

**LAPORAN PROYEK AKHIR SEMESTER GANSAL PEMBUATAN
APLIKASI MENGGUNAKAN WINDOWS FORM APP C#**



NAMA : Risma Apriliana

KELAS : X PPLG 2

NIS : 258771

JUDUL PROYEK : TIKET PEMESANAN BUS

PENGEMBANGAN PERANGKAT LUNAK DAN GIM

SMK NEGERI 1 KANDEMAN

TAHUN PELAJARAN 2025 / 2026

DESKRIPSI PROYEK

Proyek ini bertujuan untuk membuat **Aplikasi Tiket Pemesanan Bus** berbasis desktop menggunakan **C# Windows Forms** yang dirancang untuk membantu pengguna dalam melakukan pemesanan tiket secara cepat dan efisien. Aplikasi ini memungkinkan pengguna memasukkan data pemesanan, memilih tujuan, kelas bus, jumlah tiket, serta menampilkan detail lengkap pesanan beserta total pembayaran.

Fitur Utama Aplikasi :

1. Input Data Pemesan

- Nama Pemesan
- Nomor Telepon
- Tanggal Keberangkatan (Tanggal, Bulan, Tahun)
- Jam Keberangkatan

2. Pengaturan Tiket

- Tujuan Perjalanan
- Kelas Bus (Ekonomi, Bisnis, VIP, VVIP)
- Jumlah Tiket

3. Detail Pemesanan

Sistem akan menampilkan ringkasan data pemesanan: Nama, Telepon, Tanggal, Jam, Tujuan, Kelas, Jumlah Tiket.

4. Perhitungan Harga Tiket Otomatis

Harga ditentukan berdasarkan kelas bus :

- Ekonomi – Rp 150.000
- Bisnis – Rp 250.000
- VIP – Rp 400.000
- VVIP – Rp 800.000

Total pembayaran dihitung otomatis:

$$\text{Total} = \text{Harga} * \text{Jumlah Tiket}$$

5. Fitur Tombol

- Pesan : Menyimpan dan menampilkan detail pesanan
- Clear : Untuk menghapus semua data
- Exit : Untuk keluar dari aplikasi

RINGKASAN TEORI C# YANG DIGUNAKAN

1. TextBox, ComboBox, Label

Digunakan untuk menerima input dari pengguna dan menampilkan output hasil perhitungan.

2. Event Hadling (Button Clik)

Digunakan untuk menjalankan fungsi tertentu saat tombol ditekan, Misalnya tombol PESAN, CLEAR, dan EXIT.

3. Conditional Statements (IF – ELSE)

Digunakan untuk menentukan harga berdasarkan kelas bus :

```
if (cmbKelas.Text == "Ekonomi") harga = 150000;
else if (cmbKelas.Text == "Bisnis") harga = 250000;
else if (cmbKelas.Text == "VIP") harga = 400000;
else if (cmbKelas.Text == "VVIP") harga = 800000;
```

Penjelasan:

Program tersebut berfungsi untuk menentukan harga berdasarkan pilihan kelas di ComboBox. Jadi ketika pengguna memilih jenis kelas seperti Ekonomi, Bisnis, VIP, atau VVIP, program akan mencocokkan pilihan itu dan menetapkan harga yang sesuai. Masing-masing kelas sudah punya harga tetap, jadi program hanya memilih harga yang cocok dengan kelas yang dipilih pengguna.

4. Parsing Data

Mengubah input string menjadi bilangan integer :

```
int jumlah = int.Parse(txtJumlah.Text);
int harga = 0;
```

Penjelasan :

Bagian itu dipakai untuk mengambil teks dari input pengguna lalu mengubahnya menjadi angka, karena program hanya bisa menghitung jika datanya berupa angka. Setelah itu program menyiapkan variabel harga dengan nilai awal nol sebagai tempat menyimpan harga yang akan ditentukan pada langkah berikutnya.

5. Message Box

Digunakan untuk menampilkan pesan informasi, konfirmasi, dan peringatan :

```
// menampilkan pesan tiket
MessageBox.Show("Tiket Berhasil Dipesan!", "Informasi",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
```

Penjelasan :

Adalah menyampaikan berbagai jenis pesan kepada pengguna. Pesannya bisa berupa informasi yang memberi tahu sesuatu, pesan yang meminta pengguna memastikan atau menyetujui suatu pilihan, dan pesan yang mengingatkan pengguna tentang hal tertentu.

