

LAPORAN PROYEK AKHIR SEMESTER GASAL
JURUSAN PENGEMBANGAN PERANGKAT LUNAK DAN GIM
SMK NEGERI 1 KANDEMAN
TAHUN PELAJARAN 2025/2026



Disusun oleh:

Nama : Alfa Rizqi

Kelas : X PPLG 2

NIS : 258745

PEMERINTAH PROVINSI JAWA TENGAH
DINAS PENDIDIKAN DAN KEBUDAYAAN
TAHUN 2025

DAFTAR ISI

DAFTAR ISI.....	1
BAB I.....	2
PENDAHULUAN.....	2
A. Judul Proyek.....	2
B. Deskripsi Proyek.....	2
C. Ringkasan Teori C# dan Materi yang Digunakan.....	3
D. Perancangan Program.....	21
BAB II.....	24
IMPLEMENTASI PROGRAM.....	24
A. Potongan Kode Program Utama.....	24
B. Fungsi Masing-masing Bagian.....	30
C. Hasil dan Pembahasan.....	31
D. Tampilan Output Program.....	31
E. Hasil Uji Coba.....	34
BAB III.....	37
LAMPIRAN PROGRAM.....	37
A. Kode Lengkap.....	37
B. Screenshot Program.....	60

BAB I

PENDAHULUAN

A. Judul Proyek

“Aplikasi Pengelolaan Tiket Pesawat: Kapal Atas Indonesia”

B. Deskripsi Proyek

Aplikasi Pengelolaan Tiket Pesawat “Kapal Atas Indonesia” merupakan sebuah aplikasi GUI (Graphical User Interface) berbasis Windows Form yang dikembangkan menggunakan bahasa pemrograman C#. Aplikasi ini dirancang untuk mendukung proses operasional pembuatan, pengelolaan, dan verifikasi tiket pesawat secara lebih cepat, efisien, dan mudah digunakan.



Gambar 1.1. Logo Aplikasi “Kapal Atas Indonesia”.

Melalui aplikasi ini, pengguna dapat membuat tiket, melihat daftar tiket aktif maupun non aktif, serta melakukan verifikasi tiket untuk memastikan keaslian dan status penggunaannya. Akumulasi tiket yang terbuat juga akan terhitung, sehingga mempermudah pengguna untuk mengetahui jumlah total biaya tiket, dan kembalian pelanggan. Sistem juga menyediakan fitur pencetakan tiket fisik yang meningkatkan dalam hal fleksibilitas, aksesibilitas, dan keamanan tambahan. Selain itu, terdapat fitur otomatisasi pengoreksi kesalahan, sehingga dengan fitur ini, risiko kesalahan input dapat diminimalisir dan keakuratan data tiket dapat terjaga.

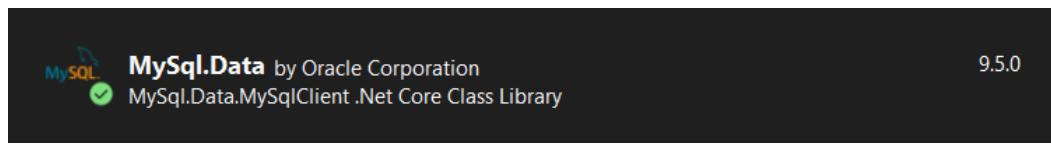
Seluruh data yang diolah oleh aplikasi disimpan secara permanen dalam *database MySQL localhost*, sehingga informasi tiket tersusun rapi, aman, dan mudah diakses kapanpun dan dimanapun apabila diperlukan. Selain itu, tiket yang telah dibuat juga dapat dihapus maupun dinonaktifkan sesuai kebutuhan operasional tanpa mengganggu keseluruhan sistem.

Dengan antarmuka berbasis Windows Form yang sederhana dan mudah dipahami, aplikasi ini memberikan pengalaman penggunaan yang praktis serta mendukung peningkatan efisiensi pelayanan pada perusahaan maskapai.

C. Ringkasan Teori C# dan Materi yang Digunakan

1. MySQL Data

MySQL dalam C# adalah cara aplikasi C# berinteraksi dengan basis data (*database*) MySQL untuk menyimpan, mengolah, dan mengambil data. Pada Visual Studio Code 2022 menggunakan NuGet Package Manager untuk menambahkan MySQL data. Pada proyek ini, MySQL data digunakan untuk menyimpan data tiket secara permanen, memuat dan mengolahnya.



```
using MySql.Data.MySqlClient; // tambahan NuGet Packages, untuk include MySQL
```

2. Variabel dan Tipe Data

Variabel adalah sebuah wadah penyimpanan data pada program yang akan digunakan selama program itu berjalan. Tipe data adalah jenis-jenis data yang akan disimpan di dalam variabel. Seperti data teks, angka, huruf dan lainnya.

Nama Tipe Data	Arti	Contoh
string	Tipe data teks	string pesan = “Saya adalah Alfa”;
int	Tipe data angka	int nilai = 100;
double	Tipe data desimal	double nilai = 100.23;
float	Tipe data pecahan	float setengah = 1.2f; (huruf f → float)
bool	Tipe data boolean (<i>true or</i>	bool isValid = true;

