

PROYEK AKHIR SEMESTER GASAL



NAMA : Roichan Aryshaputra

KELAS : X PPLG 1

NIS : 258730

JUDUL PROYEK : Aplikasi Pemesanan McDonalds (Tanpa Transaksi)

PENGEMBANGAN PERANGKAT LUNAK DAN GIM

SMK NEGERI 1 KANDEMAN

TAHUN PELAJARAN 2025/2026

DESKRIPSI PROYEK

Proyek ini bertujuan untuk membuat **aplikasi pemesanan makanan sederhana** yang digunakan pada McDonald's. Aplikasi ini membantu pelanggan untuk mengetahui total harga dari makanan dan minuman yang dia pesan.

Melalui aplikasi ini, pengguna dapat:

1. Memasukkan data pesanan pelanggan.
2. Memilih menu sesuai kategori.
3. Menghitung total harga secara otomatis berdasarkan makanan atau minuman yang dipesan.
4. Menampilkan rincian transaksi berupa makanan dan minuman yang dipesan.
5. Menampilkan riwayat transaksi yang berisi Tanggal dan Jam, Nama Pelanggan, Detail pesanan, Total harga, dan kata-kata Terima kasih, Sampai berjumpa lagi!, I'm lovin' it.

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

1. **Fungsi/Method** untuk memisahkan logika perhitungan, tampilan data, validasi, filter data, dan penyimpanan.
2. **Perulangan(looping)** untuk memfilter menu, menambah jumlah pesanan, menambahkan item ke ComboBox, menulis pesanan ke file(riwayat).
3. **Array atau List** untuk menyimpan id, daftar menu, harga, dan kategori.
4. **Variabel dan Tipe Data** untuk menyimpan informasi menu, harga, total transaksi, nama pelanggan, dan menyimpan pesanan.
5. **Operator dan Percabangan(if, else)** untuk menghitung total harga, memvalidasi input, menentukan alur program, memfilter data, memanipulasi collection.
6. **GUI (Graphical User Interface)** untuk antarmuka aplikasi yang mudah digunakan.

RINGKASAN TEORI C# YANG DIGUNAKAN

1. Fungsi/Method

Fungsi digunakan untuk memisahkan program menjadi beberapa bagian kecil agar lebih terstruktur.

Contoh kode yang digunakan:

```
private void UpdateTotal()
{
    // Menghitung total harga dari semua item pesanan
    decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
    lblTotal.Text = $"Total: Rp {total:N0}";
}
```

2. Perulangan(looping)

Digunakan untuk mengulang proses tertentu, seperti menambahkan item ke ComboBox dan menulis pesanan ke file.

Contoh kode yang digunakan:

```
foreach (var menu in menuItems)
{
    cmbPilihMenu.Items.Add(menu);
}
```

3. Array atau List

Digunakan untuk menyimpan beberapa data sejenis seperti ID, Menu, Kategori, dan Harga.

Contoh kode yang digunakan di program:

```
new Menu { Id = 1, Nama = "Big Mac", Harga = 43500, Kategori = "Burger" },
new Menu { Id = 2, Nama = "Cheeseburger Deluxe", Harga = 34500, Kategori = "Burger" },
new Menu { Id = 3, Nama = "Triple Cheeseburger", Harga = 65500, Kategori = "Burger" },
new Menu { Id = 4, Nama = "Beef Burger Deluxe", Harga = 40000, Kategori = "Burger" },
new Menu { Id = 5, Nama = "Chicken Burger Deluxe", Harga = 30000, Kategori = "Burger" },
new Menu { Id = 6, Nama = "PaMer 7 Ayam McD Gulai Krispy", Harga = 198500, Kategori = "Ayam" },
new Menu { Id = 7, Nama = "2 ALC Ayam Krispy McD Gulai", Harga = 50000, Kategori = "Ayam" },
new Menu { Id = 8, Nama = "PaNas 1 Ayam McD Gulai Krispy", Harga = 43000, Kategori = "Ayam" },
new Menu { Id = 9, Nama = "PaNas 2 Ayam McD Gulai Spicy Large", Harga = 66500, Kategori = "Ayam" },
},
new Menu { Id = 10, Nama = "2pcs Korean Soy Garlic Wings", Harga = 16500, Kategori = "Ayam" },
new Menu { Id = 11, Nama = "French Fries (M)", Harga = 15000, Kategori = "Makanan Pendamping" },
new Menu { Id = 12, Nama = "French Fries(L)", Harga = 20000, Kategori = "Makanan Pendamping" },
new Menu { Id = 13, Nama = "Hash Brown", Harga = 15000, Kategori = "Makanan Pendamping" },
new Menu { Id = 14, Nama = "Apple Pie", Harga = 12000, Kategori = "Makanan Pendamping" },
new Menu { Id = 15, Nama = "Coca-Cola (M)", Harga = 12000, Kategori = "Minuman" },
new Menu { Id = 16, Nama = "Coca-Cola (L)", Harga = 15000, Kategori = "Minuman" },
new Menu { Id = 18, Nama = "Sprite (L)", Harga = 15000, Kategori = "Minuman" },
new Menu { Id = 19, Nama = "Orange juice", Harga = 18000, Kategori = "Minuman" },
```

4. Variabel dan Tipe Data

Digunakan untuk menyimpan data yang akan digunakan didalam program.

Contoh kode yang digunakan:

```
public int Id { get; set; }
public string Nama { get; set; }
public decimal Harga { get; set; }
public string Kategori { get; set; }
```

Penjelasan:

- string → menyimpan teks (nama menu, nama pelanggan, dan nama kategori).
- int → menyimpan bilangan bulat (jumlah pesanan, ID).
- decimal → menyimpan angka dengan tingkat presisi yang tinggi (harga, total).

5. Operator

Operator digunakan untuk melakukan perhitungan atau operasi logika.

Contoh kode yang digunakan:

- `return $"{Menu.Nama} x{Jumlah} - Rp {Menu.Harga * Jumlah:N0}";`
- `if (cmbKategori.SelectedIndex == 0)`
- `if (cmbPilihMenu.SelectedIndex > 0)`
- `if (lstPesanan.SelectedIndex >= 0)`

Jenis operator yang digunakan:

- Aritmatika: *
- Perbandingan: ==, !=, >, >=
- Assignment : = .Add(), .Remove(), .Clear()

6. Percabangan(Kondisi)

Digunakan untuk menentukan alur program berdasarkan kondisi tertentu.

Contoh kode yang digunakan:

```
if (lstPesanan.SelectedIndex >= 0)
{
    ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
    orderItems.Remove(selectedItem);
    lstPesanan.Items.Remove(selectedItem);
    UpdateTotal();
}
// Jika tidak ada item yang dipilih, tampilkan peringatan
else
{
    MessageBox.Show("Silakan pilih item pesanan yang akan dihapus.", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
```

7. Input dan Output

Input digunakan untuk menerima data dari pengguna, sedangkan output menampilkan hasilnya.

Contoh:

TextBox digunakan untuk menerima input dari pengguna sedangkan ListBox digunakan untuk menampilkan hasil dari pengguna.

8. GUI(Graphical User Interface)

Digunakan untuk membuat tampilan aplikasi menjadi lebih interaktif.

Komponen yang digunakan:

Label → menampilkan nama teks

TextBox → untuk memasukkan nama pelanggan

Button → untuk menjalankan perintah simpan, reset, kurangi, tambah, dan lihat riwayat

ListBox → menampilkan data pesanan

ComboBox → menampilkan daftar pilihan

PERANCANGAN PROGRAM

Rancangan Tampilan Menu McDonald's

 Menu Pesanan McDonald's

—

□

×

Selamat Datang di McDonald'

