau Foreach Namun terdapat konsep berulang melalui Event-Driven Programming:

**Fungsi:**

Method event handler (seperti jumlah\_TextChanged) akan dipanggil berulang kali setiap kali user melakukan perubahan

Method HitungTotal() dipanggil berkali-kali saat ada perubahan pada input, menciptakan efek "perulangan" dalam merespons aksi user

#### **e. Array atau List**

***CONTOH PENGGUNAAN***

```
jm.Items.Add("Cake");
jm.Items.Add("Drink");
nm.Items.Clear();
nm.Items.Add("Brulee Cake");
```

**Fungsi:**

ComboBox.Items: Merupakan collection (kumpulan) yang berfungsi seperti list untuk menyimpan item-item dalam dropdown

Add(): Menambahkan item baru ke dalam collection

Clear(): Menghapus semua item dari collection

**Penggunaan:**

Menyimpan daftar kategori menu (Cake, Drink)

Menyimpan daftar menu spesifik yang berubah sesuai kategori

Memudahkan pengelolaan item yang ditampilkan di ComboBox

## f. Fungsi atau Method

- Event Handler Method

***CONTOH PENGGUNAAN (BUTTON1 CLICK)***

```
private void button1_Click(object sender, EventArgs e)
{
    // Validasi
    if (string.IsNullOrEmpty(nm.Text))
    {
        MessageBox.Show("Please fill the Name!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jm.Text))
    {
        MessageBox.Show("Please select Types!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(nm.Text))
    {
        MessageBox.Show("Please select Drink/Cake!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jumlah.Text) || jumlah.Text == "0")
    {
        MessageBox.Show("Please fill Amount!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
```

```

// Hitung total
HitungTotal();

// Tampilkan konfirmasi
string pesan = "===== DETAIL ORDER =====\n\n" +
"$NAME: {nno.Text}\n" +
"$TYPES: {jm.Text}\n" +
"$MENU: {nm.Text}\n" +
"$PRICE: Rp {harga.Text}\n" +
"$AMOUNT: {jumlah.Text}\n\n" +
"-----\n" +
$"TOTAL PRICE: Rp {totalHarga:N0}\n" +
$"DISCOUNT: Rp {potonganHarga:N0}\n" +
"-----\n" +
$"TOTAL PAYMENT: Rp {totalBayar:N0}";
```

```

MessageBox.Show(pesan, "Konfirmasi Pesanan",
MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

**Fungsi:**

Dipanggil saat tombol PESAN diklik  
 Melakukan validasi input  
 Menampilkan detail pesanan dalam MessageBox

**g. Input dan Output**

- **Input**

***CONTOH PENGGUNAAN 1***

```
{
  if (nm.Text == "Brulee Cake")
    harga.Text = "25000";
```

```
else if (nm.Text == "Cromboloni")
  harga.Text = "20000";
```

```
else if (nm.Text == "Korean Cream Cake")
```

```

harga.Text = "35000";

else if (nm.Text == "Mochi Crunch Cake")
    harga.Text = "25000";

else if (nm.Text == "Dalgona Coffe")
    harga.Text = "10000";

else if (nm.Text == "Squash Yakult")
    harga.Text = "15000";

else if (nm.Text == "Magic Water")
    harga.Text = "18000";

else if (nm.Text == "Matcha Cream Cloud")
    harga.Text = "14000";
}

```

**Fungsi:**

User memilih kategori atau menu dari dropdown, kemudian program membaca pilihan tersebut melalui property **.Text**

**CONTOH PENGGUNAAN 2**

```

$"NAME: {nno.Text}\n" +
$"TYPES: {jm.Text}\n" +
$"MENU: {nm.Text}\n" +
$"PRICE: Rp {harga.Text}\n" +
$"AMOUNT: {jumlah.Text}\n\n" +
"-----\n" +
$"TOTAL PRICE: Rp {totalHarga:N0}\n" +
$"DISCOUNT: Rp {potonganHarga:N0}\n" +
"-----\n" +
$"TOTAL PAYMENT: Rp {totalBayar:N0}";

```

**Fungsi:**

User mengetik nama dan jumlah di TextBox, program membaca input tersebut untuk diproses

- **Output**

***CONTOH PENGGUNAAN 1***