	<i>false</i>)	
--	----------------	--

Pada proyek ini, tipe data yang digunakan antara lain:

a. String

String adalah tipe data yang berupa teks. Tipe data ini hampir selalu digunakan di semua formulir pada proyek ini, mulai dari formulir pembuatan tiket baru, daftar tiket aktif dan tiket mati, serta verifikasi tiket. Pada proyek ini, tipe data ini digunakan untuk menyimpan nama, tanggal lahir, *e-mail*, nama negara, waktu, kelas layanan, layanan tambahan, nama hotel, penerbangan pilihan, dan informasi tambahan.

```
string nama = textBox_nama.Text; // variable nama diset ke textBox_nama.Text
string ttl = dateTime_ttl.Text; // variable ttl diset ke dateTime_ttl.Text
string email = textBox_email.Text; // variable email diset ke textBox_email.Text
string nomor_negara = combo_nomor_negara.SelectedItem?.ToString(); // mengeset variable nomor_negara ke
// combo_nomor_negara.SelectedItem yang diubah dulu menjadi string.
string negara_keberangkatan = textBox_keberangkatan.Text; // mengeset var negara_keberangkatan ke
textBox_keberangkatan.Text
string negara_tujuan = textBox_tujuan.Text; // mengeset var negara_tujuan ke textBox_tujuan.Text
string tanggal_berangkat = $"{dateTime_keberangkatan.Text}, {maskedTextBox_waktu_berangkat.Text}"; // var
// tanggal_berangkat di set menjadi gabungan dateTime_keberangkatan.Text dengan
maskedTextBox_waktu_berangkat.Text
string input_berangkat = maskedTextBox_waktu_berangkat.Text; // var input_berangkat diset ke
maskedTextBox_waktu_berangkat.Text
string kelas_layanan = ""; // membuat var kelas_layanan dengan tidak ada isinya
string layanan = ""; // var layanan dengan isi kosong
string hotel_pilihan = textBox_nama_hotel_pilihan.Text; // set var hotel_pilihan ke
textBox_nama_hotel_pilihan.Text
string penerbangan_pilihan = comboBox_penerbangan.Text; // set var penerbangan_pilihan ke
comboBox_penerbangan.Text
string informasi_tambahan = textBox_informasi_tambahan.Text; //set var informasi_tambahan ke
textBox_informasi_tambahan.Text
```

b. Int

Int adalah tipe data yang berupa angka. Pada proyek ini, tipe data *int* digunakan untuk menyimpan nomor telepon pelanggan, harga tiket, serta total harga tiket.

```
int harga = 0; // mendefinisikan awal harga menjadi 0
int total_harga = Convert.ToInt32(label_total_harga.Text); // mengisi var total_harga menjadi nilai dari
// label total harga yang diconvert ke int
```

c. Double

Double adalah tipe data yang berupa desimal. Pada proyek ini, tipe data *double* digunakan untuk menyimpan jumlah uang pelanggan dan biaya admin. *Double* disini digunakan untuk menyimpan data angka yang lebih besar.

```
double total = uang_pelanggan - Convert.ToInt32(label_total_harga.Text); // set var total (total
// kembalian) dengan uang pelanggan - label_total_harga.Text yang sudah diubah ke int
```

d. Bool

Pada proyek ini, tipe data *bool* digunakan menetapkan apakah formulir valid atau tidak.

```
bool invalid = false; // variable bool, untuk menjadi saklar, apakah invalid atau tidak
```

3. Operator

Dalam pemrograman, operator digunakan untuk melakukan operasi pada variabel dan nilai. Pada proyek ini, operator yang digunakan antara lain:

a. Operator Aritmatika

Operator aritmatika mencakup penjumlahan, pengurangan, perkalian, pembagian, sisa bagi (modulo).

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa bagi (modulo)	%

Pada proyek ini, operator aritmatika yang digunakan hanya penjumlahan dan pengurangan. Penjumlahan digunakan untuk penambahan teks pada label informasi tambahan. Sedangkan pengurangan digunakan untuk mengurangi nilai dari variabel uang pelanggan untuk harga total

```
● ● ●  
lbl_informasi.Text += "Rp. " +  
FormatMoney(Convert.ToInt32(dataGrid_ticket.CurrentRow.Cells["Total_Harga"].Value)); // ambil  
data dari dataGrid_ticket.CurrentRow.Cells, lalu di convert ke int dan diformat ke money.  
  
double total = uang_pelanggan - Convert.ToInt32(label_total_harga.Text); // set var total  
(total kembalian) dengan uang pelanggan - label_total_harga.Text yang sudah diubah ke int
```

b. Operator Perbandingan

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

Nama Operator	Simbol	Contoh

Sama dengan	$=$	$5 == 5 \rightarrow \text{true}$
Tidak sama dengan	\neq	$5 \neq 3 \rightarrow \text{true}$
Lebih besar dari	$>$	$10 > 7 \rightarrow \text{true}$
Lebih kecil dari	$<$	$3 < 8 \rightarrow \text{true}$
Lebih besar atau sama dengan	\geq	$6 \geq 6 \rightarrow \text{true}$
Lebih kecil atau sama dengan	\leq	$4 \leq 9 \rightarrow \text{true}$

Pada proyek ini, operator perbandingan yang digunakan antara lain:

1) Sama dengan

Pada proyek ini, operator perbandingan sama dengan digunakan untuk memeriksa nilai suatu variabel, memeriksa apakah pengguna memilih pilihan pada suatu *combobox* atau tidak, dan memeriksa baris yang dipilih pada *data grid view*. *Combobox* yang diperiksa seperti *combobox* nomor negara, dan kelas layanan. Variabel yang diperiksa adalah variabel *invalid*. Pada *data grid view* digunakan untuk memeriksa baris saat ini (*current row*) pada *data grid view* daftar tiket yang dibuat.

```
● ● ●
if (combo_nomor_negara.SelectedIndex == -1) // jika pengguna tidak memilih combo_nomor_negara
{
    MessageBox.Show("Nomor negara tidak dipilih", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    combo_nomor_negara.Focus(); // fokus ke combo_nomor_negara
    return; // berhentikan program
}

if (invalid == false) // jika nilai invalid tidak true
{
    dataGrid_ticket.Rows.Add(nama, ttl, email, nomor_negara, negara_keberangkatan, negara_tujuan,
    tanggal_berangkat, kelas_layanan, layanan, hotel_pilihan, penerbangan_pilihan,
    informasi_tambahan, harga); // tambahkan var tadi ke dataGrid_ticket.Rows dengan urutan yang pas

    total_harga += harga; // pengisian dan penambahan nilai var total_harga dengan var harga
    label_total_harga.Text = $"{total_harga}"; // mengubah label_total_harga.Text sesuai dengan
    harga
}

if (dataGrid_ticket.CurrentRow == null) // jika dataGrid_ticket tidak ada baris yang dipilih
{
    MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); // messagebox error ditampilkan
    return; // program diberhentikan
}
```