Nama

Kategori

Pilih Menu

Tambah

 Pesanan Kamu

IstPesanan

× Hapus

Reset

Kurangi

TOTAL: Rp 0

 CHECKOUT

 Riwayat

IMPLEMENTASI PROGRAM

Kode program utama

```
namespace projekAkhir_RoichanAryshaputra
{
    public partial class Form1 : Form
    {
        //Membuat Variabel yang berisi data data dari class menu dan ItemPesanan
        private List<Menu> menuItems;
        private List<ItemPesanan> orderItems;
        //Membuat Variabel yang berisi nama file
        private string fileRiwayat = "riwayat_pesanan.txt";
        public Form1()
        {
            //Inisialisasi komponen form
            InitializeComponent();
            //Inisialisasi list orderItems
            orderItems = new List<ItemPesanan>();
            //Memanggil method inisialisasi menuItems
            initializemenuItems();
        }

        //Inisialisasi data menu
        private void initializemenuItems()
        {
            //Menambahkan data data menu ke dalam list menuItems
            menuItems = new List<Menu>
            {
                //kategori burger
                new Menu { Id = 1, Nama = "Big Mac", Harga = 43500, Kategori = "Burger" },
                new Menu { Id = 2, Nama = "Cheeseburger Deluxe", Harga = 34500, Kategori = "Burger" },
                new Menu { Id = 3, Nama = "Triple Cheeseburger", Harga = 65500, Kategori = "Burger" },
                new Menu { Id = 4, Nama = "Beef Burger Deluxe", Harga = 40000, Kategori = "Burger" },
                new Menu { Id = 5, Nama = "Chicken Burger Deluxe", Harga = 30000, Kategori = "Burger" },
                //kategori Ayam
                new Menu { Id = 6, Nama = "PaMer 7 Ayam McD Gulai Krispy", Harga = 198500, Kategori = "Ayam" },
                new Menu { Id = 7, Nama = "2 ALC Ayam Krispy McD Gulai", Harga = 50000, Kategori = "Ayam" },
                new Menu { Id = 8, Nama = "PaNas 1 Ayam McD Gulai Krispy", Harga = 43000, Kategori = "Ayam" },
                new Menu { Id = 9, Nama = "PaNas 2 Ayam McD Gulai Spicy Large", Harga = 66500, Kategori = "Ayam" },

                new Menu { Id = 10, Nama = "2pcs Korean Soy Garlic Wings", Harga = 16500, Kategori = "Ayam" },
                //kategori Makanan Pendamping
                new Menu { Id = 11, Nama = "French Fries (M)", Harga = 15000, Kategori = "Makanan Pendamping" },
                new Menu { Id = 12, Nama = "French Fries(L)", Harga = 20000, Kategori = "Makanan Pendamping" },
                new Menu { Id = 13, Nama = "Hash Brown", Harga = 15000, Kategori = "Makanan Pendamping" },
                new Menu { Id = 14, Nama = "Apple Pie", Harga = 12000, Kategori = "Makanan Pendamping" },
                // kategori Minuman
                new Menu { Id = 15, Nama = "Coca-Cola (M)", Harga = 12000, Kategori = "Minuman" },
                new Menu { Id = 16, Nama = "Coca-Cola (L)", Harga = 15000, Kategori = "Minuman" },
                new Menu { Id = 17, Nama = "Sprite (M)", Harga = 12000, Kategori = "Minuman" },
                new Menu { Id = 18, Nama = "Sprite (L)", Harga = 15000, Kategori = "Minuman" },
                new Menu { Id = 19, Nama = "Orange juice", Harga = 18000, Kategori = "Minuman" },
            },
        }
    }
}
```

```

    new Menu { Id = 20, Nama = "McCafe latte", Harga = 25000, Kategori = "Minuman" },
    // kategori Makanan Penutup
    new Menu { Id = 21, Nama = "McFlurry Oreo", Harga = 25000, Kategori = "Makanan Penutup" },
    new Menu { Id = 22, Nama = "Sundae Chocolate", Harga = 15000, Kategori = "Makanan Penutup"
},

    new Menu { Id = 23, Nama = "Sunday Strawberry", Harga = 25000, Kategori = "Makanan Penutup" }
};
//Mengisi ComboBox Kategori dengan kategori unik dari menuItems
var kategori = menuItems.Select(m => m.Kategori).Distinct().OrderBy(k => k).ToList();
cmbKategori.Items.Add("=== Semua Kategori ===");
//Menambahkan setiap kategori ke dalam ComboBox
foreach (var Kategori in kategori)
{
    cmbKategori.Items.Add(Kategori);
}
cmbKategori.SelectedIndex = 0;

}

public class Menu
{
    //Properti dari class menu
    public int Id { get; set; }
    public string Nama { get; set; }
    public decimal Harga { get; set; }
    public string Kategori { get; set; }
    //Override method ToString untuk menampilkan nama dan harga menu
    public override string ToString()
    {
        return $"{Nama} - Rp {Harga:N0}";
    }
}

//Membuat class ItemPesanan
public class ItemPesanan
{
    //Properti dari class ItemPesanan
    public Menu Menu { get; set; }
    public int Jumlah { get; set; } = 1;
    //Constructor dari class ItemPesanan
    public ItemPesanan(Menu item)
    {
        Menu = item;
    }
    //Override method ToString untuk menampilkan nama dan harga menu
    public override string ToString()
    {
        return $"{Menu.Nama} x{Jumlah} - Rp {Menu.Harga * Jumlah:N0}";
    }
}

private void btnSimpan_Click(object sender, EventArgs e)
{

```

```

// Cek apakah ada pesanan yang disimpan
if (orderItems.Count == 0)
{
    MessageBox.Show("Tidak ada pesanan yang disimpan saat ini.", "Warning",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    return;
}
// Mendapatkan nama pelanggan, jika kosong gunakan "Pelanggan"
string namapelanggan = string.IsNullOrEmpty(txtNama.Text)
    ? "Pelanggan" : txtNama.Text.Trim();
// Menyimpan riwayat pesanan ke file teks
try
{
    // Membuka file riwayat dalam mode append
    using (StreamWriter writer = new StreamWriter(fileRiwayat, true))
    {
        // Menulis header riwayat pesanan
        writer.WriteLine("=".PadRight(70, '='));
        writer.WriteLine("                Riwayat Pesanan McDonald's");
        writer.WriteLine("=".PadRight(70, '='));
        writer.WriteLine($"Tanggal dan Jam: {DateTime.Now:dd/MM/yyyy HH:mm:ss}");
        writer.WriteLine($"Pelanggan: {namapelanggan}");
        writer.WriteLine("-".PadRight(70, '-'));
        writer.WriteLine("Detail Pesanan:");
        writer.WriteLine();

        int no = 1;

        // Menulis setiap item pesanan ke file
        foreach (var item in orderItems)
        {
            writer.WriteLine($"{no}. {item.Menu.Nama} x{item.Jumlah} - Rp {item.Menu.Harga:N0}");
            no++;
        }

        decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
        writer.WriteLine();
        writer.WriteLine("-".PadRight(70, '-'));
        writer.WriteLine($"    {"TOTAL".PadRight(40)} Rp {total:N0}");
        writer.WriteLine("=".PadRight(70, '='));
        writer.WriteLine("                Terimakasih, Sampai berjumpa lagi!");
        writer.WriteLine("                I'm lovin' it");
        writer.WriteLine("=".PadRight(70, '='));
        writer.WriteLine();
        writer.WriteLine();
    }
    MessageBox.Show($"pesanan berhasil di simpan !\n\nNama Pelanggan: {namapelanggan}\n" +
        $"Total Pembayaran: Rp {orderItems.Sum(o => o.Menu.Harga * o.Jumlah):N0}",
        "Sukses", MessageBoxButtons.OK, MessageBoxIcon.Information);

    // Reset pesanan setelah disimpan
    orderItems.Clear();
    lstPesanan.Items.Clear();
}

```



```

        txtNama.Clear();
        cmbKategori.SelectedIndex = 0;
        UpdateTotal();
    }
    // Menangani kesalahan saat menyimpan pesanan
    catch (Exception ex)
    {
        MessageBox.Show($"Gagal menyimpan pesanan. Error: {ex.Message}", "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

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

private void cmbKategori_SelectedIndexChanged(object sender, EventArgs e)
{
    // Memperbarui daftar menu berdasarkan kategori yang dipilih
    cmbPilihMenu.Items.Clear();
    cmbPilihMenu.Items.Add("=== Pilih Menu ===");

    // Menampilkan semua menu jika kategori "Semua Kategori" dipilih
    if (cmbKategori.SelectedIndex == 0)
    {
        // Tampilkan semua menu
        foreach (var menu in menuItems)
        {
            cmbPilihMenu.Items.Add(menu);
        }
    }

    // Menampilkan menu berdasarkan kategori yang dipilih
    else
    {
        string selectedCategory = cmbKategori.SelectedItem.ToString();
        var filteredMenu = menuItems.Where(m => m.Kategori == selectedCategory).ToList();
        foreach (var menu in filteredMenu)
        {
            cmbPilihMenu.Items.Add(menu);
        }
    }
    cmbPilihMenu.SelectedIndex = 0;
}

private void btnTambah_Click(object sender, EventArgs e)
{
    // Menambahkan item pesanan ke daftar pesanan
    if (cmbPilihMenu.SelectedIndex > 0)
    {
        Menu selectedMenu = (Menu)cmbPilihMenu.SelectedItem;
        // Cek apakah item pesanan sudah ada dalam daftar
    }
}