```
MessageBox.Show("Please fill the Name!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
return;
```

**Fungsi:**

Menampilkan pesan dalam bentuk pop-up dialog untuk memberi informasi atau peringatan kepada user

***CONTOH PENGGUNAAN 2***

```
// Tampilkan konfirmasi
string pesan = "===== DETAIL ORDER =====\n\n" +
    $"NAME: {nno.Text}\n" +
    $"TYPES: {jm.Text}\n" +
    $"MENU: {nm.Text}\n" +
    $"PRICE: Rp {harga.Text}\n" +
    $"AMOUNT: {jumlah.Text}\n\n" +
    "-----\n" +
    $"TOTAL PRICE: Rp {totalHarga:N0}\n" +
    $"DISCOUNT: Rp {potonganHarga:N0}\n" +
    "-----\n" +
    $"TOTAL PAYMENT: Rp {totalBayar:N0}";
```

```
MessageBox.Show(pesan, "Konfirmasi Pesanan",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
```

**Fungsi:**

Menggabungkan beberapa data input dan hasil perhitungan menjadi satu teks lengkap untuk di tampilkan ke user

**h. Componen Windows Form yang digunakan**

- **Form**

**Definisi: Jendela utama aplikasi yang menjadi wadah untuk semua komponen lainnya.**

**Fungsi:**

**Sebagai container/tempat meletakkan semua kontrol**

**Mengatur tampilan dan layout aplikasi**

- **Button**

**Definisi: Jendela utama aplikasi yang menjadi wadah untuk semua komponen lainnya.**

**Fungsi:**

**Sebagai container/tempat meletakkan semua kontrol**

**Mengatur tampilan dan layout aplikasi**

- **TextBox**

**Definisi: Jendela utama aplikasi yang menjadi wadah untuk semua komponen lainnya.**

**Fungsi:**

**Sebagai container/tempat meletakkan semua kontrol**

**Mengatur tampilan dan layout aplikasi**

- **ComboBox**

**Definisi: Dropdown list yang menampilkan daftar pilihan.**

**Fungsi:**

**Menampilkan daftar pilihan dalam bentuk dropdown**

**Membatasi input hanya pada pilihan yang tersedia**

**Menghemat ruang di form**

- **Date Time Picker**

**Definisi: Kontrol untuk memilih tanggal dan waktu dengan kalender.**

**Fungsi:**

**Memudahkan pemilihan tanggal**

**Validasi format tanggal otomatis**

**Menampilkan kalender interaktif**

- **Label**

**Definisi: Teks statis yang tidak dapat diedit oleh user.**

**Fungsi:**

**Menampilkan judul atau keterangan**

**Memberi label/identitas pada kontrol lain**

**Menampilkan informasi yang bersifat tetap**

- **Picture Box**

**Definisi:** Kontrol untuk menampilkan gambar atau image.

**Fungsi:**

**Menampilkan logo, foto, atau gambar produk**

**Menampilkan ikon**

- **Message Box**

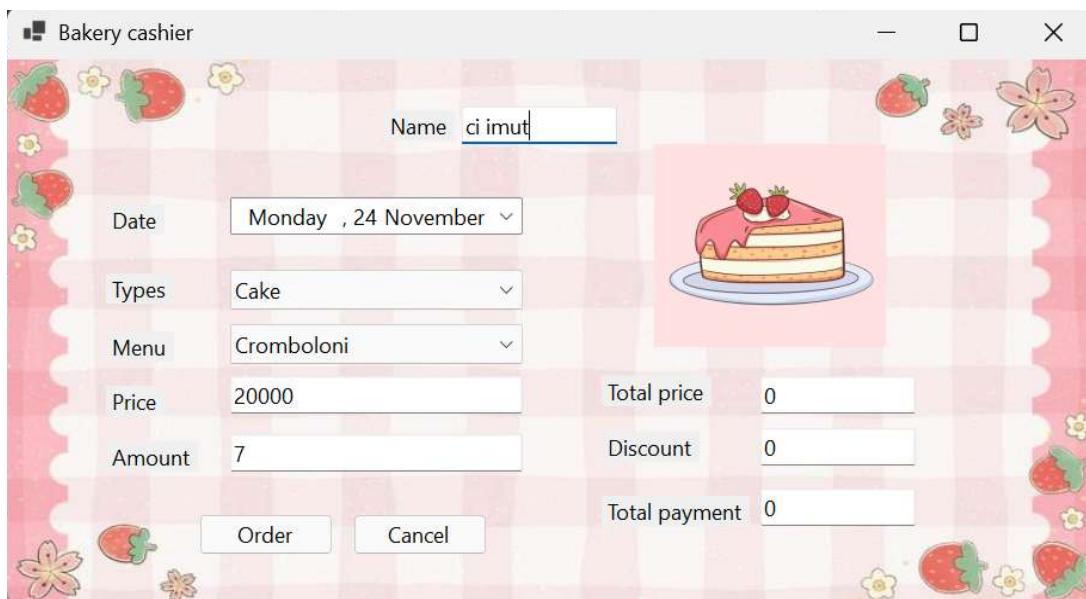
**Definisi:** Dialog box pop-up yang muncul untuk menampilkan pesan.

**Fungsi:**

**Menampilkan informasi kepada user**

**Memberi peringatan atau notifikasi**

### 3. Perancangan Program



### 4. Implementasi Program

namespace Rumah\_Makan

{

    public partial class Form1 : Form

    {

        public Form1()

        {

            InitializeComponent();

        }