6. String Interpolation

Digunakan untuk mempermudah menampilkan teks harga dan total :

```
int total = jumlah * harga;  
int Jumlah = total;  
hargatiket.Text = "Rp " + harga.ToString();  
Total.Text = "Rp " + Jumlah.ToString();
```

Penjelasan :

Bagaimana itu menghitung total uang yang harus dibayar dengan cara mengalikan jumlah tiket dengan harganya. Setelah totalnya ada, program menampilkan harga dan total tersebut dalam bentuk tulisan supaya pengguna bisa melihat berapa harga satu tiket dan berapa total keseluruhannya.

KONSEP PEMROGRAMAN YANG DIIMPLEMENTASIKAN

Berikut adalah konsep-konsep pemrograman yang digunakan dalam pembuatan aplikasi Pemesanan Tiket Bus :

1. Event Hadling (Penanganan peristiwa)

Digunakan untuk menangani aksi pengguna seperti klik tombol. Contoh :

```
// menampilkan pesan tiket  
MessageBox.Show("Tiket Berhasil Dipesan!", "Informasi",  
    MessageBoxButtons.OK, MessageBoxIcon.Information);
```

Penjelasan :

Pesan itu muncul karena dijalankan Ketika pengguna melakukan sebuah Tindakan, seperti menekan tombol untuk memesan tiket. Saat Tindakan itu terjadi, program langsung menampilkan pesannya. Jadi munculnya pesan tersebut merupakan respon dari aktivitas yang dilakukan pengguna pada saat itu.

2. Conditional Statement (IF – ELSE)

Digunakan untuk menentukan harga tiket berdasarkan kelas bus.

```
if (cmbKelas.Text == "Ekonomi") harga = 150000;  
else if (cmbKelas.Text == "Bisnis") harga = 250000;  
else if (cmbKelas.Text == "VIP") harga = 400000;  
else if (cmbKelas.Text == "VVIP") harga = 800000;
```

Penjelasan :

Bagaimana tersebut dipakai untuk mengecek kelas bus yang dipilih pengguna. Program melihat apakah yang dipilih Ekonomi, Bisnis, VIP, atau VVIP. Setelah menemukan kecocokan, program langsung memberi harga yang sesuai dengan kelas tersebut. Jadi harga tiket ditentukan berdasarkan pilihan kelas yang dipilih pengguna.

3. Parsing Data (Konversi Tipe Data)

Digunakan untuk mengubah input jumlah tiket dari string ke integer.

```
int jumlah = int.Parse(txtJumlah.Text);
int harga = 0;
```

Penjelasan :

Pada bagian itu, program mengambil nilai jumlah tiket yang awalnya berupa tulisan dari input pengguna. Karena tulisan tidak bisa dihitung, program mengubahnya menjadi bentuk angka agar bisa dipakai untuk perhitungan selanjutnya. Jadi proses tersebut membuat data yang awalnya berbentuk teks bisa dipakai sebagai angka dalam program.

4. String Interpolation & Concatenation

Digunakan untuk menampilkan hasil perhitungan harga.

```
int total = jumlah * harga;
int Jumlah = total;
hargatiket.Text = "Rp " + harga.ToString();
Total.Text = "Rp " + Jumlah.ToString();
```

Penjelasan :

Hal itu dijelaskan karena pada bagian tersebut nilai angka seperti harga dan total diubah menjadi teks, kemudian digabungkan dengan tulisan lain seperti “Rp”. Proses penggabungan antara teks dan nilai yang telah dikonversi membuat hasil perhitungan dapat ditampilkan dalam bentuk tulisan yang lebih jelas dan mudah dipahami oleh pengguna.

5. Manipulasi Kontrol Form

Digunakan untuk memindahkan data input ke bagian detail.

```
// menampilkan detail tiket
groupBox2_Enter(sender, e);
nama.Text = txtNama.Text;
telepon.Text = txtNomor.Text;
tujuan.Text = cmbTujuan.Text;
kelas.Text = cmbKelas.Text;
tanggal.Text = txtTanggal.Text;
jam.Text = cmbJam.Text;
```

Penjelasan :

Pada bagian tersebut, program mengambil data yang sudah diisi pengguna, seperti nama, nomor telepon, tujuan, kelas, tanggal, dan jam. Setelah itu, data tersebut dipindahkan ke bagian detail agar pengguna dapat melihat kembali informasi yang

sudah mereka masukkan. Dengan cara ini, bagian detail menampilkan ringkasan dari seluruh input yang sebelumnya diisi pada form.

