

**LAPORAN PROYEK AKHIR
(TIKET MASUK)**



dibuat oleh:

Nama : Alif Alfian
kelas : X PPLG @
No.Absen : 04

**PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENDIDIKAN KEBUDAYAAN
TAHUN 2025**

DAFTAR ISI

DAFTAR ISI.....	2
BAB I.....	3
PENDAHULUAN.....	3
A. Deskripsi Proyek.....	3
B. Ringkasan Teori C# dan Materi yang Digunakan.....	3
C. Perancangan Program.....	8
BAB II.....	9
IMPLEMENTASI PROGRAM.....	9
A. Potongan Program Kode Utama.....	9
B. Jelaskan Fungsi Masing-Masing Bagian.....	9
C. Hasil dan Pembahasan.....	10
D. Tampilan Output Program.....	10
E. Hasil Uji Coba.....	11
BAB III.....	12
LAMPIRAN PROGRAM.....	12
A. Kode Lengkap.....	12
B. Screenshot Program.....	12

BAB I

PENDAHULUAN

A. Deskripsi Proyek

Aplikasi pemesanan Tiket kolam Renang adalah sebuah program berbaris C# dengan antarmuka GUI (Windows Form/WPF) yang dirancang untuk mempermudah proses pemesanan tiket masuk kolam renang secara cepat dan terkomputerisasi. Sistem ini membuat petugas loket dalam mencatat transaksi, menghitung total pembayaran, mencetak struk, serta mengelola data tarif tiket.

B. Ringkasan Teori C# dan Materi yang Digunakan

1. Variabel dan Tipe Data

Variabel adalah wadah atau tempat untuk menyimpan data dalam pemrograman. Isinya bisa berubah-ubah sesuai kebutuhan program.

Contoh variabel:

nama = "Andi", umur = 17, tinggi = 165.5

Tipe data adalah jenis data yang disimpan di dalam variabel.

Tipe data menentukan bentuk dan operasi apa saja yang bisa dilakukan pada data tersebut

Jenis-jenis Tipe Data (umum)

a. String → teks

Contoh: "Halo", "Belajar Python"

```
string kriteria = comboBox1.SelectedItem?.ToString();  
string harga_tiket = comboBox2.SelectedItem?.ToString();  
string jumlah_tiket = comboBox3.SelectedItem?.ToString();  
string total_harga = $"{Convert.ToInt32(harga_tiket) * Convert.ToInt32(jumlah_tiket)}";
```

b. Integer (int) → bilangan bulat

Contoh: 10, -5

c. Float → bilangan desimal

Contoh: 3.14, 12.5

d. Boolean (bool) → benar atau salah

Contoh: True, False

e. Double → tipe data yang berupa desimal

Contoh: 3,14

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

2. Operator

Operator digunakan untuk melakukan operasi pada variabel dan nilai.

contoh:

a. Operator Aritmatika

Aritmatika adalah cabang matematika yang mempelajari operasi dasar bilangan, seperti penjumlahan, pengurangan, perkalian, dan pembagian. Aritmatika juga mempelajari sifat-sifat bilangan, seperti komutatif, asosiatif, dan distributif. Aritmatika adalah dasar dari semua cabang matematika lainnya dan digunakan dalam berbagai bidang, seperti sains, teknologi, ekonomi, dan lain-lain.

+ : Penjumlahan

- : Pengurangan

* : Perkalian

/ : Pembagian

% : Sisa bagi (modulo)

b. Operator Perbandingan Dalam C#

Operator perbandingan digunakan untuk membandingkan dua nilai dan hasilnya berupa boolean (true atau false).

Jenis-jenis Operator Perbandingan di C#:

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

c. Operator Increment & Decrement

++ : Menambah 1 nilai variabel.