2) Tidak sama dengan

Pada proyek ini, operator perbandingan tidak sama dengan digunakan untuk memeriksa baris saat ini (*current row*) yang dipilih di *data grid view* daftar tiket aktif dan daftar tiket nonaktif.

```
● ● ●  
if (dataGrid_active_ticket.CurrentRow != null) // jika dataGrid_active_ticket.CurrentRow  
dipilih  
{  
    PrintPreviewDialog preview = new PrintPreviewDialog(); // membuat var preview  
    preview.Document = printDocument1; // set ke preview dari printDocument1  
    preview.ShowDialog(); // menunjukkan preview  
}  
else // jika tidak ada row/baris yang dipilih.  
{  
    MessageBox.Show("Pilih baris tiket yang ingin anda cetak", "Error", MessageBoxButtons.OK,  
    MessageBoxIcon.Error); // messagebox error  
}
```

3) Lebih kecil

Pada proyek ini, operator perbandingan lebih kecil digunakan untuk memeriksa suatu nilai *variable*, *length* dari suatu *array*, dan *value* dari suatu *progress bar*. Variabel yang diperiksa adalah variabel total kembalian. *Length array* yang diperiksa adalah *array* nama tiket pada perulangan. *Value progress bar* yang diperiksa adalah *value progress bar* loading ketika memuat *database MySQL* tabel tiket aktif dan tabel tiket nonaktif .

```
● ● ●  
if (total < 0) MessageBox.Show("Uang pelanggan tidak cukup", "Error", MessageBoxButtons.OK,  
    MessageBoxIcon.Error); // messagebox error ditampilkan  
-----  
for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di  
array naamaa tiket  
{  
    comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); //  
    menambahkan pilihan ke comboBox_kelas_layanan.Items dengan array nama dan harga tiket index (i)  
}  
-----  
if (progressBar1.Value < progressBar1.Maximum) progressBar1.Value++; // update progress setiap  
baris
```

c. Operator Increment & Decrement

Operator *increment* digunakan untuk menambah nilai variabel sebanyak satu, sedangkan operator *decrement* digunakan untuk mengurangi nilai variabel sebanyak satu.

Nama Operator	Simbol
<i>increment</i>	++
<i>decrement</i>	--

Pada proyek ini, operator *increment* digunakan untuk menambahkan *value* dari *progress bar loading* ketika memuat *database MySQL* tabel tiket aktif dan tabel tiket nonaktif. Operator *decrement* pada proyek ini tidak digunakan.



```
if (progressBar1.Value < progressBar1.Maximum) progressBar1.Value++; // tambahkan value  
progress bar setiap memuat baris
```

d. Operator Logika (Logical Operators)

Operator logika digunakan untuk menghubungkan dua atau lebih ekspresi kondisi (yang menghasilkan true atau false) dan mengembalikan hasil berupa boolean (true atau false). Hasil operasi dari operator logika sama seperti operator perbandingan, yakni boolean *true* dan *false*.

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

Pada proyek ini, logika AND digunakan untuk memeriksa apakah *checkbox hotel* dipilih dan *textbox nama hotel* telah diisi. Logika negasi/kebalikan digunakan untuk memeriksa bahwa *textbox biaya admin* tidak kosong (*not empty*).

```

● ● ●

if (!checkBox_akomodasi_hotel.Checked &&
!string.IsNullOrEmpty(textBox_nama_hotel_pilihan.Text)) // memeriksa apakah pengguna memilih
akomodasi hotel
{
    textBox_nama_hotel_pilihan.Clear(); // membersihkan textBox_nama_hotel_pilihan
    MessageBox.Show("Anda tidak memilih layanan Akomodasi Hotel", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error); // message box error ditampilkan
}

if (string.IsNullOrWhiteSpace(hotel_pilihan) && checkBox_akomodasi_hotel.Checked) // jika var
hotel_pilihan kosong namun checkBox_akomodasi_hotel dipilih
{
    MessageBox.Show("Nama hotel untuk akomodasi tidak terisi", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
    return; // berhentikan program
}

if (!string.IsNullOrEmpty(hotel_pilihan) && !checkBox_akomodasi_hotel.Checked) // jika var
hotel_pilihan tidak kosong namun checkBox_akomodasi_hotel tidak dipilih
{
    textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
    MessageBox.Show("Anda tidak mencentang layanan Akomodasi Hotel", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    return; // berhentikan program
}

if (!string.IsNullOrEmpty(textBox_biaya_admin.Text)) // jika textBox_biaya_admin.Text tidak
kosong
{
    if (double.TryParse(textBox_biaya_admin.Text, out double biaya_admin)) // jika
textBox_biaya_admin.Text adalah angka, masukkan ke var biaya_admin
    {
        total -= biaya_admin; // total dikurang biaya admin
    }
    else MessageBox.Show("Biaya admin tidak valid", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); // messagebox error ditampilkan
}

