

**LAPORAN HASIL PROYEK AKHIR
SEMESTER GANGSAL**



Disusun oleh :

Nama : Rumaisa
Kelas : X PPLG 2
Nis : 3096345510
Judul Proyek : Pemesanan Hotel

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

DESKRIPSI PROYEK

Proyek ini bertujuan untuk membuat “**Aplikasi Pemesanan hotel**” berbasis desktop menggunakan C# Windows Forms yang dapat menampilkan informasi hotel dan cara pemesanan hotel jarak jauh untuk memudahkan pelayanan dari hotel tidak secara langsung atau berbasis online. Aplikasi ini memudahkan pegawai hotel maupun konsumen untuk pemesanan/booking kamar di seluruh dunia .

Fitur Utama Aplikasi:

1. Pencarian & Filter Canggih

- Pengguna bisa mencari hotel dengan kriteria seperti lokasi, harga, fasilitas (wifi, kolam renang, gym), tipe kamar, rating, jarak ke landmark, dll.
- Autocomplete di kolom pencarian (misal kota, nama hotel) supaya lebih cepat.

2. Ketersediaan Real-Time & Harga Dinamis

- Data kamar dan harga diperbarui secara langsung, sehingga tidak terjadi “overbooking” atau informasi yang salah.
- Harga bisa berubah sesuai musim, permintaan, promosi, dan kebijakan hotel.

3. Informasi Hotel Detail

- Menampilkan deskripsi lengkap: tipe kamar, fasilitas, kebijakan, foto, video, dan tur virtual (360°) untuk membangun kepercayaan user.
- Rating dan review dari pengguna lain supaya calon tamu bisa mempertimbangkan kualitas hotel.

4. Pembayaran dan Gateway Keamanan

- Integrasi berbagai metode pembayaran (kartu kredit, dompet digital, metode lokal).
- Dukungan multi-mata uang (untuk pengguna internasional).
- Layar review & bayar, agar pengguna bisa cek ulang sebelum konfirmasi pembayaran.

5. Manajemen Reservasi

- Konfirmasi reservasi secara instan.
- Kemampuan untuk ubah atau batalkan booking langsung lewat aplikasi (sesuai kebijakan hotel).

6. Notifikasi (push Notifications)

- Notifikasi konfirmasi, pengingat check-in atau check-out, promosi diskon, atau pengumuman penting.
- Bisa diatur supaya pengguna bisa matikan notifikasi jika tidak ingin terganggu.

7. Papan Ulasan & Rating

- Pengguna bisa memberi review setelah menginap.
- Review membantu user lain dalam pengambilan keputusan.

Konsep Pemrograman yang Diimplementasikan:

- 1. CRUD (Create, Read, Update, Delete)** - Update data reservasi atau status kamar.
- 2. Pemrograman Berorientasi Objek (OOP)** - Class seperti Hotel, Kamar, Reservasi, User.
- 3. Validasi Input & Logika Bisnis** - Mengecek ketersediaan kamar saat memesan.
- 4. Penanganan Event (Event-Driven Programming)** - Klik tombol “Cari”, “Pesan”, memilih tanggal, memilih kamar.

5. **Manajemen Sesi & Otentikasi** - Session untuk menyimpan preferensi user atau daftar favorit.
6. **Integrasi API & Pembayaran** - API peta/lokasi agar pengguna bisa melihat posisi hotel.
7. **Error Handling dan exception Management** - Menampilkan pesan kesalahan ke pengguna dengan jelas.
8. **UI/UX & Navigasi yang Efisien** - Tampilan detail kamar dengan foto.

RINGKASAN TEORI C# YANG DIGUNAKAN:

1. Pemrograman Berorientasi Objek (Object-Oriented Programming / OOP)

C# adalah bahasa pemrograman yang berbasis OOP. Beberapa konsep OOP yang digunakan:

Class dan Object → untuk membuat objek seperti Hotel, Kamar, User, Reservasi.

Encapsulation → menyembunyikan data dalam class dan mengaksesnya melalui metode.

Inheritance → contoh: Kamar Deluxe mewarisi Kamar. Polymorphism → metode seperti HitungHarga() bisa berbeda pada tiap tipe kamar.