-- : Mengurangi 1 nilai variabel.

d. Operator Logika (Logical Operators) dalam C#

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

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

Hasil operasi dari operator logika sama seperti operator perbandingan, yakni boolean True dan False.

e. Operator Penugasan

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

Operator Penugasan terdiri dari:

Nama Operator	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=

3. Percabangan Kondisi

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

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

a. Pernyataan if

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

```

if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
{
    MessageBox.Show("Uang kurang");
}

```

b. Pernyataan if-else

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

```

if (!string.IsNullOrEmpty(txtBayar.Text))
{
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
    {
        MessageBox.Show("Uang kurang");
    }
    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
    lblkembalian.Text = $"{kembalian}";
}
else MessageBox.Show("Error");

```

c. Pernyataan if-else if-else

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

d. Pernyataan switch-case

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

4. Array C#

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

Array digambarkan dalam sebuah tabel terdiri dari : value/nilai dan index/indeks
 Value / nilai: isi dari array Index : urutan array (dimulai dari 0)

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

5. Perulangan

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

a. Perulangan for

Digunakan ketika jumlah perulangan sudah diketahui di awal.

b. Perulangan while

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

c. Perulangan do-while

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

d. Perulangan foreach

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

e. Nested Loop (perulangan bersarang)

Nested Loop (atau perulangan bersarang) adalah perulangan di dalam perulangan lain.

6. Pengertian Fungsi

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

Jenis Fungsi	Ciri-ciri	Contoh
Tanpa parameter dan tanpa return	Hanya menampilkan pesan atau melakukan tugas tanpa hasil balik.	void Sapa()
Dengan parameter	Menerima data input dari luar.	void SapaNama(string nama)
Dengan return	Menghasilkan nilai dari	int Tambah(int a, int b).

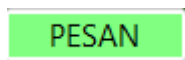
	perhitungan.	
Dengan parameter & return	Lengkap, menerima input dan menghasilkan output	double HitungLuas(double p, double l).

C. Perancangan Program

1. Latar Belakang



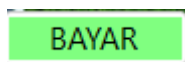
2. Desain Tombol



Tombol Pesan



Tombol Batal



Tombol Bayar



Kemunculan Kembalian

BAB II

IMPLEMENTASI PROGRAM

A. Potongan Program Kode Utama

1. Menampilkan Kriteria Tiket

```
private void button1_Click(object sender, EventArgs e)
{
    string kriteria = comboBox1.SelectedItem?.ToString();
    string harga_tiket = comboBox2.SelectedItem?.ToString();
    string jumlah_tiket = comboBox3.SelectedItem?.ToString();
    string total_harga = $"{Convert.ToInt32(harga_tiket) * Convert.ToInt32(jumlah_tiket)}";
    dataGridView2.Rows.Add(kriteria, harga_tiket, jumlah_tiket, total_harga);

    txtTotal.Text = total_harga.ToString();
}
```

2. Menghitung Total Biaya dan Kembalian

```
private void button3_Click_3(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtBayar.Text))
    {
        if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
        {
            MessageBox.Show("Uang kurang");
        }
        double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
        lblkembalian.Text = $"{kembalian}";
    }
    else MessageBox.Show("Error");
    return;
}
```

3. Pengisian Pilihan Tiket

```
public Form1()
{
    InitializeComponent();
    dataGridView1.Rows.Add("Dewasa", "10.000");
    dataGridView1.Rows.Add("Anak-anak", "20.000");
    dataGridView1.Rows.Add("Remaja", "25.000");
}
```

B. Jelaskan Fungsi Masing-Masing Bagian

1. Fungsi Halaman Pengelolaan Tiket

Halaman Tiket Pengelolaan biasanya digunakan untuk mencatat, melacak, dan menyelesaikan berbagai *issue* atau permintaan yang terjadi dalam operasional kasir dan toko. Fitur ini umum dijumpai pada sistem kasir modern yang memiliki modul manajemen pelayanan atau operasional.

2. Fungsi Pemilihan Tiket Pesanan

Fitur Pemilihan Tiket Pesanan digunakan untuk memilih dan mengelola tiket/nota/order yang sedang berjalan (ongoing order). Berguna saat terdapat banyak transaksi atau pesanan yang belum selesai.