```

private void Form1_Load(object sender, EventArgs e)
{
}

// Variabel untuk menyimpan hasil perhitungan
private decimal totalHarga = 0;
private decimal potonganHarga = 0;
private decimal totalBayar = 0;
private decimal grandTotal = 0;

private void Fo_Load(object sender, EventArgs e)
{
    jm.Items.Add("Cake");
    jm.Items.Add("Drink");
    nm.Items.Clear();
}

private void jm_SelectedIndexChanged(object sender, EventArgs e)
{
    if (jm.Text == "Cake")
    {
        nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
        nm.Items.Clear();
        nm.Text = "";
        nm.Items.Add("Brulee Cake");
        nm.Items.Add("Cromboloni");
        nm.Items.Add("Korean Cream Cake");
        nm.Items.Add("Mochi Crunch Cake");
        nm.SelectedIndexChanged += nm_SelectedIndexChanged;
    }
}

else if (jm.Text == "Drink")
{
}

```

```

        nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
        nm.Items.Clear();
        nm.Text = "";
        nm.Items.Add("Dalgona Coffe");
        nm.Items.Add("Squash Yakult");
        nm.Items.Add("Magic Water");
        nm.Items.Add("Matcha Cream Cloud");
        nm.SelectedIndexChanged += nm_SelectedIndexChanged;
    }
}

private void nm_SelectedIndexChanged(object sender, EventArgs e)
{
    if (nm.Text == "Brulee Cake")
        harga.Text = "25000";

    else if (nm.Text == "Cromboloni")
        harga.Text = "20000";

    else if (nm.Text == "Korean Cream Cake")
        harga.Text = "35000";

    else if (nm.Text == "Mochi Crunch Cake")
        harga.Text = "25000";

    else if (nm.Text == "Dalgona Coffe")
        harga.Text = "10000";

    else if (nm.Text == "Squash Yakult")
        harga.Text = "15000";

    else if (nm.Text == "Magic Water")
        harga.Text = "18000";
}

```

```

else if (nm.Text == "Matcha Cream Cloud")
    harga.Text = "14000";

// Hitung total setelah harga berubah
HitungTotal();
}

// Method untuk menghitung total, diskon, dan total bayar
private void HitungTotal()
{
    try
    {
        // Ambil harga
        decimal hargaSatuan = 0;
        decimal.TryParse(harga.Text, out hargaSatuan);

        // Ambil jumlah
        int jml = 0;
        int.TryParse(jumlah.Text, out jml);

        // Hitung total harga
        totalHarga = hargaSatuan * jml;

        // Hitung diskon
        if (totalHarga >= 50000)
            potonganHarga = 10000;
        else
            potonganHarga = 0;

        // Hitung total bayar
        totalBayar = totalHarga - potonganHarga;

        // Tampilkan hasil
        th.Text = totalHarga.ToString("N0");
    }
}

```

```

disk.Text = potonganHarga.ToString("N0");
tb.Text = totalBayar.ToString("N0");
}
catch (Exception ex)
{
    MessageBox.Show("Error: " + ex.Message);
}
}

// Event saat jumlah berubah
private void jumlah_TextChanged(object sender, EventArgs e)
{
    HitungTotal();
}