```

e. Operator Penugasan

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

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

Pada proyek ini, operator penugasan yang digunakan antara lain:

1) Pengisian nilai

Pada proyek ini, penugasan pengisian nilai digunakan untuk mendefinisikan nilai awal dari suatu variabel dan mengisi kembali nilainya seperti. Selain itu, juga digunakan untuk mengisi item dari *array*, seperti *array* nama tiket, harga tiket, dan nama penerbangan. Penugasan pengisian nilai juga digunakan dalam pengisian nilai variabel pada suatu perulangan.



```
bool invalid = false; // variable bool, untuk menjadi saklar, apakah invalid atau tidak

int harga = 0; // mendefinisikan awal harga menjadi 0
int total_harga = Convert.ToInt32(label_total_harga.Text); // mengisi var total_harga menjadi
// nilai dari label total harga yang diconvert ke int

lbl_informasi.Text = ""; // mengosongkan lbl_informasi.Text

lbl_kembalian.Text = $"Rp. {total}"; // mengganti lbl_kembalian.Text dengan total

FormTiketAktif fta = new FormTiketAktif();

string[] namaTiket = new string[] { "Ekonomi", "Ekonomi Premium", "Bisnis", "Kelas Utama (First
Class)" }; // menginisialisasi nama dari tiket

int[] hargaTiket = new int[] { 500000, 1000000, 1500000, 2000000 }; // menginisialisasi harga
// dari tiket

string[] namaPenerbangan = new string[] { "Indonesia", "Citilink", "Lion Air", "Batik Air",
"AirAsia" };

for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di
// array naamaa tiket
{
    comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); // menambahkan pilihan ke comboBox_kelas_layanan.Items dengan array nama dan harga tiket index (i)
}
```

2) Pengisian dan penambahan

Pada proyek ini, penugasan pengisian dan penambahan digunakan untuk menambahkan nilai dari suatu variabel, baik variabel angka atau teks. Pada informasi tambahan tiket, penugasan pengisian dan penambahan digunakan untuk menambah isi dari label informasi tambahan seperti informasi pribadi, detail perjalanan, dan rincian perjalanan. Selain itu, pada penggabungan nilai dari variabel nomor negara dengan nomor telepon dan penambahan layanan tambahan juga menggunakan penugasan pengisian dan penambahan. Pada variabel angka seperti total harga digunakan untuk menambah nilai dari total harga dengan harga tambahannya.

```

● ● ●

lbl_informasi.Text += "# Informasi Pribadi";
lbl_informasi.Text += "\n • Nama Kontak: ";
lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString() ??
"; // ambil data dari dataGrid_ticket.CurrentRow.Cells
lbl_informasi.Text += "\n • Tanggal lahir: ";
lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_TTL"].Value?.ToString() ?? "";
// ambil data dari dataGrid_ticket.CurrentRow.Cells
lbl_informasi.Text += "\n • E-mail: ";
lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString() ?? "";
// ambil data dari dataGrid_ticket.CurrentRow.Cells
lbl_informasi.Text += "\n • No. Telp: ";
lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nomor_Telp"].Value?.ToString() ??
"; // ambil data dari dataGrid_ticket.CurrentRow.Cells

total_harga += harga; // pengisian dan penambahan nilai var total_harga dengan var harga

if(checkBox_tiket_maskapai.Checked) // jika checkBox_tiket_maskapai dipilih
{
    layanan += "Tiket Maskapai Penerbangan"; // menambahkan pilihan ke var layanan
}
if (checkBox_rent_car.Checked) // jika checkBox_rent_car dipilih
{
    layanan += ", Rental Mobil"; // menambahkan pilihan ke var layanan
}
if (checkBox_akomodasi_hotel.Checked) // jika checkBox_akomodasi_hotel dipilih
{
    layanan += ", Akomodasi Hotel"; // menambahkan pilihan ke var layanan
}

if (int.TryParse(textBox_nomor_telepon.Text, out int nomor_telepon)) // mendekripsi kalau
textBox_nomor_telepon.Text adalah angka, dan dimasukkan ke var nomor_telepon
{
    nomor_negara += $", {nomor_telepon}"; // menggabungkan nomor_negara dengan nomor telepon
}

```

3) Pengisian dan pengurangan

Pada proyek ini, penugasan pengisian dan pengurangan digunakan untuk mengurangi nilai dari suatu variabel. Penerapannya ketika pengguna menghapus tiket yang sudah dikirim sebelumnya akan mengurangi nilai dari total harga sesuai dengan harga tiket yang telah dihapus.

```

● ● ●

if (double.TryParse(textBox_biaya_admin.Text, out double biaya_admin)) // jika
textBox_biaya_admin.Text adalah angka, masukkan ke var biaya_admin
{
    total -= biaya_admin; // total dikurang biaya admin
}

total_harga -= Convert.ToInt32(dataGrid_ticket.CurrentRow.Cells["Total_Harga"].Value); // var
total_harga lalu di kurang nilai dari sel Total_Harga pada dataGrid_ticket row saat ini yang
sudah diubah ke int

```

4. Percabangan Kondisi

Percabangan kondisi dalam pemrograman adalah konsep yang 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).