```

```

var existingItem = orderItems.FirstOrDefault(o => o.Menu.Id == selectedMenu.Id);
// Jika sudah ada, tambahkan jumlahnya
if (existingItem != null)
{
    // Menambahkan jumlah item pesanan
    existingItem.Jumlah++;
    // Memperbarui tampilan item pesanan di ListBox
    int index = lstPesanan.Items.IndexOf(existingItem);
    lstPesanan.Items[index] = existingItem;
}
else
{
    ItemPesanan itemPesanan = new ItemPesanan(selectedMenu);
    orderItems.Add(itemPesanan);
    lstPesanan.Items.Add(itemPesanan);
}

UpdateTotal();
}
// Jika tidak ada menu yang dipilih, tampilkan peringatan
else
{
    MessageBox.Show("Silakan pilih menu terlebih dahulu.", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

private void btnHapus_Click(object sender, EventArgs e)
{
    // Menghapus item pesanan yang dipilih dari daftar pesanan
    if (lstPesanan.SelectedIndex >= 0)
    {
        ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
        orderItems.Remove(selectedItem);
        lstPesanan.Items.Remove(selectedItem);
        UpdateTotal();
    }
    // Jika tidak ada item yang dipilih, tampilkan peringatan
    else
    {
        MessageBox.Show("Silakan pilih item pesanan yang akan dihapus.", "Peringatan",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

// Memperbarui total harga pesanan
private void UpdateTotal()
{
    // Menghitung total harga dari semua item pesanan
    decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
    lblTotal.Text = $"Total: Rp {total:N0}";
}

```

```

private void btnReset_Click(object sender, EventArgs e)
{
    // Mereset daftar pesanan setelah konfirmasi dari pengguna
    if (orderItems.Count > 0)
    {
        // Menampilkan dialog konfirmasi
        DialogResult result = MessageBox.Show("Apakah Anda yakin ingin mereset pesanan?",
            "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        // Jika pengguna memilih "Yes", reset daftar pesanan
        if (result == DialogResult.Yes)
        {
            orderItems.Clear();
            lstPesanan.Items.Clear();
            UpdateTotal();
        }
    }
}

// Membuka file riwayat pesanan dengan Notepad
private void btnRiwayat_Click(object sender, EventArgs e)
{
    // Memanggil method untuk membuka riwayat dengan Notepad
    BukaRiwayat();
}

// Method untuk membuka file riwayat pesanan dengan Notepad
private void BukaRiwayat()
{
    // Cek apakah file riwayat ada
    try
    {
        // Jika file tidak ada, tampilkan pesan informasi
        if (!File.Exists(fileRiwayat))
        {
            MessageBox.Show("Belum ada riwayat pesanan.\n\nSilakan lakukan checkout terlebih
dahulu.",
                "File Tidak Ditemukan",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            return;
        }
        // Membuka file riwayat dengan Notepad
        System.Diagnostics.Process.Start("notepad.exe", fileRiwayat);
    }
    // Menangani kesalahan saat membuka file riwayat
    catch (Exception ex)
    {
        // Menampilkan pesan error jika gagal membuka file
        MessageBox.Show($"Gagal membuka file riwayat!\n\nError: {ex.Message}",
            "Error",
            MessageBoxButtons.OK,
            MessageBoxIcon.Error);
    }
}

```

```

    }

    private void Form1_Load_1(object sender, EventArgs e)
    {

    }

    private void btnKurangi_Click(object sender, EventArgs e)
    {
        // Mengurangi jumlah item pesanan yang dipilih
        if (lstPesanan.SelectedIndex >= 0)
        {
            int selectedIndex = lstPesanan.SelectedIndex;
            ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
            // Cek apakah jumlah item lebih dari 1
            if (selectedItem.Jumlah > 1)
            {
                // Kurangi jumlah item pesanan
                selectedItem.Jumlah--;

                // Update tampilan ListBox
                lstPesanan.Items.Clear();
                // tambah semua item pesanan kembali ke ListBox
                foreach (var item in orderItems)
                {
                    lstPesanan.Items.Add(item);
                }

                lstPesanan.SelectedIndex = selectedIndex;
            }
            // Jika jumlah item adalah 1, hapus item dari daftar
            else
            {
                orderItems.Remove(selectedItem);
                lstPesanan.Items.Remove(selectedItem);
            }

            UpdateTotal();
        }
        else
        {
            MessageBox.Show("Silakan pilih item pesanan yang akan dikurangi.", "Peringatan",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}

```

Penjelasan Kode Program

1. Deklarasi Namespace dan Class

Namespace projekAkhir_RoichanAryshaputra

```
{  
    public partial class Form1 : Form  
    {  
    }  
}
```

- Namespace projekAkhir_RoichanAryshaputra → ruang lingkup program agar tidak bentrok dengan program lain.
- Class Form1 : Form → kelas utama tempat kode dijalankan.

2. Fungsi initializemenuItems()

```
private void initializemenuItems()  
{  
    //Menambahkan data data menu ke dalam list menuItems  
    menuItems = new List<Menu>  
    {  
        //kategori burger  
        new Menu { Id = 1, Nama = "Big Mac", Harga = 43500, Kategori = "Burger" },  
        new Menu { Id = 2, Nama = "Cheeseburger Deluxe", Harga = 34500, Kategori = "Burger" },  
        new Menu { Id = 3, Nama = "Triple Cheeseburger", Harga = 65500, Kategori = "Burger" },  
        new Menu { Id = 4, Nama = "Beef Burger Deluxe", Harga = 40000, Kategori = "Burger" },  
        new Menu { Id = 5, Nama = "Chicken Burger Deluxe", Harga = 30000, Kategori = "Burger" },  
        //kategori Ayam  
        new Menu { Id = 6, Nama = "PaMer 7 Ayam McD Gulai Krispy", Harga = 198500, Kategori = "Ayam" },  
        new Menu { Id = 7, Nama = "2 ALC Ayam Krispy McD Gulai", Harga = 50000, Kategori = "Ayam" },  
        new Menu { Id = 8, Nama = "PaNas 1 Ayam McD Gulai Krispy", Harga = 43000, Kategori = "Ayam" },  
        new Menu { Id = 9, Nama = "PaNas 2 Ayam McD Gulai Spicy Large", Harga = 66500, Kategori = "Ayam" },  
    },  
    new Menu { Id = 10, Nama = "2pcs Korean Soy Garlic Wings", Harga = 16500, Kategori = "Ayam" },  
    //kategori Makanan Pendamping  
    new Menu { Id = 11, Nama = "French Fries (M)", Harga = 15000, Kategori = "Makanan Pendamping" },  
    new Menu { Id = 12, Nama = "French Fries (L)", Harga = 20000, Kategori = "Makanan Pendamping" },  
    new Menu { Id = 13, Nama = "Hash Brown", Harga = 15000, Kategori = "Makanan Pendamping" },  
    new Menu { Id = 14, Nama = "Apple Pie", Harga = 12000, Kategori = "Makanan Pendamping" },  
    // kategori Minuman  
    new Menu { Id = 15, Nama = "Coca-Cola (M)", Harga = 12000, Kategori = "Minuman" },  
    new Menu { Id = 16, Nama = "Coca-Cola (L)", Harga = 15000, Kategori = "Minuman" },  
    new Menu { Id = 17, Nama = "Sprite (M)", Harga = 12000, Kategori = "Minuman" },  
    new Menu { Id = 18, Nama = "Sprite (L)", Harga = 15000, Kategori = "Minuman" },  
    new Menu { Id = 19, Nama = "Orange juice", Harga = 18000, Kategori = "Minuman" },  
    new Menu { Id = 20, Nama = "McCafe latte", Harga = 25000, Kategori = "Minuman" },  
    // kategori Makanan Penutup  
    new Menu { Id = 21, Nama = "McFlurry Oreo", Harga = 25000, Kategori = "Makanan Penutup" },  
    new Menu { Id = 22, Nama = "Sundae Chocolate", Harga = 15000, Kategori = "Makanan Penutup" },  
    new Menu { Id = 23, Nama = "Sunday Strawberry", Harga = 25000, Kategori = "Makanan Penutup" }  
};  
//Mengisi ComboBox Kategori dengan kategori unik dari menuItems  
var kategori = menuItems.Select(m => m.Kategori).Distinct().OrderBy(k => k).ToList();
```

```

cmbKategori.Items.Add("=== Semua Kategori ===");
//Menambahkan setiap kategori ke dalam ComboBox
foreach (var Kategori in kategori)
{
    cmbKategori.Items.Add(Kategori);
}
cmbKategori.SelectedIndex = 0;
}

```

- Fungsi ini bertujuan untuk membuat dan mengisi daftar menu kemudian mengisi ComboBox kategori dengan kategori-kategori yang ada.