2. Event-Driven Programming (Pemrograman Berbasis Event)

Pada aplikasi Windows Forms, program berjalan berdasarkan event dari pengguna:

Klik tombol Cari, Pesan, Simpan, Hapus

Perubahan pilihan combobox

Input teks oleh user

Di C#, event ini ditangani menggunakan event handler, seperti:

```
private void btnPesan_Click(object sender, EventArgs e)
```

```
{
```

```
    // Perintah saat tombol ditekan
```

```
}
```

3. Struktur Kontrol (Control Statement)

Digunakan untuk logika pemesanan:

If/else → cek ketersediaan kamar

Switch → memilih tipe hotel / tipe kamar

Loop (for, while) → menampilkan daftar kamar/hotel

4. Variabel dan Tipe Data

Digunakan untuk menyimpan informasi seperti:

Nama hotel (string)

Harga kamar (int / double)

Tanggal check-in check-out (DateTime)

Jumlah malam (int)

5. Method/Fungsi

Method dibuat untuk:

Menghitung harga total

Menampilkan daftar hotel

Menyimpan data pemesanan

Contoh:

```
double HitungTotal(double hargaPerMalam, int malam)
{
    return hargaPerMalam * malam;
}
```

6. Manipulasi Data & Collection (List)

C# menggunakan List<T> untuk menyimpan banyak data, contoh daftar kamar atau daftar hotel:

```
List<Hotel> daftarHotel = new List<Hotel>();
```

7. Exception Handling

Digunakan untuk menangani kesalahan, misalnya input salah atau data tidak ditemukan:

```
try
{
    // kode yang berpotensi error
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
```

8. Database Connection (SQL Connection)

C# sering menghubungkan aplikasi ke database menggunakan:

SqlConnection

SqlCommand

SqlDataAdapter

Contoh sederhana:

```
SqlConnection conn = new SqlConnection(koneksi);
conn.Open();
```

9. Binding Data ke UI

C# Windows Forms menggunakan DataGridView, ComboBox, dan ListView untuk menampilkan data hotel, kamar, atau pemesanan.

10. Namespace dan Library C#

Digunakan untuk mengakses fitur penting:

System → tipe data dasar

System.Data → mengolah data

System.Windows.Forms → membuat tampilan form

System.Data.SqlClient → koneksi SQL Server

✓ Ringkasannya

Dalam aplikasi pemesanan hotel, teori C# yang digunakan meliputi:

OOP, event-driven programming, kontrol logika, manipulasi data, penggunaan database, method, exception handling, dan komponen UI Windows Forms.

PERENCANAAN PROGRAM

Struktur Class / Folder

```
HotelBookingApp
├── FormMain.cs (Main Form)
├── FormBooking.cs (Form input pemesanan)
├── FormListBooking.cs (Tabel daftar pemesanan)
└── Models
    ├── Hotel.cs
    ├── Room.cs
    ├── Booking.cs
    └── Customer.cs
└── Database
    └── DbManager.cs (CRUD database / file JSON)
└── Helpers
    └── PriceCalculator.cs (Penghitungan total harga)
```

Alur Kerja Program

1. **User Input** - User memilih: Nama hotel, Jenis kamar, Tanggal check-in & check-out, Data diri tamu, Input diambil dari TextBox, ComboBox, dan DateTimePicker.
2. **Validation** - Program mengecek: Input kosong, Tanggal check-out harus lebih besar dari check-in, Room tersedia
3. **Load Data Hotel & Room** -
 - o Data dapat disimpan dalam:
 - o Database lokal (SQLite)
 - o Atau file JSON (mengikuti contoh video)
 - o Program melakukan:
 DbManager.LoadHotels()
 DbManager.LoadRooms()
4. **Hitung Harga** - Menggunakan helper:
 PriceCalculator.CalculateTotal(room.pricePerNight, nights)
5. **Save Booking** - Program menyimpan data pemesanan ke database/JSON:
 DbManager.SaveBooking(booking)
6. **Display Data** - Data ditampilkan pada:
 - o Label → Nama hotel, tipe kamar, harga
 - o DataGridView → Riwayat pemesanan
 - o PictureBox → Foto hotel (jika ada)
7. **Edit / Delete Booking (CRUD)** - Dari DataGridView:
 - o User bisa klik Edit → memunculkan FormBooking
 - o Klik Delete → menghapus data pemesanan