Pada proyek ini, percabangan kondisi yang digunakan antara lain:

a. Pernyataan if

Pernyataan if digunakan untuk menjalankan sebuah blok kode hanya jika kondisinya bernilai *true* (benar). Pada proyek ini, pernyataan *if* digunakan untuk memeriksa kolom nama hotel dari *data grid view* daftar tiket dan memeriksa kolom yang dipilih pengguna pada *data grid view* tiket *draft*, tiket aktif, dan tiket mati. Pemeriksaan *length* dari *array* nama penerbangan, serta pemeriksaan pada *checkbox* nama hotel, dan layanan tambahan.

```
● ● ●

if(!string.IsNullOrEmpty(dataGrid_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString()))
{
    lbl_informasi.Text += "\n • Hotel Pilihan: ";
    lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString() ?? ""; // ambil data
    dari dataGrid_ticket.CurrentRow.Cells
}

if (!checkBox_akkomodasi_hotel.Checked &&
!string.IsNullOrEmpty(textBox_nama_hotel_pilihan.Text)) // memeriksa apakah pengguna memilih
akkomodasi hotel
{
    textBox_nama_hotel_pilihan.Clear(); // membersihkan textBox_nama_hotel_pilihan
    MessageBox.Show("Anda tidak memilih layanan Akkomodasi Hotel", "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error); // message box error ditampilkan
}

total_harga -= Convert.ToInt32(dataGrid_ticket.CurrentRow.Cells["Total_Harga"].Value); // var
total harga lalu di kurang nilai dari sel Total_Harga pada dataGrid_ticket row saat ini yang
sudah diubah ke int

if(checkBox_tiket_maskapai.Checked) // jika checkBox_tiket_maskapai dipilih
{
    layanan += "Tiket Maskapai Penerbangan"; // menambahkan pilihan ke var layanan
}

if (string.IsNullOrWhiteSpace(hotel_pilihan) && checkBox_akkomodasi_hotel.Checked) // jika var
hotel_pilihan kosong namun checkBox_akkomodasi_hotel dipilih
{
    MessageBox.Show("Nama hotel untuk akomodasi tidak terisi", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
    return; // berhentikan program
}
```

b. Pernyataan if-else

Pernyataan *if-else* digunakan ketika kita memiliki dua pilihan. Jika kondisi *if* benar, jalankan blok *if*. Jika tidak (*else*), jalankan blok *else*. Pada proyek ini, digunakan untuk mendeteksi apakah nomor telepon valid atau tidak. Memeriksa apakah input waktu keberangkatan apakah valid atau tidak. Selain itu juga memeriksa apakah pengguna telah memilih kolom pada *data grid view* atau tidak.

```

● ● ●

if (int.TryParse(textBox_nomor_telepon.Text, out int nomor_telepon)) // mendeteksi kalau
    textBox_nomor_telepon.Text adalah angka, dan dimasukkan ke var nomor_telepon
{
    nomor_negara += $", {nomor_telepon}"; // menggabungkan nomor_negara dengan nomor telepon
}
else // jika textBox_nomor_telepon.Text tidak angka
{
    MessageBox.Show("Nomor telepon tidak valid", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    textBox_nomor_telepon.Focus(); // fokus ke textBox_nomor_telepon
    return; //berhentikan program
}

if (DateTime.TryParseExact(input_berangkat, "HH:mm",
    System.Globalization.CultureInfo.InvariantCulture, System.Globalization.DateTimeStyles.None,
    out DateTime waktu))
{
}

else
{
    // ✗ Jika tidak valid
    MessageBox.Show("Waktu berangkat tidak valid! Gunakan format HH:mm (contoh: 14:30)",
        "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // message box
    warning ditampilkan
    maskedTextBox_waktu_berangkat.Focus(); // fokus ke maskedTextBox_waktu_berangkat
    invalid = true; // invalid menjadi true
    return; // berhentikan program
}

if (dataGridView_ticket.CurrentRow != null) // jika pengguna tidak memilih baris dari
    dataGridView_ticket
{
    int total_harga = Convert.ToInt32(label_total_harga.Text); // isi var total harga dengan
    label_total_harga.Text yang diubah menjadi int
    total_harga -= Convert.ToInt32(dataGridView_ticket.CurrentRow.Cells["Total_Harga"].Value); // var total harga lalu di kurang nilai dari sel Total_Harga pada dataGridView_ticket row saat ini
    yang sudah diubah ke int
    label_total_harga.Text = $"{total_harga}"; // set label_total_harga.Text ke
    label_total_harga.Text
    dataGridView_ticket.Rows.RemoveAt(dataGridView_ticket.CurrentRow.Index); // menghapus baris yang
    dipilih
    lbl_informasi.Text = ""; // mengosongkan lbl_informasi.Text
}
else
{
    MessageBox.Show("Pilih baris tiket yang ingin dihapus.", "Error", MessageBoxButtons.OK,
    MessageBoxIcon.Error); // message box error di tampilkan
}

```

c. Pernyataan if-else if-else

Pernyataan *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. Pada proyek ini, Pernyataan *if-else if-else* digunakan untuk memeriksa apakah pengguna memilih satu pilihan dari pilihan yang ada di *combobox* kelas layanan atau tidak.

```

● ● ●

// tiket default (ekonomi, ekonomi premium, bisnis, kelas atas)
if (comboBox_kelas_layanan.SelectedIndex == 0) // jika comboBox_kelas_layanan dipilih index 0
{
    harga = hargaTiket[0]; // var harga diubah menyesuaikan dengan array hargaTiket index 0
}
else if (comboBox_kelas_layanan.SelectedIndex == 1) // jika tidak dan comboBox_kelas_layanan dipilih index 1
{
    harga = hargaTiket[1]; // var harga diubah menyesuaikan dengan array hargaTiket index 1
}
else if (comboBox_kelas_layanan.SelectedIndex == 2) // jika tidak dan comboBox_kelas_layanan dipilih index 2
{
    harga = hargaTiket[2]; // var harga diubah menyesuaikan dengan array hargaTiket index 2
}
else if (comboBox_kelas_layanan.SelectedIndex == 3) // jika tidak dan comboBox_kelas_layanan dipilih index 3
{
    harga = hargaTiket[3]; // var harga diubah menyesuaikan dengan array hargaTiket index 3
}
else // tiket tambahan
{
    harga = hargaTiket[comboBox_kelas_layanan.SelectedIndex]; // var harga diubah menyesuaikan dengan array hargaTiket index sesuai index pilihan
}