3. Fungsi UpdateTotal()

```

private void UpdateTotal()
{
    // Menghitung total harga dari semua item pesanan
    decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
    lblTotal.Text = $"Total: Rp {total:N0}";
}

```

- Menghitung total dari semua item pesanan menggunakan LINQ (Language Integrated Query) dan di hitung dengan cara = Harga * Jumlah kemudia di tampilkan hasil nya di tabel dengan format Rupiah (:N0 untuk pemisah ribuan).

4. Fungsi BukaRiwayat

```

private void BukaRiwayat()
{
    // Cek apakah file riwayat ada
    try
    {
        // Jika file tidak ada, tampilkan pesan informasi
        if (!File.Exists(fileRiwayat))
        {
            MessageBox.Show("Belum ada riwayat pesanan.\n\nSilakan lakukan checkout terlebih dahulu.",
                "File Tidak Ditemukan",
                MessageBoxButtons.OK,
                MessageBoxIcon.Information);
            return;
        }
        // Membuka file riwayat dengan Notepad
        System.Diagnostics.Process.Start("notepad.exe", fileRiwayat);
    }
    // Menangani kesalahan saat membuka file riwayat
    catch (Exception ex)
    {
        // Menampilkan pesan error jika gagal membuka file
        MessageBox.Show($"Gagal membuka file riwayat!\n\nError: {ex.Message}",
            "Error",
            MessageBoxButtons.OK,

```

```

        MessageBoxIcon.Error);
    }
}

```

- Kode ini berfungsi untuk membuka file “riwayat_pesanan.txt” menggunakan notepad.

5. Class Menu

```

public class Menu
{
    //Properti dari class menu
    public int Id { get; set; }
    public string Nama { get; set; }
    public decimal Harga { get; set; }
    public string Kategori { get; set; }
    //Override method ToString untuk menampilkan nama dan harga menu
    public override string ToString()
    {
        return $"{Nama} - Rp {Harga:N0}";
    }
}

```

- Class untuk merepresentasikan data menu makanan.
- Method Override digunakan saat menu ditampilkan di ComboBox.

6. Class ItemPesanan

```

public class ItemPesanan
{
    //Properti dari class ItemPesanan
    public Menu Menu { get; set; }
    public int Jumlah { get; set; } = 1;
    //Constructor dari class ItemPesanan
    public ItemPesanan(Menu item)
    {
        Menu = item;
    }
}

```

- Class yang merepresentasikan item ke dalam ListBox.
- Cara kerjanya adalah menerima parameter menu saat objek dibuat, kemudian jumlah otomatis di-set ke 1.
- Fungsi dari class ini adalah menyimpan informasi pesanan dengan jumlahnya.

7. Kode btnSimpan

```

private void btnSimpan_Click(object sender, EventArgs e)
{
    // Cek apakah ada pesanan yang disimpan
    if (orderItems.Count == 0)
    {
        MessageBox.Show("Tidak ada pesanan yang disimpan saat ini.", "Warning",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

```

        return;
    }
    // Mendapatkan nama pelanggan, jika kosong gunakan "Pelanggan"
    string namapelanggan = string.IsNullOrEmpty(txtNama.Text)
        ? "Pelanggan" : txtNama.Text.Trim();
    // Menyimpan riwayat pesanan ke file teks
    try
    {
        // Membuka file riwayat dalam mode append
        using (StreamWriter writer = new StreamWriter(fileRiwayat, true))
        {
            // Menulis header riwayat pesanan
            writer.WriteLine("=".PadRight(70, '='));
            writer.WriteLine("                Riwayat Pesanan McDonald's");
            writer.WriteLine("=".PadRight(70, '='));
            writer.WriteLine($"Tanggal dan Jam: {DateTime.Now:dd/MM/yyyy HH:mm:ss}");
            writer.WriteLine($"Pelanggan: {namapelanggan}");
            writer.WriteLine("-".PadRight(70, '-'));
            writer.WriteLine("Detail Pesanan:");
            writer.WriteLine();

            int no = 1;

            // Menulis setiap item pesanan ke file
            foreach (var item in orderItems)
            {
                writer.WriteLine($"{no}. {item.Menu.Nama} x{item.Jumlah} - Rp
{item.Menu.Harga:N0}");
                no++;
            }
            decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
            writer.WriteLine();
            writer.WriteLine("-".PadRight(70, '-'));
            writer.WriteLine($"    {"TOTAL".PadRight(40)} Rp {total:N0}");
            writer.WriteLine("=".PadRight(70, '='));
            writer.WriteLine("                Terimakasih, Sampai berjumpa lagi!");
            writer.WriteLine("                                I'm lovin' it");
            writer.WriteLine("=".PadRight(70, '='));
            writer.WriteLine();
            writer.WriteLine();
        }
        MessageBox.Show($"pesanan berhasil di simpan !\n\nNama Pelanggan: {namapelanggan}\n" +
            $"Total Pembayaran: Rp {orderItems.Sum(o => o.Menu.Harga * o.Jumlah):N0}",
            "Sukses", MessageBoxButtons.OK, MessageBoxIcon.Information);

        // Reset pesanan setelah disimpan
        orderItems.Clear();
        lstPesanan.Items.Clear();
        txtNama.Clear();
        cmbKategori.SelectedIndex = 0;
        UpdateTotal();
    }
    // Menangani kesalahan saat menyimpan pesanan

```



```

        catch (Exception ex)
        {
            MessageBox.Show($"Gagal menyimpan pesanan. Error: {ex.Message}", "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

- Event handler untuk menyimpan pesanan ke file dan menampilkan konfirmasi.
- Proses nya adalah memvalidasi pesanan → ambil nama pelanggan → menyimpan ke file → mereset form

8. Kode cmbKategori

```

private void cmbKategori_SelectedIndexChanged(object sender, EventArgs e)
{
    // Memperbarui daftar menu berdasarkan kategori yang dipilih
    cmbPilihMenu.Items.Clear();
    cmbPilihMenu.Items.Add("=== Pilih Menu ===");

    // Menampilkan semua menu jika kategori "Semua Kategori" dipilih
    if (cmbKategori.SelectedIndex == 0)
    {
        // Tampilkan semua menu
        foreach (var menu in menuItems)
        {
            cmbPilihMenu.Items.Add(menu);
        }
    }

    // Menampilkan menu berdasarkan kategori yang dipilih
    else
    {
        string selectedCategory = cmbKategori.SelectedItem.ToString();
        var filteredMenu = menuItems.Where(m => m.Kategori == selectedCategory).ToList();
        foreach (var menu in filteredMenu)
        {
            cmbPilihMenu.Items.Add(menu);
        }
    }
    cmbPilihMenu.SelectedIndex = 0;
}

```

- Event handler untuk memfilter menu berdasarkan kategori yang dipilih.
- Cara kerjanya Adalah saat aplikasi dijalankan langsung menampilkan ("=== Pilih Menu ==="), jika pengguna tetap memilih ("=== Pilih Menu ===") semua menu akan ditampilkan di cmbMenu. Jika pengguna memilih kategori lain seperti burger, minuman, atau yang lainnya akan di tampilkan pilihan menu yang sesuai dengan kategori yang dipilih.

9. Kode btnTambah

```

private void btnTambah_Click(object sender, EventArgs e)
{

```

```

// Menambahkan item pesanan ke daftar pesanan
if (cmbPilihMenu.SelectedIndex > 0)
{
    Menu selectedMenu = (Menu)cmbPilihMenu.SelectedItem;
    // Cek apakah item pesanan sudah ada dalam daftar
    var existingItem = orderItems.FirstOrDefault(o => o.Menu.Id == selectedMenu.Id);
    // Jika sudah ada, tambahkan jumlahnya
    if (existingItem != null)
    {
        // Menambahkan jumlah item pesanan
        existingItem.Jumlah++;
        // Memperbarui tampilan item pesanan di ListBox
        int index = lstPesanan.Items.IndexOf(existingItem);
        lstPesanan.Items[index] = existingItem;
    }
    else
    {
        ItemPesanan itemPesanan = new ItemPesanan(selectedMenu);
        orderItems.Add(itemPesanan);
        lstPesanan.Items.Add(itemPesanan);
    }

    UpdateTotal();
}
// Jika tidak ada menu yang dipilih, tampilkan peringatan
else
{
    MessageBox.Show("Silakan pilih menu terlebih dahulu.", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

```

- Event untuk menambah menu ke keranjang pesanan(ListBox)
- Fungsi kode ini adalah menambah menu ke Listbox dan menghitung jumlah harga yang akan di tampilkan di label. Kode ini juga akan menambah jumlah pesanan sesuai dengan tombol ini di tekan.