// Tombol PESAN (button1)
private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(nno.Text))
    {
        MessageBox.Show("Please fill the Name!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jm.Text))
    {
        MessageBox.Show("Please select Types!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(nm.Text))

```

```

    {
        MessageBox.Show("Please select Drink/Cake!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jumlah.Text) || jumlah.Text == "0")
    {
        MessageBox.Show("Please fill Amount!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Hitung total
    HitungTotal();

    // Tampilkan NOTA pesanan
    string pesan = "===== DETAIL ORDER =====\n\n" +
        $"NAME: {nno.Text}\n" +
        $"TYPES: {jm.Text}\n" +
        $"MENU: {nm.Text}\n" +
        $"PRICE: Rp {harga.Text}\n" +
        $"AMOUNT: {jumlah.Text}\n\n" +
        "-----\n" +
        $"TOTAL PRICE: Rp {totalHarga:N0}\n" +
        $"DISCOUNT: Rp {potonganHarga:N0}\n" +
        "-----\n" +
        $"TOTAL PAYMENT: Rp {totalBayar:N0}";

    MessageBox.Show(pesan, "Konfirmasi Pesanan", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
}

// Tombol BERSIH (button2)

```

```

private void
    bton2_Click(object sender, EventArgs e)
{
    nno.Text = "";
    jm.Text = "";
    jm.Items.Clear();
    nm.Items.Clear();
    nm.Text = "";
    harga.Text = "";
    jumlah.Text = "";
    th.Text = "";
    disk.Text = "";
    tb.Text = "";

    totalHarga = 0;
    potonganHarga = 0;
    totalBayar = 0;
}

}

}

```

## 5. Penjelasan Kode Program

### a. Deklarasi Namespace dan Class

namespace Rumah\_Makan

{

public partial class Form1 : Form

*Kode ini mendeklarasikan namespace bernama **Rumah\_Makan** yang berfungsi sebagai wadah untuk mengorganisir kode program. Class **Form1** merupakan turunan dari class **Form** yang merupakan bagian dari Windows Forms Application.*

**b. Constructor Form1**

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

*Constructor adalah method khusus yang dipanggil pertama kali saat form dibuat. Method `InitializeComponent()` berfungsi untuk menginisialisasi semua komponen visual yang ada pada form (seperti textbox, combobox, button, dll).*

**c. Deklarasi Variabel Global**

```
private decimal totalHarga = 0;
private decimal potonganHarga = 0;
private decimal totalBayar = 0;
private decimal grandTotal = 0;
```

*Variabel-variabel ini dideklarasikan dengan tipe data `decimal` untuk menyimpan nilai mata uang dengan presisi tinggi. Variabel `totalHarga` untuk menyimpan total harga sebelum diskon, `potonganHarga` untuk nilai diskon, `totalBayar` untuk total yang harus dibayar setelah diskon, dan `grandTotal` untuk keperluan akumulasi (meskipun belum digunakan dalam kode ini).*

**d. Event Handler Fo\_Load**

```
private void Fo_Load(object sender, EventArgs e)
{
    jm.Items.Add("Cake");
    jm.Items.Add("Drink");
    nm.Items.Clear();
}
```

*Method ini dijalankan saat form pertama kali dimuat. Berfungsi untuk mengisi ComboBox `jm` (jenis menu) dengan dua pilihan: "Cake" dan "Drink". ComboBox `nm` (nama menu) dikosongkan terlebih dahulu menggunakan `Clear()`.*

**e. Event Handler jm\_SelectedIndexChanged**

```
private void jm_SelectedIndexChanged(object sender, EventArgs e)
{
    if (jm.Text == "Cake")
    {
        nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
        nm.Items.Clear();
        nm.Text = "";
        nm.Items.Add("Brulee Cake");
        nm.Items.Add("Cromboloni");
        nm.Items.Add("Korean Cream Cake");
        nm.Items.Add("Mochi Crunch Cake");
        nm.SelectedIndexChanged += nm_SelectedIndexChanged;
    }
}
```