IMPLEMENTASI PROGRAM

1. Form1.cs - MAin Application

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Projek_Hotel
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {

        }

        private void Form1_Load(object sender, EventArgs e)
        {

        }

        private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
        {
            // Menampilkan harga berdasarkan jenis kamar yang dipilih
            if (cbJenis.SelectedItem.ToString() == "VIP")
            {
                txtHarga.Text = "750000";
            }
            else if (cbJenis.SelectedItem.ToString() == "Kelas I")
            {
                txtHarga.Text = "500000";
            }
            else if (cbJenis.SelectedItem.ToString() == "Kelas II")
```

```

        {
            txtHarga.Text = "300000";
        }
        else if (cbJenis.SelectedItem.ToString() == "Kelas III")
        {
            txtHarga.Text = "150000";
        }
    }

private void txtLama_TextChanged(object sender, EventArgs e)
{
    // Biaya Total dihitung berdasarkan lama menginap dan harga kamar
    if (txtLama.Text != "" && txtHarga.Text != "")
    {
        int lama = int.Parse(txtLama.Text);
        int harga = int.Parse(txtHarga.Text);
        int total = lama * harga;
        txtBiaya.Text = total.ToString();
    }
    else
    {
        txtBiaya.Text = "";
    }
}

private void txtJumlah_TextChanged(object sender, EventArgs e)
{
    // Menghitung kembalian berdasarkan jumlah bayar dan biaya total
    if (txtJumlah.Text != "" && txtLama.Text != "")
    {
        int jumlah = int.Parse(txtJumlah.Text);
        int biaya = int.Parse(txtBiaya.Text);
        int kembalian = jumlah - biaya;
        txtKembali.Text = kembalian.ToString();
    }
    else
    {
        txtKembali.Text = "";
    }
}

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

```

```

// Menghitung biaya total ketika harga kamar berubah
if (txtLama.Text != "" && txtHarga.Text != "")
{
    int lama = int.Parse(txtLama.Text);
    int harga = int.Parse(txtHarga.Text);
    int total = lama * harga;
    txtLama.Text = total.ToString();
}
else
{
    txtLama.Text = "";
}
}

private void txtBiaya_TextChanged(object sender, EventArgs e)
{
    // Menghitung kembalian ketika biaya total berubah
    if (txtJumlah.Text != "" && txtLama.Text != "")
    {
        int jumlah = int.Parse(txtJumlah.Text);
        int total = int.Parse(txtLama.Text);
        int kembalian = jumlah - total;
        txtKembali.Text = kembalian.ToString();
    }
    else
    {
        txtKembali.Text = "";
    }
}

private void btClose_Click(object sender, EventArgs e)
{
    // Menutup aplikasi
    Application.Exit();
}

private void btNew_Click(object sender, EventArgs e)
{
    // Membersihkan semua inputan pada form
    txtNama.Text = "";
    cbJenis.Text = "";
    txtHarga.Text = "";
    txtLama.Text = "";
    txtBiaya.Text = "";
}

```

```

        txtJumlah.Text = "";
        txtKembali.Text = "";
        txtAlamat.Text = "";
        txtNomor.Text = "";
        dateTimePicker1.Value = DateTime.Now;
    }