6. Clear / Reset Form

Menghapus seluruh input dengan mengosongkan TextBox dan ComboBox.

```
// Textbox  
txtNama.Text = "";  
txtNomor.Text = "";  
txtJumlah.Text = "";  
txtTanggal.Text = "";  
  
// combobox  
cmbBulan.Text = "";  
cmbTahun.Text = "";  
cmbJam.Text = "";  
cmbTujuan.Text = "";  
cmbKelas.Text = "";  
  
// Detail  
nama.Text = "";  
telepon.Text = "";  
tanggal.Text = "";  
jam.Text = "";  
lblJml.Text = "";  
hargatiket.Text = "";  
Total.Text = "";
```

Penjelasan :

Kode ini berfungsi untuk mengembalikan seluruh elemen input pada form ke kondisi awal. Setiap TextBox dan ComboBox diisi ulang dengan nilai kosong ("") sehingga tidak ada data lama yang tersisa. Selain itu, bagian detail tiket yang menampilkan ringkasan data juga dikosongkan agar tidak menampilkan informasi dari pengisian sebelumnya. Proses ini memastikan bahwa form siap digunakan kembali untuk memasukkan data baru tanpa adanya nilai yang tertinggal dari input sebelumnya.

7. Dialog Konfirmasi (MessageBox)

Digunakan pada tombol Exit.

```
{  
    DialogResult tanya;  
  
    tanya = MessageBox.Show("Yakin ingin keluar?",  
                           "Exit",  
                           MessageBoxButtons.YesNo,  
                           MessageBoxIcon.Question);
```

Penjelasan :

Kode itu membuat pertanyaan muncul sebelum aplikasi ditutup. Pertanyaan itu meminta kepastian dari pengguna, apakah benar-benar ingin keluar atau tidak. Jawaban pengguna disimpan, sehingga program tahu harus lanjut menutup aplikasi atau tetap berjalan.

PERANCANGAN PROGRAM

Alur Kerja Aplikasi :

- Pengguna Mengisi Data Pemesan.
- Pengguna Memilih Tujuan, Kelas, Dan Jumlah Tiket.

- Sistem Menghitung Total Harga Berdasarkan Input.
- Sistem Menampilkan Detail Pemesanan.
- Pengguna Dapat Menghapus Data Atau Keluar Dari Aplikasi.

Struktur Form

- GroupBox 1 = Form Input Pemesanan
- GroupBox 2 = Tampilan Detail Pemesanan
- Button = Pesan, Clear, Exit
- PictureBox = Menampilkan Gambar Bus

IMPLEMENTASI PROGRAM

1. Tombol PESAN

Kode untuk menampilkan detail dan menghitung total harga :

```
int jumlah = int.Parse(txtJumlah.Text);
int harga = 0;
if (cmbKelas.Text == "Ekonomi") harga = 150000;
else if (cmbKelas.Text == "Bisnis") harga = 250000;
else if (cmbKelas.Text == "VIP") harga = 400000;
else if (cmbKelas.Text == "VVIP") harga = 800000;
int total = jumlah * harga;
int Jumlah = total;
hargatiket.Text = "Rp " + harga.ToString();
Total.Text = "Rp " + Jumlah.ToString();
```

Penjelasan :

Kode ini pertama mengambil jumlah tiket yang dimasukkan pengguna. Setelah itu, program mengecek kelas bus yang dipilih untuk menentukan harganya. Kalau kelasnya berbeda, harganya juga berbeda. Setelah tahu harga per tiket, program menghitung total biaya dengan mengalikan jumlah tiket dan harga tersebut. Lalu hasilnya ditampilkan pada bagian harga dan total, supaya pengguna bisa melihat berapa yang harus dibayar.

2. Tombol CLEAR

```
| {  
    // Textbox  
    txtNama.Text = "";  
    txtNomor.Text = "";  
    txtJumlah.Text = "";  
    txtTanggal.Text = "";  
  
    // combobox  
    cmbBulan.Text = "";  
    cmbTahun.Text = "";  
    cmbJam.Text = "";  
    cmbTujuan.Text = "";  
    cmbKelas.Text = "";  
  
    // Detail  
    nama.Text = "";  
    telepon.Text = "";  
    tanggal.Text = "";  
    jam.Text = "";  
    lblJml.Text = "";  
    hargatiket.Text = "";  
    Total.Text = "";  
}
```

Penjelasan :

Kode tersebut berfungsi membersihkan semua isi form. Setiap kotak teks dan setiap pilihan yang sebelumnya sudah terisi akan dikosongkan kembali. Bagian detail yang menampilkan hasil juga ikut dihapus. Hasilnya, form kembali ke keadaan awal, sehingga bisa dipakai lagi untuk mengisi data baru.