```

else if (jm.Text == "Drink")
{
    nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
    nm.Items.Clear();
    nm.Text = "";
    nm.Items.Add("Dalgona Coffe");
    nm.Items.Add("Squash Yakult");
    nm.Items.Add("Magic Water");
    nm.Items.Add("Matcha Cream Cloud");
    nm.SelectedIndexChanged += nm_SelectedIndexChanged;
}
}

```

*Method ini menggunakan **struktur percabangan IF-ELSE** untuk menangani perubahan pilihan jenis menu. Ketika user memilih "Cake", ComboBox nama menu akan diisi dengan menu cake (Brulee Cake, Cromboloni, Korean Cream Cake, Mochi Crunch Cake). Ketika memilih "Drink", akan diisi dengan menu minuman (Dalgona Coffe, Squash Yakult, Magic Water, Matcha Cream Cloud). Event handler sementara dilepas (-=) dan dipasang kembali (+=) untuk menghindari trigger berulang.*

#### f. Event Handler nm\_SelectedIndexChanged

```

private void nm_SelectedIndexChanged(object sender, EventArgs e)
{
    if (nm.Text == "Brulee Cake")
        harga.Text = "25000";

    else if (nm.Text == "Cromboloni")
        harga.Text = "20000";

    else if (nm.Text == "Korean Cream Cake")
        harga.Text = "35000";

    else if (nm.Text == "Mochi Crunch Cake")
        harga.Text = "25000";

    else if (nm.Text == "Dalgona Coffe")
        harga.Text = "10000";

    else if (nm.Text == "Squash Yakult")
        harga.Text = "15000";

    else if (nm.Text == "Magic Water")
        harga.Text = "18000";
}

```

```
else if (nm.Text == "Matcha Cream Cloud")
    harga.Text = "14000";
```

```
// Hitung total setelah harga berubah
HitungTotal();
```

```
}
```

*Method ini menggunakan struktur percabangan IF-ELSE-IF untuk menentukan harga berdasarkan menu yang dipilih. Setiap menu memiliki harga yang berbeda. Setelah harga ditentukan, method HitungTotal() dipanggil untuk menghitung total pembayaran.*

**g. Method HitungTotal()**

```
try
{
    // Ambil harga
    decimal hargaSatuan = 0;
    decimal.TryParse(harga.Text, out hargaSatuan);

    // Ambil jumlah
    int jml = 0;
    int.TryParse(jumlah.Text, out jml);
```

```
// Hitung total harga
totalHarga = hargaSatuan * jml;
```

*Method ini adalah fungsi buatan sendiri yang berfungsi untuk menghitung total harga, diskon, dan total bayar. Menggunakan TryParse() untuk mengkonversi string menjadi angka dengan aman (menghindari error jika input tidak valid). Method ini menggunakan operator aritmatika (perkalian dan pengurangan) serta operator perbandingan (>=) untuk menghitung diskon.*

**h. Logika Perhitungan Diskon**

```
// Hitung total harga
totalHarga = hargaSatuan * jml;
```

```
// Hitung diskon
if (totalHarga >= 50000)
    potonganHarga = 10000;
else
    potonganHarga = 0;
```

```
// Hitung total bayar
totalBayar = totalHarga - potonganHarga;
```

*Bagian ini menggunakan **operator aritmatika** untuk menghitung total harga (perkalian harga satuan dengan jumlah). Kemudian menggunakan **struktur percabangan IF-ELSE** untuk menentukan diskon: jika total harga  $\geq$  Rp 50.000, maka dapat diskon Rp 10.000, jika tidak maka diskon Rp 0. Total bayar dihitung dengan mengurangi total harga dengan potongan.*

i. **Pembuatan String Nota Pesanan**

```
// Tampilkan NOTA pesanan
string pesan = "===== DETAIL ORDER =====\n\n" +
    $"NAME: {nno.Text}\n" +
    $"TYPES: {jm.Text}\n" +
    $"MENU: {nm.Text}\n" +
    $"PRICE: Rp {harga.Text}\n" +
    $"AMOUNT: {jumlah.Text}\n\n" +
    "-----\n" +
    $"TOTAL PRICE: Rp {totalHarga:N0}\n" +
    $"DISCOUNT: Rp {potonganHarga:N0}\n" +
    "-----\n" +
    $"TOTAL PAYMENT: Rp {totalBayar:N0}";
```

```
MessageBox.Show(pesan, "Konfirmasi Pesanan", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
```

*Kode ini menggunakan string interpolation (tanda \$) untuk membuat format nota pesanan yang rapi. Karakter \n berfungsi sebagai baris baru (newline). Operator + digunakan untuk concatenation (penggabungan string). Format {totalHarga:N0} digunakan untuk menampilkan angka dengan pemisah ribuan tanpa desimal.*

j. **Event Handler button1\_Click (Tombol ORDER)**