3. Fungsi Menghitung Total Harga

Fitur Menghitung Total Harga adalah salah satu fungsi utama dalam aplikasi kasir untuk menentukan jumlah yang harus dibayar pelanggan berdasarkan item yang dibeli.

4. Fungsi Menghitung Kembalian

Fungsi menghitung kembalian digunakan untuk menentukan berapa uang yang harus dikembalikan kepada pelanggan setelah mereka membayar.

C. Hasil dan Pembahasan

Aplikasi “Kasir Tiket Kolam Renang” berhasil berjalan dengan baik. Fitur yang dapat digunakan:

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

D. Tampilan Output Program

The screenshot displays the 'KOLAM BERENANG' application interface. At the top, the title 'KOLAM BERENANG' is prominently displayed. Below the title, there are three dropdown menus for selecting 'KRITERIA', 'HARGA TIKET KOLAM', and 'JUMLAH TIKET'. To the right of these inputs is a table with two columns: 'Kriteria' and 'Harga Tiket'. Below the dropdowns are two buttons: a green 'PESAN' button and a red 'BATAL' button. At the bottom, there is a larger table with four columns: 'Kriteria', 'Harga Tiket', 'Jumlah Tiket', and 'Total Harga'. Below this table, the text 'TOTAL HARGA :' is visible. The background of the interface features a scenic view of a swimming pool with palm trees and a blue sky.

E. Hasil Uji Coba

Uji coba dilakukan pada fitur utama aplikasi kasir untuk penjualan tiket kolam renang. Berdasarkan hasil pengujian, aplikasi bekerja dengan baik dan memenuhi kebutuhan operasional berikut:

1. Pemilihan Jenis Tiket
 - a. Kasir dapat memilih tiket dewasa, anak-anak, paket keluarga, dan promo dengan mudah.
 - b. Harga tiket otomatis muncul sesuai pilihan, meminimalkan kesalahan input.
2. Perhitungan Total Harga
 - a. Sistem menghitung total harga secara otomatis berdasarkan jumlah tiket dan harga per unit.
 - b. Jika ada biaya tambahan, seperti sewa pelampung atau gazebo, total akhir langsung diperbarui.
3. Perhitungan Kembalian
 - a. Saat kasir memasukkan jumlah uang yang dibayarkan pelanggan, sistem langsung menampilkan kembalian yang harus diberikan.
 - b. Sistem memberi peringatan jika uang yang diterima kurang dari total harga.

BAB III

LAMPIRAN PROGRAM

A. Kode Lengkap

```

string nama = textBox1.Text.ToString();
string notelpn = textBox2.Text.ToString();
string tanggal = comboBox6.Text.ToString();
string jam = comboBox3.Text.ToString();
string tujuan = comboBox4.Text.ToString();
string jumlah = comboBox5.Text.ToString();

if (!string.IsNullOrEmpty(txtBayar.Text))
{
    if (Convert.ToDouble(txtBayar.Text) < Convert.ToDouble(txtTotal.Text))
    {
        MessageBox.Show("Uang kurang");
    }
    double kembalian = Convert.ToDouble(txtBayar.Text) - Convert.ToDouble(txtTotal.Text);
    lblkembalian.Text = $"{kembalian}";
}
else MessageBox.Show("Error");

public Form1()
{
    InitializeComponent();
    dataGridView1.Rows.Add("Dewasa", "10.000");
    dataGridView1.Rows.Add("Anak-anak", "20.000");
    dataGridView1.Rows.Add("Remaja", "25.000");
}

private void button1_Click(object sender, EventArgs e)
{
    string kriteria = comboBox1.SelectedItem?.ToString();
    string harga_tiket = comboBox2.SelectedItem?.ToString();
    string jumlah_tiket = comboBox3.SelectedItem?.ToString();
    string total_harga = $"{Convert.ToInt32(harga_tiket) * Convert.ToInt32(jumlah_tiket)}";
    dataGridView2.Rows.Add(kriteria, harga_tiket, jumlah_tiket, total_harga);

    txtTotal.Text = total_harga.ToString();
}

```

B. Screenshot Program