3. Tombol EXIT

```
{  
    DialogResult tanya;  
  
    tanya = MessageBox.Show("Yakin ingin keluar?",  
                           "Exit",  
                           MessageBoxButtons.YesNo,  
                           MessageBoxIcon.Question);  
  
    if (tanya == DialogResult.Yes)  
    {  
        Application.Exit();  
    }  
}
```

Penjelasan :

Kode itu menampilkan pertanyaan dulu sebelum aplikasi ditutup. Pertanyaan ini muncul supaya program tahu apakah pengguna benar-benar ingin keluar atau tidak. Jika pengguna memilih “Yes”, aplikasi akan langsung ditutup. Tapi kalau memilih “No”, aplikasi tetap berjalan seperti biasa.

LAMPIRAN

1. Tampilan Aplikasi

TIKET PEMESANAN BUSS

PEMESANAN TIKET

Nama Pemesanan	<input type="text"/>	Tujuan	<input type="text"/>
Nomor Telepon	<input type="text"/>	Kelas	<input type="text"/>
Tgl. Pemberangkatan	<input type="text"/> / <input type="text"/> / <input type="text"/>	Jumlah Tiket	<input type="text"/>
Jam Kebrangakatan	<input type="text"/>		

DETAIL

Nama :-	
Telepon :-	
Tanggal :- / - / -	
Jam :-	
Tujuan :-	
Kelas :-	
Jumlah :-	
Harga / Tiket : Rp	
Total Bayar : Rp	



PESAN CLEAR EXIT

2. Kode Program Lengkap

```
    ...
    // menampilkan pesan tiket
    MessageBox.Show("Tiket Berhasil Dipesan!", "Informasi",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
    // menampilkan detail tiket
    groupBox2_Enter(sender, e);
    nama.Text = txtNama.Text;
    telepon.Text = txtNomor.Text;
    tujuan.Text = cmbTujuan.Text;
    kelas.Text = cmbKelas.Text;
    tanggal.Text = txtTanggal.Text;
    jam.Text = cmbJam.Text;
    /lblJumlah.Text = txtJumlah.Text;
    bulan.Text = cmbBulan.Text;
    tahun.Text = cmbTahun.Text;
    jml.Text = txtJumlah.Text;
    int jumlah = int.Parse(txtJumlah.Text);
    int harga = 0;
    if (cmbKelas.Text == "Ekonomi") harga = 150000;
    else if (cmbKelas.Text == "Bisnis") harga = 250000;
    else if (cmbKelas.Text == "VIP") harga = 400000;
    else if (cmbKelas.Text == "VVIP") harga = 800000;
    int total = jumlah * harga;
    int Jumlah = total;
    hargatiket.Text = "Rp " + harga.ToString();
    Total.Text = "Rp " + Jumlah.ToString();
}

{
    ...
    // Textbox
    txtNama.Text = "";
    txtNomor.Text = "";
    txtJumlah.Text = "";
    txtTanggal.Text = "";

    // combobox
    cmbBulan.Text = "";
    cmbTahun.Text = "";
    cmbJam.Text = "";
    cmbTujuan.Text = "";
    cmbKelas.Text = "";

    // Detail
    nama.Text = "";
    telepon.Text = "";
    tanggal.Text = "";
    jam.Text = "";
    lblJml.Text = "";
    hargatiket.Text = "";
    Total.Text = "";
}
{
    DialogResult tanya;

    tanya = MessageBox.Show("Yakin ingin keluar?",
                           "Exit",
                           MessageBoxButtons.YesNo,
                           MessageBoxIcon.Question);
    if (tanya == DialogResult.Yes)
    {
        Application.Exit();
    }
}

{
    ...
    int jumlah = int.Parse(txtJumlah.Text);
    int harga = 0;

    if (cmbKelas.Text == "Ekonomi") harga = 150000;
    else if (cmbKelas.Text == "Bisnis") harga = 250000;
    else if (cmbKelas.Text == "VIP") harga = 400000;
    else if (cmbKelas.Text == "VVIP") harga = 800000;

    int total = jumlah * harga;
    int Jumlah = total;

    // === DETAIL ===
    tanggal.Text = tanggal.Text;
    jam.Text = jam.Text;
    /lblJumlah.Text = jumlah.ToString();
    hargatiket.Text = "Rp " + harga.ToString();
    Total.Text = "Rp " + total.ToString();
}
```