    private void groupBox1_Enter(object sender, EventArgs e)
    {
        }
    }
}

```

PENJELASAN FUNGSI MASING-MASING BAGIAN

cbJenis_SelectedIndexChanged :

Tujuan: menetapkan harga otomatis sesuai pilihan jenis kamar.

Alur: saat user memilih jenis, event ini aktif → set txtHarga.Text. Menggunakan switch memudahkan pemeliharaan bila nanti ingin menambah tipe kamar.

txtLama_TextChanged :

Tujuan: menghitung total biaya ($lama \times harga$) secara real-time saat user mengetik jumlah hari menginap.

Validasi: gunakan TryParse untuk memastikan input numerik. Jika invalid, kosongkan atau beri pesan. Mengupdate txtBiaya.

txtHarga_TextChanged :

Tujuan: jika harga dirubah manual (mis. admin edit), perbarui total biaya. Menerapkan validasi sama.

txtBiaya_TextChanged :

Tujuan: setiap perubahan biaya total harus memicu perhitungan ulang kembalian jika pengguna sudah memasukkan jumlah bayar.

Catatan: jangan gunakan txtLama di sini; sebelumnya itu bug.

txtJumlah_TextChanged :

Tujuan: menghitung kembalian (jumlah bayar - total biaya) secara otomatis setelah user mengetik nominal pembayaran. Gunakan validasi untuk menghindari crash.

btNew_Click :

Reset semua input ke keadaan awal. Juga mengembalikan DateTimePicker ke waktu sekarang.

btClose_Click :

Keluar aplikasi

FUNGSI BAGIAN - BAGIAN KODE PROGRAM

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

Penjelasan fungsi coding :

1. using System;
Memanggil library dasar C# untuk fungsi umum (contoh: Console, Math, tipe data, dll).
2. using System.Collections.Generic;
Untuk menggunakan koleksi data seperti List<T>, Dictionary< TKey, TValue >, dll.
3. using System.ComponentModel;
Digunakan untuk komponen desain/form seperti backgroundworker, binding, dan properti kontrol.
4. using System.Data;
Digunakan untuk pengolahan data, seperti DataSet, DataTable, koneksi database, dsb.
5. using System.Drawing;
Digunakan untuk gambar, warna, ukuran, font, dan pengaturan grafis lainnya pada form.
6. using System.Linq;
Memungkinkan kamu memakai fitur LINQ untuk query data (contoh: Where, Select, OrderBy).
7. using System.Text;
Untuk manipulasi teks dan encoding, termasuk StringBuilder.
8. using System.Threading.Tasks;
Untuk menjalankan proses secara asynchronous (async/await).
9. using System.Windows.Forms;
Library utama untuk membuat aplikasi Windows Forms (form, button, textbox, event klik, dll).

```

private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
{
    // Menampilkan harga berdasarkan jenis kamar yang dipilih
    if (cbJenis.SelectedItem.ToString() == "VIP")
    {
        txtHarga.Text = "750000";
    }
    else if (cbJenis.SelectedItem.ToString() == "Kelas I")
    {
        txtHarga.Text = "500000";
    }
    else if (cbJenis.SelectedItem.ToString() == "Kelas II")
    {
        txtHarga.Text = "300000";
    }
    else if (cbJenis.SelectedItem.ToString() == "Kelas III")
    {
        txtHarga.Text = "150000";
    }
}

```

Penjelasan fungsi coding :

cbJenis_SelectedIndexChanged

Kode ini adalah event handler yang akan berjalan setiap kali pilihan di ComboBox cbJenis berubah.

Tujuan utamanya:

Untuk menampilkan harga kamar otomatis ke dalam txtHarga sesuai jenis kamar yang dipilih pada cbJenis.

Cara kerjanya:

1. User memilih jenis kamar di ComboBox cbJenis.

2. Event SelectedIndexChanged dipicu.

3. Program mengecek jenis kamar yang dipilih:

Jika VIP → harga = 750000

Jika Kelas I → harga = 500000

Jika Kelas II → harga = 300000

Jika Kelas III → harga = 150000

4. Harga ditampilkan di TextBox txtHarga.

Fungsi Lengkapnya:

Mempermudah user mengisi form.

Menghindari kesalahan input harga.

Menghubungkan ComboBox dan TextBox secara otomatis.

```

private void txtLama_TextChanged(object sender, EventArgs e)
{
    // Biaya Total dihitung berdasarkan lama menginap dan harga kamar
    if (txtLama.Text != "" && txtHarga.Text != "")
    {
        int lama = int.Parse(txtLama.Text);
        int harga = int.Parse(txtHarga.Text);
        int total = lama * harga;
        txtBiaya.Text = total.ToString();
    }
    else
    {
        txtBiaya.Text = "";
    }
}

```

Penjelasan Fungsi coding :

txtLama_TextChanged

Fungsi ini akan berjalan setiap kali isi textbox txtLama berubah, misalnya ketika user mengetik angka lama menginap.

Tujuan:

Menghitung Biaya Total berdasarkan:

Lama menginap (txtLama)

Harga kamar (txtHarga)

Lalu hasilnya ditampilkan pada txtBiaya

Penjelasan Baris per Baris :

1. Trigger Event

```
private void txtLama_TextChanged(object sender, EventArgs e)
Event ini aktif ketika teks pada txtLama berubah.
```

2. Mengecek apakah input terisi