10. Kode btnHapus

```

private void btnHapus_Click(object sender, EventArgs e)
{
    // Menghapus item pesanan yang dipilih dari daftar pesanan
    if (lstPesanan.SelectedIndex >= 0)
    {
        ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
        orderItems.Remove(selectedItem);
        lstPesanan.Items.Remove(selectedItem);
        UpdateTotal();
    }
    // Jika tidak ada item yang dipilih, tampilkan peringatan
    else
    {
        MessageBox.Show("Silakan pilih item pesanan yang akan dihapus.", "Peringatan",

```

```

        MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```

- Event handler untuk menghapus item pesanan yang dipilih dari keranjang tidak peduli berapa pun jumlahnya.

11. Kode btnReset

```

private void btnReset_Click(object sender, EventArgs e)
{
    // Mereset daftar pesanan setelah konfirmasi dari pengguna
    if (orderItems.Count > 0)
    {
        // Menampilkan dialog konfirmasi
        DialogResult result = MessageBox.Show("Apakah Anda yakin ingin mereset pesanan?",
            "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
        // Jika pengguna memilih "Yes", reset daftar pesanan
        if (result == DialogResult.Yes)
        {
            orderItems.Clear();
            lstPesanan.Items.Clear();
            UpdateTotal();
        }
    }
}

```

- Event handler untuk mereset seluruh pesanan dengan konfirmasi.
- Fitur kode ini adalah memunculkan konfirmasi untuk mencegah reset karena tidak sengaja, membersihkan semua item di ListBox, dan mengubah label informasi harga menjadi Rp 0.

12. Kode btnRiwayat

```

private void btnRiwayat_Click(object sender, EventArgs e)
{
    // Memanggil method untuk membuka riwayat dengan Notepad
    BukaRiwayat();
}

```

- Event handler untuk membuka riwayat pesanan.
- Fungsi kode nya adalah memanggil method BukaRiwayat().

13. Kode btnKurangi

```

private void btnKurangi_Click(object sender, EventArgs e)
{
    // Mengurangi jumlah item pesanan yang dipilih
    if (lstPesanan.SelectedIndex >= 0)
    {
        int selectedIndex = lstPesanan.SelectedIndex;
        ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
        // Cek apakah jumlah item lebih dari 1
        if (selectedItem.Jumlah > 1)
        {

```

```

        // Kurangi jumlah item pesanan
        selectedItem.Jumlah--;

        // Update tampilan ListBox
        lstPesanan.Items.Clear();
        // tambah semua item pesanan kembali ke ListBox
        foreach (var item in orderItems)
        {
            lstPesanan.Items.Add(item);
        }

        lstPesanan.SelectedIndex = selectedIndex;
    }
    // Jika jumlah item adalah 1, hapus item dari daftar
    else
    {
        orderItems.Remove(selectedItem);
        lstPesanan.Items.Remove(selectedItem);
    }

    UpdateTotal();
}
else
{
    MessageBox.Show("Silakan pilih item pesanan yang akan dikurangi.", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
}
}

```

- Event handler untuk mengurangi jumlah item pesanan.
- Kode tersebut akan mengurangi jumlah pesanan yang dipilih oleh pengguna dari ListBox, jika jumlah pesanan tersisa 1 dan pengguna tetap menekan Button Kurangi kode akan menghapus item tersebut.

Tampilan Output hasil Program

 Menu Pesanan McDonald's

Selamat Datang di McDonald'

Nama

Roichan

Kategori

Minuman

Pilih Menu

Coca-Cola (M) - Rp 12,000

Tambah

 Pesanan Kamu

PaMer 7 Ayam McD Gulai Krispy x2 - Rp 397,000
Coca-Cola (M) x2 - Rp 24,000

× Hapus

Reset

Kurangi

Total: Rp 421,000

 CHECKOUT

 Riwayat

LAMPIRAN

Kode Program Lengkap form1

```
1 namespace proyekAkhir_RoichanAryshaputra
2 {
3     4 references
4     public partial class Form1 : Form
5     {
6         //Membuat Variabel yang berisi data data dari class menu dan ItemPesanan
7         private List<Menu> menuItems;
8         private List<ItemPesanan> orderItems;
9         //Membuat Variabel yang berisi nama file
10        private string fileRiwayat = "riwayat_pesanan.txt";
11        1 reference
12        public Form1()
13        {
14            //Inisialisasi komponen form
15            InitializeComponent();
16            //Inisialisasi list orderItems
17            orderItems = new List<ItemPesanan>();
18            //Memanggil method inisialisasi menuItems
19            initializeMenuItems();
20        }
21
22        //Inisialisasi data menu
23        1 reference
24        private void initializeMenuItems()
25        {
26            //Menambahkan data data menu ke dalam list menuItems
27            menuItems = new List<Menu>
28            {
29                //kategori burger
30                new Menu { Id = 1, Nama = "Big Mac", Harga = 43500, Kategori = "Burger" },
31                new Menu { Id = 2, Nama = "Cheeseburger Deluxe", Harga = 34500, Kategori = "Burger" },
32                new Menu { Id = 3, Nama = "Triple Cheeseburger", Harga = 65500, Kategori = "Burger" },
33                new Menu { Id = 4, Nama = "Beef Burger Deluxe", Harga = 40000, Kategori = "Burger" },
34                new Menu { Id = 5, Nama = "Chicken Burger Deluxe", Harga = 30000, Kategori = "Burger" },
35                //kategori Ayam
36                new Menu { Id = 6, Nama = "PaMer 7 Ayam McD Gulai Krispy", Harga = 198500, Kategori = "Ayam" },
37                new Menu { Id = 7, Nama = "2 ALC Ayam Krispy McD Gulai", Harga = 50000, Kategori = "Ayam" },
38                new Menu { Id = 8, Nama = "PaNas 1 Ayam McD Gulai Krispy", Harga = 43000, Kategori = "Ayam" },
39                new Menu { Id = 9, Nama = "PaNas 2 Ayam McD Gulai Spicy Large", Harga = 66500, Kategori = "Ayam" },
40                new Menu { Id = 10, Nama = "2pcs Korean Soy Garlic Wings", Harga = 16500, Kategori = "Ayam" },
41                //kategori Makanan Pendamping
42                new Menu { Id = 11, Nama = "French Fries (M)", Harga = 15000, Kategori = "Makanan Pendamping" },
43                new Menu { Id = 12, Nama = "French Fries(L)", Harga = 20000, Kategori = "Makanan Pendamping" },
44                new Menu { Id = 13, Nama = "Hash Brown", Harga = 15000, Kategori = "Makanan Pendamping" },
45                new Menu { Id = 14, Nama = "Apple Pie", Harga = 12000, Kategori = "Makanan Pendamping" },
46                // kategori Minuman
47                new Menu { Id = 15, Nama = "Coca-Cola (M)", Harga = 12000, Kategori = "Minuman" },
48                new Menu { Id = 16, Nama = "Coca-Cola (L)", Harga = 15000, Kategori = "Minuman" },
49                new Menu { Id = 17, Nama = "Sprite (M)", Harga = 12000, Kategori = "Minuman" },
50                new Menu { Id = 18, Nama = "Sprite (L)", Harga = 15000, Kategori = "Minuman" },
51                new Menu { Id = 19, Nama = "Orange juice", Harga = 18000, Kategori = "Minuman" },
52                new Menu { Id = 20, Nama = "McCafe latte", Harga = 25000, Kategori = "Minuman" },
53                // kategori Makanan Penutup
54                new Menu { Id = 21, Nama = "McFlurry Oreo", Harga = 25000, Kategori = "Makanan Penutup" },
55                new Menu { Id = 22, Nama = "Sundae Chocolate", Harga = 15000, Kategori = "Makanan Penutup" },
56                new Menu { Id = 23, Nama = "Sunday Strawberry", Harga = 25000, Kategori = "Makanan Penutup" }
57            };
58            //Mengisi ComboBox Kategori dengan kategori unik dari menuItems
59            var kategori = menuItems.Select(m => m.Kategori).Distinct().OrderBy(k => k).ToList();
60            cmbKategori.Items.Add("=== Semua Kategori ===");
61            //Menambahkan setiap kategori ke dalam ComboBox
62            foreach (var Kategori in kategori)
63            {
64                cmbKategori.Items.Add(Kategori);
65            }
66            cmbKategori.SelectedIndex = 0;
67        }
68    }
69 }
```