```

d. Nested if (if bersarang)

Nested if (if bersarang) dalam C# adalah pernyataan *if* yang diletakkan di dalam pernyataan *if* lainnya, sehingga menciptakan struktur percabangan yang lebih kompleks untuk mengevaluasi beberapa kondisi secara hierarkis. Pada proyek ini, *nested if (if bersarang)* digunakan untuk pemeriksaan bertahap dari memeriksa apakah uang pelanggan valid atau tidak, lalu dilanjutkan pemeriksaan input biaya admin kosong atau tidak, lalu dilanjutkan juga pemeriksaan apakah input biaya admin angka yang valid atau tidak, dan akan menentukan harga tiket.

```

● ● ●

if (double.TryParse(textBox_uang_pelanggan.Text, out double uang_pelanggan)) // jika textBox_uang_pelanggan.Text adalah angka, dimasukkan ke var uang_pelanggan
{
    double total = uang_pelanggan - Convert.ToInt32(label_total_harga.Text); // set var total (total kembalian) dengan uang pelanggan - label_total_harga.Text yang sudah diubah ke int
    if (!string.IsNullOrEmpty(textBox_biaya_admin.Text)) // jika textBox_biaya_admin.Text tidak kosong
    {
        if (double.TryParse(textBox_biaya_admin.Text, out double biaya_admin)) // jika textBox_biaya_admin.Text adalah angka, masukkan ke var biaya_admin
        {
            total -= biaya_admin; // total dikurang biaya admin
        }
        else MessageBox.Show("Biaya admin tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
    }
    lbl_kembalian.Text = $"Rp. {total}"; // mengganti lbl_kembalian.Text dengan total
    if (total < 0) MessageBox.Show("Uang pelanggan tidak cukup", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
}

```

e. Pernyataan switch-case

Pernyataan *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. Pada proyek ini, Pernyataan *if-else if-else* digunakan untuk memeriksa apakah pengguna memilih satu pilihan dari pilihan yang ada di *combobox* kelas layanan atau tidak, dan akan menentukan nama tiket.

```
● ● ●

switch (comboBox_kelas_layanan.SelectedIndex) // membuat pilihan dari
comboBox_kelas_layanan.SelectedIndex
{
    case 0:
        kelas_layanan = "Ekonomi"; // kelas layanan di ubah
        break;
    case 1:
        kelas_layanan = "Ekonomi Premium"; // kelas layanan di ubah
        break;
    case 2:
        kelas_layanan = "Bisnis"; // kelas layanan di ubah
        break;
    case 3:
        kelas_layanan = "Kelas Utama (First Class)"; // kelas layanan di ubah
        break;
    default:
        break;
}
```

5. Array

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* adalah isi dari *array*. *Indeks* adalah urutan *array* (dimulai dari 0).

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

Urutan indeks dalam *array* dimulai dari angka 0, kemudian 1, 2, dan seterusnya. Ini berarti elemen pertama dalam *array* berada pada indeks 0, elemen kedua pada indeks 1, dan seterusnya.

Pada proyek ini, *array* digunakan untuk mendefinisikan dan menyimpan data dari nama tiket, harga tiket, dan nama penerbangan.



```
string[] namaTiket = new string[] { "Ekonomi", "Ekonomi Premium", "Bisnis", "Kelas Utama (First Class)" }; // menginisialisasi nama dari tiket

int[] hargaTiket = new int[] { 500000, 1000000, 1500000, 2000000 }; // menginisialisasi harga dari tiket

string[] namaPenerbangan = new string[] { "Indonesia", "Citilink", "Lion Air", "Batik Air", "AirAsia" };

for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di array naamaa tiket
{
    comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); // menambahkan pilihan ke comboBox_kelas_layanan.Items dengan array nama dan harga tiket index (i)
}
```

6. Perulangan (Looping)

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

Nama Perulangan	Fungsi
for	Digunakan ketika jumlah perulangan sudah diketahui di awal.
while	Digunakan ketika perulangan harus berlanjut selama suatu kondisi benar. Kondisi diperiksa sebelum blok kode dieksekusi.
do-while	Mirip dengan <i>while</i> , tetapi blok kode dieksekusi minimal satu kali sebelum kondisi diperiksa.
foreach	Digunakan untuk mengiterasi elemen dalam suatu koleksi (seperti <i>array</i> atau <i>list</i>) tanpa perlu menggunakan indeks.
Nested Loop	<i>Nested Loop</i> (perulangan bersarang) adalah perulangan di dalam perulangan lain. Artinya, satu <i>loop</i> berada di dalam <i>loop</i> lainnya, dan <i>loop</i> bagian dalam akan dijalankan sepenuhnya untuk setiap iterasi <i>loop</i> bagian luar.