```
if (txtLama.Text != "" && txtHarga.Text != "")
```

Artinya:

Hitungan total hanya dilakukan jika:

txtLama sudah diisi

txtHarga sudah diisi

Kalau salah satu kosong → tidak bisa hitung.

3. Mengambil nilai dan mengubah menjadi integer

```
int lama = int.Parse(txtLama.Text);
```

```
int harga = int.Parse(txtHarga.Text);
```

int.Parse() mengubah teks menjadi angka.

lama = jumlah hari menginap

harga = harga kamar per hari

4. Menghitung total biaya
int total = lama * harga;
Perhitungan:
> Total biaya = lama menginap × harga kamar
5. Menampilkan hasil ke textbox biaya
txtBiaya.Text = total.ToString();
Mengubah total menjadi teks
Lalu ditampilkan di txtBiaya
6. Jika input kosong → biaya dikosongkan
else
{
 txtBiaya.Text = "";
}
Jika user menghapus input, maka total biaya ikut dihapus.

```
private void txtJumlah_TextChanged(object sender, EventArgs e)
{
    // Menghitung kembalian berdasarkan jumlah bayar dan biaya total
    if (txtJumlah.Text != "" && txtLama.Text != "")
    {
        int jumlah = int.Parse(txtJumlah.Text);
        int biaya = int.Parse(txtBiaya.Text);
        int kembalian = jumlah - biaya;
        txtKembali.Text = kembalian.ToString();
    }
    else
    {
        txtKembali.Text = "";
    }
}
```

Penjelasan Fungsi coding :

txtJumlah_TextChanged

Fungsi ini akan berjalan setiap kali isi TextBox txtJumlah berubah (misalnya saat user mengetik angka pada kolom Jumlah Bayar).

1. Mengecek apakah kedua textbox sudah terisi

if (txtJumlah.Text != "" && txtLama.Text != "")

Artinya:

Perhitungan hanya dilakukan jika txtJumlah tidak kosong, dan

txtLama tidak kosong (meskipun sebenarnya variabel yang dipakai adalah txtBiaya, bukan txtLama).

> Ini untuk mencegah error ketika melakukan int.Parse pada textbox yang kosong.

2. Mengubah teks menjadi angka

```
int jumlah = int.Parse(txtJumlah.Text);
```

```
int biaya = int.Parse(txtBiaya.Text);
```

Mengambil nilai dari TextBox txtJumlah → dianggap sebagai jumlah uang yang dibayar.

Mengambil nilai dari TextBox txtBiaya → dianggap sebagai biaya total yang harus dibayar.

int.Parse digunakan agar teks bisa dihitung secara numerik.

3. Menghitung kembalian

```
int kembalian = jumlah - biaya;
```

Rumusnya:

> Kembalian = Jumlah yang dibayar – Total biaya

Jika user membayar lebih besar, hasilnya positif.

Jika kurang, hasilnya negatif.

4. Menampilkan hasil ke textbox kembalian

```
txtKembali.Text = kembalian.ToString();
```

Mengubah angka kembalian menjadi teks, lalu menampilkannya di txtKembali.

5. Jika textbox kosong, kosongkan juga kembalian

```
else
```

```
{
```

```
    txtKembali.Text = "";
```

```
}
```

Agar kolom kembalian tidak menampilkan nilai lama ketika textbox kosong.

```
private void txtHarga_TextChanged(object sender, EventArgs e)
{
    // Menghitung biaya total ketika harga kamar berubah
    if (txtLama.Text != "" && txtHarga.Text != "")
    {
        int lama = int.Parse(txtLama.Text);
        int harga = int.Parse(txtHarga.Text);
        int total = lama * harga;
        txtLama.Text = total.ToString();
    }
    else
    {
        txtLama.Text = "";
    }
}
```

Penjelasan Fungsi coding :

txtHarga_TextChanged

Fungsi ini dijalankan setiap kali isi textbox harga (txtHarga) berubah.