```
// Tombol ORDER (button1)
private void button1_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(nno.Text))
    {
        MessageBox.Show("Please fill the Name!", "Warning",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jm.Text))
    {
        MessageBox.Show("Please select Types!", "Warning",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
```

```

    }

    if (string.IsNullOrEmpty(nm.Text))
    {
        MessageBox.Show("Please select Drink/Cake!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    if (string.IsNullOrEmpty(jumlah.Text) || jumlah.Text == "0")
    {
        MessageBox.Show("Please fill Amount!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    // Hitung total
    HitungTotal();

    // Tampilkan NOTA pesanan
    string pesan = "===== DETAIL ORDER =====\n\n" +
    $"NAME: {nno.Text}\n" +
    $"TYPES: {jm.Text}\n" +
    $"MENU: {nm.Text}\n" +
    $"PRICE: Rp {harga.Text}\n" +
    $"AMOUNT: {jumlah.Text}\n\n" +
    "-----\n" +
    $"TOTAL PRICE: Rp {totalHarga:N0}\n" +
    $"DISCOUNT: Rp {potonganHarga:N0}\n" +
    "-----\n" +
    $"TOTAL PAYMENT: Rp {totalBayar:N0}";

    MessageBox.Show(pesan, "Konfirmasi Pesanan",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
}

```

*Method ini menggunakan **struktur percabangan IF** bertingkat untuk validasi input. Setiap field (nama, jenis menu, nama menu, jumlah) divalidasi apakah sudah diisi atau belum. Jika ada yang kosong, akan muncul MessageBox peringatan dan program akan keluar dari method menggunakan **return**. Setelah semua validasi lolos, sistem akan menampilkan nota pesanan dalam bentuk MessageBox yang berisi **output** detail pesanan lengkap dengan format string yang rapi.*

**k. Event Handler bton2\_Click (Tombol CANCEL)**

```
// Tombol CANCEL (button2)
private void
    bton2_Click(object sender, EventArgs e)
{
    nno.Text = "";
    jm.Text = "";
    jm.Items.Clear();
    nm.Items.Clear();
    nm.Text = "";
    harga.Text = "";
    jumlah.Text = "";
    th.Text = "";
    disk.Text = "";
    tb.Text = "";

    totalHarga = 0;
    potonganHarga = 0;
    totalBayar = 0;
}
```

*Method ini berfungsi untuk mereset/membersihkan semua input dan output pada form. Setiap TextBox dan ComboBox dikosongkan dengan memberikan nilai string kosong (""). Method `Items.Clear()` digunakan untuk menghapus semua item dalam ComboBox.*

**l. String.IsNullOrEmpty untuk Validasi Input**

```
if (string.IsNullOrEmpty(nno.Text))
{
    MessageBox.Show("Please fill the Name!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}

if (string.IsNullOrEmpty(jm.Text))
{
    MessageBox.Show("Please select Types!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}

if (string.IsNullOrEmpty(nm.Text))
{
    MessageBox.Show("Please select Drink/Cake!", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Warning);
```