Pada proyek ini, perulangan yang digunakan antara lain:

a. Perulangan for

Digunakan ketika jumlah perulangan sudah diketahui di awal. Pada proyek ini, perulangan *for* mengulang penambahan pilihan pada *combobox* sebanyak *length* dari *array* nama tiket.

```
● ● ●

for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di array naamaa tiket
{
    comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); // menambahkan pilihan ke comboBox_kelas_layanan.Items dengan array nama dan harga tiket index (i)
}
```

b. Perulangan while

Digunakan ketika perulangan harus berlanjut selama suatu kondisi benar. Kondisi diperiksa sebelum blok kode dieksekusi. Pada proyek ini, digunakan untuk melakukan perulangan dalam penambahan baris pada *data grid view* daftar tiket aktif dan daftar tiket non aktif sebanyak baris yang ada pada *database MySQL* tabel tiket dan tiket mati.

```
● ● ●

while (rd.Read())
{
    // Tambahkan row baru
    dataGrid_active_ticket.Rows.Add(
        rd["id"].ToString(),
        rd["nama_kontak"].ToString(),
        rd["tanggal_lahir"].ToString(),
        rd["email"].ToString(),
        rd["no_telp"].ToString(),
        rd["negara_keberangkatan"].ToString(),
        rd["negara_tujuan"].ToString(),
        rd["tanggal_waktu_keberangkatan"].ToString(),
        rd["kelas_layanan"].ToString(),
        rd["layanan_lainnya"].ToString(),
        rd["hotel_pilihan"].ToString(),
        rd["penerbangan_pilihan"].ToString(),
        rd["informasi_tambahan"].ToString(),
        rd["harga_tiket"].ToString()
    );

    if (progressBar1.Value < progressBar1.Maximum)
        progressBar1.Value++; // ambahkan value progress bar setiap memuat baris
}
```

c. Perulangan do-while

Digunakan ketika perulangan harus berlanjut selama suatu kondisi benar. Kondisi diperiksa sebelum blok kode dieksekusi. Pada proyek ini, digunakan untuk mengulang penambahan *combobox* penerbangan sebanyak *length* dari *array* nama penerbangan.

```

● ● ●

if(namaPenerbangan.Length > 0) // jika array namaPenerbangan item nya lebih dari 0
{
    int totalnamaPenerbangan = 0; // set jumlah awal 0
    do // lakukan sekali
    {
        comboBox_penerbangan.Items.Add($"{namaPenerbangan[totalnamaPenerbangan]}"); // tambahkan pilihan di comboBox_penerbangan.Items sesuai index dari array namaPenerbangan
        totalnamaPenerbangan++; // tambah 1
    }
    while (totalnamaPenerbangan < namaPenerbangan.Length); // jika masih kurang dari jumlah
    nama penerbangan ulangi lagi
}

```

7. Fungsi (Method)

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 <i>return</i>	Hanya menampilkan pesan atau melakukan tugas tanpa hasil balik.	void (Sapa)
Dengan parameter	Menerima data input dari luar.	void SapaNama(string nama)
Dengan <i>return</i>	Menghasilkan nilai dari perhitungan.	int Tambah(int a, int b)
Dengan parameter & <i>return</i>	Lengkap, menerima input dan menghasilkan output.	double HitungLuas(double p, double l)

Pada proyek ini, fungsi (*method*) yang digunakan antara lain:

- a. Tanpa parameter dan tanpa return

Hanya menampilkan pesan atau melakukan tugas tanpa hasil balik. Pada proyek ini, dibuat fungsi yang digunakan untuk menampilkan data dengan menambahkan baris ke *data grid view* dari tabel yang ada di *database MySQL* tabel tiket dan tiket mati.

```

● ● ●

private void TampilkanData()
{
    string koneksi = "server=localhost;user id=root;password=;database=katasi;" // menulis
    database mysql yang akan dihubungkan
    MySqlConnection conn = new MySqlConnection(koneksi); // hubungkan mysql
    conn.Open(); // buka mysql

    // 1. Hitung total baris dulu
    MySqlCommand countCmd = new MySqlCommand("SELECT COUNT(*) FROM tiket", conn);
    int total = Convert.ToInt32(countCmd.ExecuteScalar());

    // Set progress bar
    progressBar1.Value = 0;
    progressBar1.Maximum = total;
    progressBar1.Visible = true;

    // 2. Ambil data

    string query = "SELECT * FROM tiket"; // perintah untuk memilih baris di tabel tiket
    MySqlCommand cmd = new MySqlCommand(query, conn);
    MySqlDataReader rd = cmd.ExecuteReader();

    dataGrid_active_ticket.Rows.Clear(); // menghapus isi lama
    while (rd.Read())
    {
        // Tambahkan row baru
        dataGrid_active_ticket.Rows.Add(
            rd["id"].ToString(),
            rd["nama_kontak"].ToString(),
            rd["tanggal_lahir"].ToString(),
            rd["email"].ToString(),
            rd["no_telp"].ToString(),
            rd["negara_keberangkatan"].ToString(),
            rd["negara_tujuan"].ToString(),
            rd["tanggal_waktu_keberangkatan"].ToString(),
            rd["kelas_layanan"].ToString(),
            rd["layanan_lainnya"].ToString(),
            rd["hotel_pilihan"].ToString(),
            rd["penerbangan_pilihan"].ToString(),
            rd["informasi_tambahan"].ToString(),
            rd["harga_tiket"].ToString()
        );

        if (progressBar1.Value < progressBar1.Maximum)
            progressBar1.Value++; // ambahkan value progress bar setiap memuat baris
    }

    rd.Close();
    conn.Close();

    // 4. Sembunyikan progress bar setelah selesai
    progressBar1.Visible = false;
}