```
private void txtHarga_TextChanged(object sender, EventArgs e)
{
    // Menghitung biaya total ketika harga kamar berubah
    if (txtLama.Text != "" && txtHarga.Text != "")
    {
        int lama = int.Parse(txtLama.Text);
        int harga = int.Parse(txtHarga.Text);
        int total = lama * harga;
        txtLama.Text = total.ToString();
    }
    else
    {
        txtLama.Text = "";
    }
}
```

Penjelasan Baris per Baris

1. Event Handler

```
private void txtHarga_TextChanged(object sender, EventArgs e)
```

Fungsi ini otomatis dipanggil ketika isi textbox harga kamar berubah.

2. Mengecek apakah kedua textbox terisi

```
if (txtLama.Text != "" && txtHarga.Text != "")
```

Supaya tidak terjadi error FormatException, program memastikan:

txtLama tidak kosong

txtHarga tidak kosong

3. Mengubah teks menjadi angka

```
int lama = int.Parse(txtLama.Text);
```

```
int harga = int.Parse(txtHarga.Text);
```

Mengubah teks menjadi tipe data integer agar bisa dihitung.

4. Menghitung total biaya

```
int total = lama * harga;
```

Total biaya = lama menginap × harga kamar per malam.

```

private void txtBiaya_TextChanged(object sender, EventArgs e)
{
    // Menghitung kembalian ketika biaya total berubah
    if (txtJumlah.Text != "" && txtLama.Text != "")
    {
        int jumlah = int.Parse(txtJumlah.Text);
        int total = int.Parse(txtLama.Text);
        int kembalian = jumlah - total;
        txtKembali.Text = kembalian.ToString();
    }
    else
    {
        txtKembali.Text = "";
    }
}

```

**Penjelasan Fungsi
txtBiaya_TextChanged**

Fungsi ini akan berjalan setiap kali isi textbox biaya total (txtBiaya) berubah.

Tetapi... di dalam kodennya kamu memakai txtLama sebagai total, bukan txtBiaya. Nanti aku jelaskan bagian salahnya ya.

Berikut kodenya:

```

private void txtBiaya_TextChanged(object sender, EventArgs e)
{
    // Menghitung kembalian ketika biaya total berubah
    if (txtJumlah.Text != "" && txtLama.Text != "")
    {
        int jumlah = int.Parse(txtJumlah.Text);
        int total = int.Parse(txtLama.Text);
        int kembalian = jumlah - total;
        txtKembali.Text = kembalian.ToString();
    }
    else
    {
        txtKembali.Text = "";
    }
}

```

**Penjelasan Fungsi coding :
txtBiaya_TextChanged**

Fungsi ini akan berjalan setiap kali isi textbox biaya total (txtBiaya) berubah.

Penjelasan Baris per Baris :

1. Event Handler

```
private void txtBiaya_TextChanged(object sender, EventArgs e)
```

Fungsi ini dipanggil ketika textbox Biaya Total berubah isi.

2. Mengecek apakah input terisi semua

```
if (txtJumlah.Text != "" && txtLama.Text != "")
```

3. Parsing / mengubah teks jadi angka

```
int jumlah = int.Parse(txtJumlah.Text);
```

```
int total = int.Parse(txtLama.Text);
```

jumlah: uang yang dibayar

total: biaya total (TAPI DIAMBIL DARI txtLama → SALAH)

4. Menghitung kembalian

```
int kembalian = jumlah - total;
```

rumus kembalian = jumlah bayar – total biaya

5. Menampilkan ke textbox kembalian

```
txtKembali.Text = kembalian.ToString();
```

6. Jika input kosong

```
else
```

```
{
```

```
    txtKembali.Text = "";
```

```
}
```

Mengosongkan textbox kembalian.

```
private void btClose_Click(object sender, EventArgs e)
```

```
{
```

// Menutup aplikasi

```
Application.Exit();
```

```
}
```

Penjelasan Fungsi coding :

1. Event Handler tombol

```
private void btClose_Click(object sender, EventArgs e)
```

Fungsi ini dijalankan ketika tombol btClose ditekan.

object sender

= objek yang memicu event (yaitu tombol btClose).

EventArgs e

= informasi event klik.

2. Menutup aplikasi

```
Application.Exit();
```

Perintah ini digunakan untuk menutup seluruh aplikasi, bukan hanya form saat ini.

Jadi: Semua form yang sedang terbuka akan ditutup Proses aplikasi dihentikan