```

        return;
    }

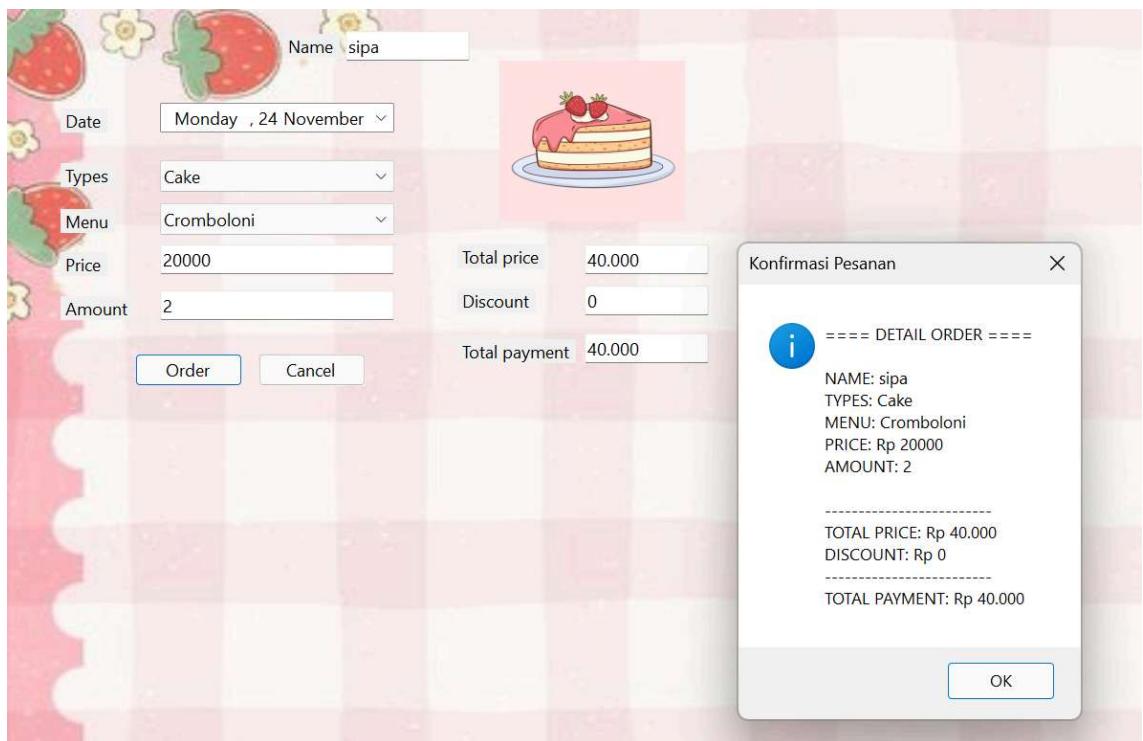
    if (string.IsNullOrEmpty(jumlah.Text) || jumlah.Text == "0")
    {
        MessageBox.Show("Please fill Amount!", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

```

*Method `string.IsNullOrEmpty()` digunakan untuk validasi input, memeriksa apakah TextBox kosong atau berisi null. Ini merupakan cara yang lebih aman daripada hanya memeriksa `== ""` karena juga menangani nilai null. Keyword `return` digunakan untuk menghentikan eksekusi method jika validasi gagal.*

## 6. Tampilan Output Hasil Program

### A. TAMPILAN JIKA SEMUA BAGIAN TER ISI



## B. TAMPILAN JIKA SALAH SATU BAGIAN TIDAK TERISI



### 7. Lampiran

```
1  namespace Rumah_Makan
2  {
3      4 references
4      public partial class Form1 : Form
5      {
6          1 reference
7          public Form1()
8          {
9              InitializeComponent();
10         }
11     }
12     0 references
13     private void Form1_Load(object sender, EventArgs e)
14     {
15         // Variabel untuk menyimpan hasil perhitungan
16         private decimal totalHarga = 0;
17         private decimal potonganHarga = 0;
18         private decimal totalBayar = 0;
19         private decimal grandTotal = 0;
20     }
21     1 reference
22     private void Fo_Load(object sender, EventArgs e)
23     {
24         jm.Items.Add("Cake");
25         jm.Items.Add("Drink");
26         nm.Items.Clear();
27     }
28     1 reference
29     private void jm_SelectedIndexChanged(object sender, EventArgs e)
30     {
31         if (jm.Text == "Cake")
32         {
33             nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
34             nm.Items.Clear();
35             nm.Text = "";
36             nm.Items.Add("Brulee Cake");
37         }
38     }
39 }
```