```

68 //Membuat class menu
69 29 references
70 public class Menu
71 {
72     //Properti dari class menu
73     25 references
74     public int Id { get; set; }
75     26 references
76     public string Nama { get; set; }
77     29 references
78     public decimal Harga { get; set; }
79     25 references
80     public string Kategori { get; set; }
81     //Override method ToString untuk menampilkan nama dan harga menu
82     0 references
83     public override string ToString()
84     {
85         return $"{Nama} - Rp {Harga:N0}";
86     }
87 }
88 //Membuat class ItemPesanan
89 9 references
90 public class ItemPesanan
91 {
92     //Properti dari class ItemPesanan
93     9 references
94     public Menu Menu { get; set; }
95     9 references
96     public int Jumlah { get; set; } = 1;
97     //Constructor dari class ItemPesanan
98     1 reference
99     public ItemPesanan(Menu item)
100     {
101         Menu = item;
102     }
103     //Override method ToString untuk menampilkan nama dan harga menu
104     0 references
105     public override string ToString()
106     {
107         return $"{Menu.Nama} x{Jumlah} - Rp {Menu.Harga * Jumlah:N0}";
108     }
109 }
110 1 reference
111 private void btnSimpan_Click(object sender, EventArgs e)
112 {
113     // Cek apakah ada pesanan yang disimpan
114     if (orderItems.Count == 0)
115     {
116         MessageBox.Show("Tidak ada pesanan yang disimpan saat ini.", "Warning",
117             MessageBoxButtons.OK, MessageBoxIcon.Warning);
118         return;
119     }
120     // Mendapatkan nama pelanggan, jika kosong gunakan "Pelanggan"
121     string namapelanggan = string.IsNullOrEmpty(txtNama.Text)
122         ? "Pelanggan" : txtNama.Text.Trim();
123     // Menyimpan riwayat pesanan ke file teks
124     try
125     {
126         // Membuka file riwayat dalam mode append
127         using (StreamWriter writer = new StreamWriter(fileRiwayat, true))
128         {
129             // Menulis header riwayat pesanan
130             writer.WriteLine($" ".PadRight(70, ' '));
131             writer.WriteLine("                Riwayat Pesanan McDonald's");
132             writer.WriteLine($" ".PadRight(70, ' '));
133             writer.WriteLine($"Tanggal dan Jam: {DateTime.Now:dd/MM/yyyy HH:mm:ss}");
134             writer.WriteLine($"Pelanggan: {namapelanggan}");
135             writer.WriteLine($" ".PadRight(70, ' '));
136             writer.WriteLine("Detail Pesanan:");
137             writer.WriteLine();
138             int no = 1;

```

```

130
131
132 // Menulis setiap item pesanan ke file
133 foreach (var item in orderItems)
134 {
135     writer.WriteLine($"{no}. {item.Menu.Nama} x{item.Jumlah} - Rp {item.Menu.Harga:N0}");
136     no++;
137 }
138 decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
139 writer.WriteLine();
140 writer.WriteLine($"-".PadRight(70, '-'));
141 writer.WriteLine($" {"TOTAL".PadRight(40)} Rp {total:N0}");
142 writer.WriteLine($"=".PadRight(70, '='));
143 writer.WriteLine("        Terimakasih, Sampai berjumpa lagi!");
144 writer.WriteLine("        I'm lovin' it");
145 writer.WriteLine($"=".PadRight(70, '='));
146 writer.WriteLine();
147 writer.WriteLine();
148 }
149 MessageBox.Show($"pesanan berhasil di simpan !\n\nNama Pelanggan: {namapelanggan}\n" +
150     $"Total Pembayaran: Rp {orderItems.Sum(o => o.Menu.Harga * o.Jumlah):N0}",
151     "Sukses", MessageBoxButtons.OK, MessageBoxIcon.Information);
152
153 // Reset pesanan setelah disimpan
154 orderItems.Clear();
155 lstPesanan.Items.Clear();
156 txtNama.Clear();
157 cmbKategori.SelectedIndex = 0;
158 UpdateTotal();
159 }
160 // Menangani kesalahan saat menyimpan pesanan
161 catch (Exception ex)
162 {
163     MessageBox.Show($"Gagal menyimpan pesanan. Error: {ex.Message}", "Error",
164         MessageBoxButtons.OK, MessageBoxIcon.Error);
165 }
166
167 1 reference
168 private void cmbPilihMenu_SelectedIndexChanged(object sender, EventArgs e)
169 {
170 }
171
172 1 reference
173 private void cmbKategori_SelectedIndexChanged(object sender, EventArgs e)
174 {
175     // Memperbarui daftar menu berdasarkan kategori yang dipilih
176     cmbPilihMenu.Items.Clear();
177     cmbPilihMenu.Items.Add("=== Pilih Menu ===");
178
179     // Menampilkan semua menu jika kategori "Semua Kategori" dipilih
180     if (cmbKategori.SelectedIndex == 0)
181     {
182         // Tampilkan semua menu
183         foreach (var menu in menuItems)
184         {
185             cmbPilihMenu.Items.Add(menu);
186         }
187     }
188
189     // Menampilkan menu berdasarkan kategori yang dipilih
190     else
191     {
192         string selectedCategory = cmbKategori.SelectedItem.ToString();
193         var filteredMenu = menuItems.Where(m => m.Kategori == selectedCategory).ToList();
194         foreach (var menu in filteredMenu)
195         {
196             cmbPilihMenu.Items.Add(menu);
197         }
198     }
199     cmbPilihMenu.SelectedIndex = 0;

```



```

201 private void btnTambah_Click(object sender, EventArgs e)
202 {
203     // Menambahkan item pesanan ke daftar pesanan
204     if (cmbPilihMenu.SelectedIndex > 0)
205     {
206         Menu selectedMenu = (Menu)cmbPilihMenu.SelectedItem;
207         // Cek apakah item pesanan sudah ada dalam daftar
208         var existingItem = orderItems.FirstOrDefault(o => o.Menu.Id == selectedMenu.Id);
209         // Jika sudah ada, tambahkan jumlahnya
210         if (existingItem != null)
211         {
212             // Menambahkan jumlah item pesanan
213             existingItem.Jumlah++;
214             // Memperbarui tampilan item pesanan di ListBox
215             int index = lstPesanan.Items.IndexOf(existingItem);
216             lstPesanan.Items[index] = existingItem;
217         }
218         else
219         {
220             ItemPesanan itemPesanan = new ItemPesanan(selectedMenu);
221             orderItems.Add(itemPesanan);
222             lstPesanan.Items.Add(itemPesanan);
223         }
224
225         UpdateTotal();
226     }
227     // Jika tidak ada menu yang dipilih, tampilkan peringatan
228     else
229     {
230         MessageBox.Show("Silakan pilih menu terlebih dahulu.", "Peringatan",
231             MessageBoxButtons.OK, MessageBoxIcon.Warning);
232     }
233 }
234
235 1 reference
236 private void btnHapus_Click(object sender, EventArgs e)
237 {
238     // Menghapus item pesanan yang dipilih dari daftar pesanan
239     if (lstPesanan.SelectedIndex >= 0)
240     {
241         ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
242         orderItems.Remove(selectedItem);
243         lstPesanan.Items.Remove(selectedItem);
244         UpdateTotal();
245     }
246     // Jika tidak ada item yang dipilih, tampilkan peringatan
247     else
248     {
249         MessageBox.Show("Silakan pilih item pesanan yang akan dihapus.", "Peringatan",
250             MessageBoxButtons.OK, MessageBoxIcon.Warning);
251     }
252 }
253
254 // Memperbarui total harga pesanan
255 5 references
256 private void UpdateTotal()
257 {
258     // Menghitung total harga dari semua item pesanan
259     decimal total = orderItems.Sum(o => o.Menu.Harga * o.Jumlah);
260     lblTotal.Text = $"Total: Rp {total:N0}";
261 }
262
263 1 reference
264 private void btnReset_Click(object sender, EventArgs e)
265 {
266     // Mereset daftar pesanan setelah konfirmasi dari pengguna
267     if (orderItems.Count > 0)
268     {
269         // Menampilkan dialog konfirmasi
270         DialogResult result = MessageBox.Show("Apakah Anda yakin ingin mereset pesanan?",
271             "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question);
272         // Jika pengguna memilih "Yes", reset daftar pesanan
273         if (result == DialogResult.Yes)
274         {

```