```

private void btNew_Click(object sender, EventArgs e)
{
    // Membersihkan semua inputan pada form
    txtNama.Text = "";
    cbJenis.Text = "";
    txtHarga.Text = "";
    txtLama.Text = "";
    txtBiaya.Text = "";
    txtJumlah.Text = "";
    txtKembali.Text = "";
    txtAlamat.Text = "";
    txtNomor.Text = "";
    dateTimePicker1.Value = DateTime.Now;
}

```

Penjelasan Fungsi coding :

btNew_Click

Kode ini dijalankan ketika pengguna menekan tombol New / Baru pada aplikasi. Tujuannya adalah mengosongkan semua input di form agar siap digunakan untuk memasukkan data baru.

Penjelasan Baris per Baris

```

private void btNew_Click(object sender, EventArgs e)
{
    // Membersihkan semua inputan pada form
}

```

Ketika tombol btNew di klik, event ini dijalankan.

```

txtNama.Text = "";
cbJenis.Text = "";
txtHarga.Text = "";
txtLama.Text = "";
txtBiaya.Text = "";
txtJumlah.Text = "";
txtKembali.Text = "";
txtAlamat.Text = "";
txtNomor.Text = "";

```

Semua TextBox dan ComboBox di-reset menjadi kosong ("").

Artinya, data yang sebelumnya diinput akan dibersihkan.

```
dateTimePicker1.Value = DateTime.Now;
```

Mengatur kembali DateTimePicker ke tanggal dan waktu saat ini (sekarang).

Jadi tanggal pemesanan kembali fresh.

Hasil dan pembahasan (analisis perilaku, validasi & edge cases)

1. Validasi input

Sangat disarankan menambahkan pemeriksaan angka negatif, batas maksimal (mis. lama > 0), dan menolak teks non-angka. Gunakan int.TryParse, dan tampilkan MessageBox apabila input tidak valid.

2. Pengurutan event

Karena banyak event saling memengaruhi (harga → biaya → kembalian), pastikan event tidak menyebabkan loop tak berujung. Misal, saat menghitung total, jangan menulis ulang txtLama.Text (itu bug yang sebelumnya menyebabkan perilaku aneh).

3. Handling overflow

Jika (harga * lama) bisa melebihi batas int, pertimbangkan long. Namun untuk harga kamar biasanya masih dalam int.

4. UX

Nonaktifkan txtHarga apabila kamu ingin harga hanya dipilih dari daftar (mencegah user edit). Jika admin perlu edit, sediakan mode admin.

5. Error messages

Tambahkan errorProvider atau ToolTip untuk memberi tahu user jika input salah.

6. Internationalization / Format

Untuk tampilan uang, formatkan dengan ToString("N0") atau culture-specific formatting agar tampil lebih rapi (750.000 atau 750,000 sesuai locale)

7. Persistence (opsional)

Jika ingin menyimpan transaksi, tambahkan penyimpanan ke file teks, CSV atau database SQLite.

LAMPIRAN (KODE LENGKAP, SCREENSHOT PROGRAM)

Kode Program Lengkap

1. Form1.cs

```
1     using System;
2     using System.Collections.Generic;
3     using System.ComponentModel;
4     using System.Data;
5     using System.Drawing;
6     using System.Linq;
7     using System.Text;
8     using System.Threading.Tasks;
9     using System.Windows.Forms;
10
11    namespace Projek_Hotel
12    {
13        public partial class Form1 : Form
14        {
15            public Form1()
16            {
17                InitializeComponent();
18            }
19
20            private void textBox3_TextChanged(object sender, EventArgs e)
21            {
22            }
23
24            private void Form1_Load(object sender, EventArgs e)
25            {
26            }
27
28            private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
29            {
30
31            private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
32            {
33                // Menampilkan harga berdasarkan jenis kamar yang dipilih
34                if (cbJenis.SelectedItem.ToString() == "VIP")
35                {
36                    txtHarga.Text = "750000";
37                }
38                else if (cbJenis.SelectedItem.ToString() == "Kelas I")
39                {
40                    txtHarga.Text = "500000";
41                }
42                else if (cbJenis.SelectedItem.ToString() == "Kelas II")
43                {
44                    txtHarga.Text = "300000";
45                }
46                else if (cbJenis.SelectedItem.ToString() == "Kelas III")
47                {
48                    txtHarga.Text = "150000";
49                }
50
51            private void txtLama_TextChanged(object sender, EventArgs e)
52            {
53                // Biaya Total dihitung berdasarkan lama menginap dan harga kamar
54                if (txtLama.Text != "" && txtHarga.Text != "")
55                {
56                    int lama = int.Parse(txtLama.Text);
57                    int harga = int.Parse(txtHarga.Text);
58                    int total = lama * harga;
59                    txtBiaya.Text = total.ToString();
60                }
61            }
62        }
63    }
64}
```