```

34             nm.Items.Add("Brulee Cake");
35             nm.Items.Add("Cromboloni");
36             nm.Items.Add("Korean Cream Cake");
37             nm.Items.Add("Mochi Crunch Cake");
38             nm.SelectedIndexChanged += nm_SelectedIndexChanged;
39         }
40     }
41     else if (jm.Text == "Drink")
42     {
43         nm.SelectedIndexChanged -= nm_SelectedIndexChanged;
44         nm.Items.Clear();
45         nm.Text = "";
46         nm.Items.Add("Dalgona Coffe");
47         nm.Items.Add("Squash Yakult");
48         nm.Items.Add("Magic Water");
49         nm.Items.Add("Matcha Cream Cloud");
50         nm.SelectedIndexChanged += nm_SelectedIndexChanged;
51     }
52 }
53
5 references
54 private void nm_SelectedIndexChanged(object sender, EventArgs e)
55 {
56     if (nm.Text == "Brulee Cake")
57         harga.Text = "25000";
58
59     else if (nm.Text == "Cromboloni")
60         harga.Text = "20000";
61
62     else if (nm.Text == "Korean Cream Cake")
63         harga.Text = "35000";
64
65     else if (nm.Text == "Mochi Crunch Cake")
66         harga.Text = "25000";
67
68     else if (nm.Text == "Dalgona Coffe")
69         harga.Text = "10000";
70
71     else if (nm.Text == "Squash Yakult")
72         harga.Text = "15000";
73
74     else if (nm.Text == "Magic Water")
75         harga.Text = "18000";
76
77     else if (nm.Text == "Matcha Cream Cloud")
78         harga.Text = "14000";
79
80     // Hitung total setelah harga berubah
81     HitungTotal();
82 }
83
84 // Method untuk menghitung total, diskon, dan total bayar
85 3 references
86 private void HitungTotal()
87 {
88     try
89     {
90         // Ambil harga
91         decimal hargaSatuan = 0;
92         decimal.TryParse(harga.Text, out hargaSatuan);
93
94         // Ambil jumlah
95         int jml = 0;
96         int.TryParse(jumlah.Text, out jml);
97
98         // Hitung total harga
99         totalHarga = hargaSatuan * jml;
100
101         // Hitung diskon
102         if (totalHarga >= 50000)
103             potonganHarga = 10000;
104         else
105             potonganHarga = 0;

```

```

105
106 // Hitung total bayar
107 totalBayar = totalHarga - potonganHarga;
108
109 // Tampilkan hasil
110 th.Text = totalHarga.ToString("N0");
111 disk.Text = potonganHarga.ToString("N0");
112 tb.Text = totalBayar.ToString("N0");
113 }
114 catch (Exception ex)
115 {
116 MessageBox.Show("Error: " + ex.Message);
117 }
118
119
120 // Event saat jumlah berubah
121 private void jumlah_TextChanged(object sender, EventArgs e)
122 {
123     HitungTotal();
124 }
125
126 // Tombol ORDER (button1)
127 private void button1_Click(object sender, EventArgs e)
128 {
129
130     if (string.IsNullOrEmpty(nno.Text))
131     {
132         MessageBox.Show("Please fill the Name!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
133         return;
134     }
135
136     if (string.IsNullOrEmpty(jm.Text))
137     {
138         MessageBox.Show("Please select Types!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
139         return;
140     }
141
142     if (string.IsNullOrEmpty(nm.Text))
143     {
144         MessageBox.Show("Please select Drink/Cake!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
145         return;
146     }
147
148     if (string.IsNullOrEmpty(jumlah.Text) || jumlah.Text == "0")
149     {
150         MessageBox.Show("Please fill Amount!", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
151         return;
152     }
153
154     // Hitung total
155     HitungTotal();
156
157 // Tampilkan NOTA pesanan
158 string pesan = "===== DETAIL ORDER =====\n\n" +
159 $"NAME: {nno.Text}\n" +
160 $"TYPES: {jm.Text}\n" +
161 $"MENU: {nm.Text}\n" +
162 $"PRICE: Rp {harga.Text}\n" +
163 $"AMOUNT: {jumlah.Text}\n\n" +
164 $"-----\n" +
165 $"TOTAL PRICE: Rp {totalHarga:N0}\n" +
166 $"DISCOUNT: Rp {potonganHarga:N0}\n" +
167 $"-----\n" +
168 $"TOTAL PAYMENT: Rp {totalBayar:N0}";
169
170     MessageBox.Show(pesan, "Konfirmasi Pesanan", MessageBoxButtons.OK, MessageBoxIcon.Information);
171 }
172
173 // Tombol CANCEL (button2)
174 private void button2_Click(object sender, EventArgs e)
175 {
176     nno.Text = "";
177     jm.Text = "";

```

```
180     nm.Items.Clear();
181     nm.Text = "";
182     harga.Text = "";
183     jumlah.Text = "";
184     th.Text = "";
185     disk.Text = "";
186     tb.Text = "";
187 
188     totalHarga = 0;
189     potonganHarga = 0;
190     totalBayar = 0;
191 
192 }
193 
194 private void nno_TextChanged(object sender, EventArgs e)
195 {
196 }
197 }
198 }
```

## 8. Screenshot Tampilan Program