```

270         {
271             // Menangani klik tombol checkout
272             orderItems.Clear();
273             lstPesanan.Items.Clear();
274             UpdateTotal();
275         }
276     }
277 }
278
279 // Membuka file riwayat pesanan dengan Notepad
280 1 reference
281 private void btnRiwayat_Click(object sender, EventArgs e)
282 {
283     // Memanggil method untuk membuka riwayat dengan Notepad
284     BukaRiwayat();
285 }
286
287 // Method untuk membuka file riwayat pesanan dengan Notepad
288 1 reference
289 private void BukaRiwayat()
290 {
291     // Cek apakah file riwayat ada
292     try
293     {
294         // Jika file tidak ada, tampilkan pesan informasi
295         if (!File.Exists(FileRiwayat))
296         {
297             MessageBox.Show("Belum ada riwayat pesanan.\n\nSilakan lakukan checkout terlebih dahulu.",
298                 "File Tidak Ditemukan",
299                 MessageBoxButtons.OK,
300                 MessageBoxIcon.Information);
301             return;
302         }
303         // Membuka file riwayat dengan Notepad
304         System.Diagnostics.Process.Start("notepad.exe", fileRiwayat);
305     }
306     // Menangani kesalahan saat membuka file riwayat
307     catch (Exception ex)
308     {
309         // Menampilkan pesan error jika gagal membuka file
310         MessageBox.Show($"Gagal membuka file riwayat!\n\nError: {ex.Message}",
311             "Error",
312             MessageBoxButtons.OK,
313             MessageBoxIcon.Error);
314     }
315 }
316
317 1 reference
318 private void Form1_Load_1(object sender, EventArgs e)
319 {
320 }
321
322 1 reference
323 private void btnKurangi_Click(object sender, EventArgs e)
324 {
325     // Mengurangi jumlah item pesanan yang dipilih
326     if (lstPesanan.SelectedIndex >= 0)
327     {
328         int selectedIndex = lstPesanan.SelectedIndex;
329         ItemPesanan selectedItem = (ItemPesanan)lstPesanan.SelectedItem;
330         // Cek apakah jumlah item lebih dari 1
331         if (selectedItem.Jumlah > 1)
332         {
333             // Kurangi jumlah item pesanan
334             selectedItem.Jumlah--;
335
336             // Update tampilan ListBox
337             lstPesanan.Items.Clear();
338             // tambah semua item pesanan kembali ke ListBox
339             foreach (var item in orderItems)
340             {
341                 lstPesanan.Items.Add(item);
342             }
343         }
344     }
345 }

```

```

339     }
340
341     lstPesanan.SelectedIndex = selectedIndex;
342 }
343 // Jika jumlah item adalah 1, hapus item dari daftar
344 else
345 {
346     orderItems.Remove(selectedItem);
347     lstPesanan.Items.Remove(selectedItem);
348 }
349
350     UpdateTotal();
351 }
352 else
353 {
354     MessageBox.Show("Silakan pilih item pesanan yang akan dikurangi.", "Peringatan",
355         MessageBoxButtons.OK, MessageBoxIcon.Warning);
356 }
357 }
358
359 1 reference
360 private void lstPesanan_SelectedIndexChanged(object sender, EventArgs e)
361 {
362 }
363 }
364
365 }

```

Code Program Form1.Designer

```
1 namespace projekAkhir_RoichanAryshaputra
2 {
3     4 references
4     partial class Form1
5     {
6         /// <summary>
7         /// Required designer variable.
8         /// </summary>
9         private System.ComponentModel.IContainer components = null;
10
11         /// <summary>
12         /// Clean up any resources being used.
13         /// </summary>
14         /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
15         0 references
16         protected override void Dispose(bool disposing)
17         {
18             if (disposing && (components != null))
19             {
20                 components.Dispose();
21             }
22             base.Dispose(disposing);
23         }
24
25         1 reference
26         private void InitializeComponent()
27         {
28             System.ComponentModel.ComponentResourceManager resources = new System.ComponentModel.ComponentResourceManager(typeof(Form1));
29             lblJudul = new Label();
30             lblNama = new Label();
31             txtNama = new TextBox();
32             lblKategori = new Label();
33             cmbKategori = new ComboBox();
34             lblPilihMenu = new Label();
35             cmbPilihMenu = new ComboBox();
36             btnTambah = new Button();
37             lblPesanan = new Label();
38             lstPesanan = new ListBox();
39             btnHapus = new Button();
40             btnReset = new Button();
41             lblTotal = new Label();
42             btnSimpan = new Button();
43             btnRiwayat = new Button();
44             btnKurangi = new Button();
45             SuspendLayout();
46             //
47             // lblJudul
48             //
49             lblJudul.AutoSize = true;
50             lblJudul.Font = new Font("Bahnschrift", 19.8000011F, FontStyle.Bold, GraphicsUnit.Point, 0);
51             lblJudul.ForeColor = Color.FromArgb(255, 188, 13);
52             lblJudul.Location = new Point(58, 9);
53             lblJudul.Name = "lblJudul";
54             lblJudul.Size = new Size(458, 41);
55             lblJudul.TabIndex = 0;
56             lblJudul.Text = "Selamat Datang di McDonald'";
```

```

53 //
54 // lblNama
55 //
56 lblNama.Font = new Font("Arial", 10.2F, FontStyle.Bold, GraphicsUnit.Point, 0);
57 lblNama.ForeColor = Color.FromArgb(255, 188, 13);
58 lblNama.Location = new Point(30, 100);
59 lblNama.Name = "lblNama";
60 lblNama.Size = new Size(62, 25);
61 lblNama.TabIndex = 1;
62 lblNama.Text = "Nama";
63 //
64 // txtNama
65 //
66 txtNama.Location = new Point(160, 100);
67 txtNama.Name = "txtNama";
68 txtNama.Size = new Size(270, 28);
69 txtNama.TabIndex = 2;
70 //
71 // lblKategori
72 //
73 lblKategori.Font = new Font("Arial", 10.2F, FontStyle.Bold, GraphicsUnit.Point, 0);
74 lblKategori.ForeColor = Color.FromArgb(255, 188, 13);
75 lblKategori.Location = new Point(30, 145);
76 lblKategori.Name = "lblKategori";
77 lblKategori.Size = new Size(100, 25);
78 lblKategori.TabIndex = 3;
79 lblKategori.Text = "Kategori";
80 //
81 // cmbKategori
82 //
83 cmbKategori.DropDownStyle = ComboBoxStyle.DropDownList;
84 cmbKategori.Font = new Font("Arial", 10.2F, FontStyle.Regular, GraphicsUnit.Point, 0);
85 cmbKategori.FormattingEnabled = true;
86 cmbKategori.Location = new Point(160, 145);
87 cmbKategori.Name = "cmbKategori";
88 cmbKategori.Size = new Size(270, 27);
89 cmbKategori.TabIndex = 4;
90 cmbKategori.SelectedIndexChanged += cmbKategori_SelectedIndexChanged;
91 //
92 // lblPilihMenu
93 //
94 lblPilihMenu.Font = new Font("Arial", 10.2F, FontStyle.Bold, GraphicsUnit.Point, 0);
95 lblPilihMenu.ForeColor = Color.FromArgb(255, 188, 13);
96 lblPilihMenu.Location = new Point(30, 190);
97 lblPilihMenu.Name = "lblPilihMenu";
98 lblPilihMenu.Size = new Size(100, 25);
99 lblPilihMenu.TabIndex = 5;
100 lblPilihMenu.Text = "Pilih Menu";
101 //
102 // cmbPilihMenu
103 //
104 cmbPilihMenu.DropDownStyle = ComboBoxStyle.DropDownList;
105 cmbPilihMenu.Font = new Font("Arial", 10.2F, FontStyle.Regular, GraphicsUnit.Point, 0);
106 cmbPilihMenu.FormattingEnabled = true;
107 cmbPilihMenu.Location = new Point(30, 215);
108 cmbPilihMenu.Name = "cmbPilihMenu";
109 cmbPilihMenu.Size = new Size(440, 27);
110 cmbPilihMenu.TabIndex = 6;
111 cmbPilihMenu.SelectedIndexChanged += cmbPilihMenu_SelectedIndexChanged;

```