```
1 reference
30  ⑨  ✓  private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
1 reference
51
52
53
54
55
56
57
58
59
60
```

private void cbJenis_SelectedIndexChanged(object sender, EventArgs e)
{
 // Menampilkan harga berdasarkan jenis kamar yang dipilih
 if (cbJenis.SelectedItem.ToString() == "VIP")
 {
 txtHarga.Text = "750000";
 }
 else if (cbJenis.SelectedItem.ToString() == "Kelas I")
 {
 txtHarga.Text = "500000";
 }
 else if (cbJenis.SelectedItem.ToString() == "Kelas II")
 {
 txtHarga.Text = "300000";
 }
 else if (cbJenis.SelectedItem.ToString() == "Kelas III")
 {
 txtHarga.Text = "150000";
 }
}

private void txtLama_TextChanged(object sender, EventArgs e)
{
 // Biaya Total dihitung berdasarkan lama menginap dan harga kamar
 if (txtLama.Text != "" && txtHarga.Text != "")
 {
 int lama = int.Parse(txtLama.Text);
 int harga = int.Parse(txtHarga.Text);
 int total = lama * harga;
 txtBiaya.Text = total.ToString();
 }
}

```

59             txtBiaya.Text = total.ToString();
60         }
61     }
62     else
63     {
64         txtBiaya.Text = "";
65     }
66 }
67
1 reference
68     private void txtJumlah_TextChanged(object sender, EventArgs e)
69     {
70         // Menghitung kembalian berdasarkan jumlah bayar dan biaya total
71         if (txtJumlah.Text != "" & txtLama.Text != "")
72         {
73             int jumlah = int.Parse(txtJumlah.Text);
74             int biaya = int.Parse(txtBiaya.Text);
75             int kembalian = jumlah - biaya;
76             txtKembali.Text = kembalian.ToString();
77         }
78         else
79         {
80             txtKembali.Text = "";
81         }
82     }
83
1 reference
84     private void txtHarga_TextChanged(object sender, EventArgs e)
85     {
86         // Menghitung biaya total ketika harga kamar berubah
87         if (txtLama.Text != "" & txtHarga.Text != "")
88         {
89             int lama = int.Parse(txtLama.Text);
90             int harga = int.Parse(txtHarga.Text);
91             int total = lama * harga;
92             txtLama.Text = total.ToString();
93
94         }
95         else
96         {
97             txtLama.Text = "";
98         }
99
1 reference
100    private void txtBiaya_TextChanged(object sender, EventArgs e)
101    {
102        // Menghitung kembalian ketika biaya total berubah
103        if (txtJumlah.Text != "" & txtLama.Text != "")
104        {
105            int jumlah = int.Parse(txtJumlah.Text);
106            int total = int.Parse(txtLama.Text);
107            int kembalian = jumlah - total;
108            txtKembali.Text = kembalian.ToString();
109        }
110        else
111        {
112            txtKembali.Text = "";
113        }
114    }
115
1 reference
116    private void btClose_Click(object sender, EventArgs e)
117    {
118        // Menutup aplikasi
119        Application.Exit();
120    }
121
1 reference
122    private void btNew_Click(object sender, EventArgs e)

```

```

119 } Application.Exit();
120 }
121 }
122 }
123 }
124 // Membersihkan semua inputan pada form
125 txtNama.Text = "";
126 cbJenis.Text = "";
127 txtHarga.Text = "";
128 txtLama.Text = "";
129 txtBiaya.Text = "";
130 txtJumlah.Text = "";
131 txtKembali.Text = "";
132 txtAlamat.Text = "";
133 txtNomor.Text = "";
134 dateTimePicker1.Value = DateTime.Now;
135 }
136 }
137 }
138 }
139 }
140 }
141 }
142 }
143 }

```

Screenshot Tampilan Aplikasi