```
112 //
113 // btnTambah
114 //
115 btnTambah.BackColor = Color.FromArgb(218, 41, 28);
116 btnTambah.Cursor = Cursors.Hand;
117 btnTambah.FlatStyle = FlatStyle.Flat;
118 btnTambah.Location = new Point(476, 209);
119 btnTambah.Name = "btnTambah";
120 btnTambah.Size = new Size(102, 35);
121 btnTambah.TabIndex = 7;
122 btnTambah.Text = "Tambah";
123 btnTambah.UseVisualStyleBackColor = false;
124 btnTambah.Click += btnTambah_Click;
125 //
126 // lblPesanan
127 //
128 lblPesanan.AutoSize = true;
129 lblPesanan.Font = new Font("Arial", 12F, FontStyle.Bold, GraphicsUnit.Point, 0);
130 lblPesanan.ForeColor = Color.FromArgb(41, 41, 41);
131 lblPesanan.Location = new Point(30, 265);
132 lblPesanan.Name = "lblPesanan";
133 lblPesanan.Size = new Size(173, 24);
134 lblPesanan.TabIndex = 8;
135 lblPesanan.Text = "\U0001f6d2Pesanan Kamu";
136 //
137 // lstPesanan
138 //
139 lstPesanan.BackColor = Color.White;
140 lstPesanan.BorderStyle = BorderStyle.FixedSingle;
141 lstPesanan.Font = new Font("Arial", 10.2F, FontStyle.Regular, GraphicsUnit.Point, 0);
142 lstPesanan.FormattingEnabled = true;
143 lstPesanan.ItemHeight = 19;
144 lstPesanan.Location = new Point(30, 295);
145 lstPesanan.Name = "lstPesanan";
146 lstPesanan.Size = new Size(580, 173);
147 lstPesanan.TabIndex = 9;
148 lstPesanan.SelectedIndexChanged += lstPesanan_SelectedIndexChanged;
149 //
150 // btnHapus
151 //
152 btnHapus.BackColor = Color.FromArgb(239, 68, 68);
153 btnHapus.Cursor = Cursors.Hand;
154 btnHapus.FlatStyle = FlatStyle.Flat;
155 btnHapus.Font = new Font("Arial", 9F, FontStyle.Bold, GraphicsUnit.Point, 0);
156 btnHapus.Location = new Point(30, 485);
157 btnHapus.Name = "btnHapus";
158 btnHapus.Size = new Size(140, 38);
159 btnHapus.TabIndex = 10;
160 btnHapus.Text = "x Hapus";
161 btnHapus.UseVisualStyleBackColor = false;
162 btnHapus.Click += btnHapus_Click;
```

```

163 //
164 // btnReset
165 //
166 btnReset.BackColor = Color.FromArgb(107, 114, 128);
167 btnReset.Cursor = Cursors.Hand;
168 btnReset.FlatStyle = FlatStyle.Flat;
169 btnReset.Font = new Font("Arial", 9F, FontStyle.Bold, GraphicsUnit.Point, 0);
170 btnReset.Location = new Point(184, 485);
171 btnReset.Name = "btnReset";
172 btnReset.Size = new Size(140, 38);
173 btnReset.TabIndex = 11;
174 btnReset.Text = "Reset";
175 btnReset.UseVisualStyleBackColor = false;
176 btnReset.Click += btnReset_Click;
177 //
178 // lblTotal
179 //
180 lblTotal.Location = new Point(30, 538);
181 lblTotal.Name = "lblTotal";
182 lblTotal.Size = new Size(580, 50);
183 lblTotal.TabIndex = 12;
184 lblTotal.Text = "TOTAL: Rp 0";
185 lblTotal.TextAlign = ContentAlignment.MiddleCenter;
186 //
187 // btnSimpan
188 //
189 btnSimpan.BackColor = Color.FromArgb(34, 197, 94);
190 btnSimpan.Cursor = Cursors.Hand;
191 btnSimpan.FlatStyle = FlatStyle.Flat;
192 btnSimpan.Location = new Point(30, 594);
193 btnSimpan.Name = "btnSimpan";
194 btnSimpan.Size = new Size(280, 45);
195 btnSimpan.TabIndex = 13;
196 btnSimpan.Text = "💰 CHECKOUT";
197 btnSimpan.UseVisualStyleBackColor = false;
198 btnSimpan.Click += btnSimpan_Click;
199 //
200 // btnRiwayat
201 //
202 btnRiwayat.BackColor = Color.FromArgb(255, 188, 13);
203 btnRiwayat.Cursor = Cursors.Hand;
204 btnRiwayat.FlatStyle = FlatStyle.Flat;
205 btnRiwayat.ForeColor = Color.FromArgb(41, 41, 41);
206 btnRiwayat.Location = new Point(340, 591);
207 btnRiwayat.Name = "btnRiwayat";
208 btnRiwayat.Size = new Size(280, 45);
209 btnRiwayat.TabIndex = 14;
210 btnRiwayat.Text = "📅 Riwayat";
211 btnRiwayat.UseVisualStyleBackColor = false;
212 btnRiwayat.Click += btnRiwayat_Click;
213 //
214 // btnKurangi
215 //
216 btnKurangi.BackColor = Color.FromArgb(107, 114, 128);
217 btnKurangi.Cursor = Cursors.Hand;
218 btnKurangi.FlatStyle = FlatStyle.Flat;
219 btnKurangi.Font = new Font("Arial", 9F, FontStyle.Bold, GraphicsUnit.Point, 0);
220 btnKurangi.Location = new Point(338, 485);
221 btnKurangi.Name = "btnKurangi";
222 btnKurangi.Size = new Size(140, 38);
223 btnKurangi.TabIndex = 15;

```



```

224         btnKurangi.Text = Kurangi;
225         btnKurangi.UseVisualStyleBackColor = false;
226         btnKurangi.Click += btnKurangi_Click;
227         //
228         // Form1
229         //
230         BackColor = Color.FromArgb(218, 41, 28);
231         ClientSize = new Size(632, 653);
232         Controls.Add(btnKurangi);
233         Controls.Add(btnRiwayat);
234         Controls.Add(btnSimpan);
235         Controls.Add(lblTotal);
236         Controls.Add(btnReset);
237         Controls.Add(btnHapus);
238         Controls.Add(lstPesanan);
239         Controls.Add(lblPesanan);
240         Controls.Add(btnTambah);
241         Controls.Add(cmbPilihMenu);
242         Controls.Add(lblPilihMenu);
243         Controls.Add(cmbKategori);
244         Controls.Add(lblKategori);
245         Controls.Add(txtNama);
246         Controls.Add(lblNama);
247         Controls.Add(lblJudul);
248         Font = new Font("Arial", 10.8F, FontStyle.Bold, GraphicsUnit.Point, 0);
249         ForeColor = Color.White;
250         Icon = (Icon)resources.GetObject("$this.Icon");
251         Name = "Form1";
252         StartPosition = FormStartPosition.CenterScreen;
253         Text = "Menu Pesanan McDonald's";
254         Load += Form1_Load_1;
255         ResumeLayout(false);
256         PerformLayout();
257     }
258     private Label lblJudul;
259     private Label lblNama;
260     private TextBox txtNama;
261     private Label lblKategori;
262     private ComboBox cmbKategori;
263     private Label lblPilihMenu;
264     private ComboBox cmbPilihMenu;
265     private Button btnTambah;
266     private Label lblPesanan;
267     private ListBox lstPesanan;
268     private Button btnHapus;
269     private Button btnReset;
270     private Label lblTotal;
271     private Button btnSimpan;
272     private Button btnRiwayat;
273     private Button btnKurangi;
274 }
275 }
276

```


Screenshot Hasil Program

 Menu Pesanan McDonald's

Selamat Datang di McDonald'

Nama

Roichan

Kategori

Minuman

Pilih Menu

Coca-Cola (M) - Rp 12,000

Tambah

 **Pesanan Kamu**

PaMer 7 Ayam McD Gulai Krispy x2 - Rp 397,000
Coca-Cola (M) x2 - Rp 24,000

× Hapus

Reset

Kurangi

Total: Rp 421,000

 **CHECKOUT**

 **Riwayat**