```

b. Dengan parameter

Menerima data input dari luar. Pada proyek ini, dibuat fungsi yang digunakan untuk memindahkan dan menghapus baris yang ada pada *data grid view* sekaligus baris yang ada pada *database MySQL* tabel tiket dan tiket mati.

```

● ● ●

private void HapusData(string id) // fungsi hapus tiket sesuai id
{
    string koneksi = "server=localhost;user id=root;password=;database=katas"; // menulis
    database mysql yang akan dihubungkan
    using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
    {
        conn.Open(); // buka mysql
        string query = "DELETE FROM tiket WHERE id = @id"; // perintah querry untuk hapus pada
        baris sesuai id
        MySqlCommand cmd = new MySqlCommand(query, conn); // membuat cmd menjadi perintah
        cmd.Parameters.AddWithValue("@id", id); // tambah parameter ID ke perintah
        cmd.ExecuteNonQuery(); // di perintah di eksekusi
    }
}

private void PindahkanBaris(string id) // fungsi untuk pindahkan baris ke tabel mati
{
    string koneksi = "server=localhost;user id=root;password=;database=katas"; // menulis
    database mysql yang akan dihubungkan
    using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
    {
        conn.Open(); // buka mysql

        // 1) Insert ke tabel tujuan
        string insertQuery = "INSERT INTO tiket_mati SELECT * FROM tiket WHERE id = @id"; // perintah querry sql, INSERT untuk menambah baris baru ke tabel_mati yang sama seperti baris yang ada di tabel tiket
        MySqlCommand cmdInsert = new MySqlCommand(insertQuery, conn); // membuat cmdinsert dari querry
        cmdInsert.Parameters.AddWithValue("@id", id); // penambahan parameter id baris
        cmdInsert.ExecuteNonQuery(); // perintah di eksekusi

        // 2) Hapus dari tabel asal
        string deleteQuery = "DELETE FROM tiket WHERE id = @id"; // perintah querry dihapus
        baris sebelumnya dari tabel tiket
        MySqlCommand cmdDelete = new MySqlCommand(deleteQuery, conn); // cmddelete dibuat dari querry
        cmdDelete.Parameters.AddWithValue("@id", id); // parameter id ditambahkan ke cmd querry
        cmdDelete.ExecuteNonQuery(); // eksesuki cmd
    }
}

```

c. Dengan parameter & return

Lengkap, menerima input dan menghasilkan output. Pada proyek ini, dibuat fungsi yang digunakan untuk memeriksa string apakah email yang valid atau tidak. Juga fungsi untuk format angka ke rupiah.

```

● ● ●

private bool IsValidEmail(string email) // fungsi untuk mendeteksi apakah suatu string adalah
email yang valid
{
    if (string.IsNullOrWhiteSpace(email)) // jika email kosong
        return false;

    // Pola regex sederhana dan umum digunakan
    string pattern = @"^@[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"; // mendeteksi apakah
    ada char '@' dan char '.'

    return Regex.IsMatch(email, pattern); // jika ada lanjut return, nilai true
}

```

```

● ● ●

public static string FormatMoney(int angka)
{
    return angka.ToString("#,0").Replace(",",".");
}

```

D. Perancangan Program

1. Konsep Awal

Aplikasi ini dirancang untuk dapat membuat tiket dengan menyediakan formulir dan data tiket akan tersimpan secara permanen di basis data MySQL *localhost*, serta dilengkapi dengan bantuan administrasi yang mempermudah dalam penghitungan akumulasi tiket. Tiket yang telah dibuat dapat dijadikan menjadi tiket aktif dan tersimpan pada tabel tiket aktif basis data MySQL. Tiket aktif dapat dimatikan sehingga masuk ke daftar tiket mati dan terpisah dengan tabel tiket aktif dan memasuki tabel tiket mati basis data MySQL.

Setiap tiket telah ditandai dengan identitas angka, sehingga dapat diverifikasi dan diperiksa statusnya. Selain itu, aplikasi ini juga dilengkapi fitur koreksi yang meminimalisir terjadinya kesalahan dalam pengiriman input.

2. Struktur Program

- a. FormMenu.cs (Utama)
- b. FormFormulir.cs (Membuat formulir)
- c. FormTiketAktif.cs (Daftar tiket aktif)
- d. FormTiketMati.cs (Daftar tiket mati)
- e. FormVerifikasi.cs (Verifikasi tiket)

3. Aset Program

- a. Logo Aplikasi



Gambar 1.2. Logo Aplikasi “Kapal Atas Indonesia”

- b. Desain Tombol



Gambar 1.3 Desain Tombol Buat Tiket Baru



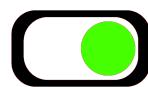
Gambar 1.4 Desain Tombol Daftar Tiket Aktif



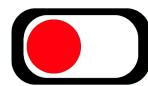
Gambar 1.5 Desain Tombol Daftar Tiket Mati



Gambar 1.6 Desain Tombol Verifikasi Tiket



Gambar 1.7 Desain Tombol Aktifkan Tiket



Gambar 1.8 Desain Tombol Matikan Tiket



Gambar 1.9 Desain Tombol Hapus Tiket



Gambar 1.10 Desain Tombol Refresh Data Tiket



Gambar 1.11 Desain Tombol Menu



Gambar 1.12 Desain Tombol Keluar Aplikasi

BAB II

IMPLEMENTASI PROGRAM

A. Potongan Kode Program Utama

1. Kirim Formulir

```
● ● ●

private void btn_kirim_formulir_Click(object sender, EventArgs e) // jika tombol kirim di klik
{
    bool invalid = false; // variable bool, untuk menjadi saklar, apakah invalid atau tidak

    string nama = textBox_nama.Text; // variable nama diset ke textBox_nama.Text
    string ttl = dateTime_ttl.Text; // variable ttl diset ke dateTime_ttl.Text
    string email = textBox_email.Text; // variable email diset ke textBox_email.Text
    string nomor_negara = comboBox_nomor_negara.SelectedItem?.ToString(); // mengeset variable nomor_negara ke
    comboBox_nomor_negara.SelectedItem yang diubah dulu menjadi string.
    string negara_keberangkatan = textBox_keberangkatan.Text; // mengeset var negara_keberangkatan ke textBox_keberangkatan.Text
    string negara_tujuan = textBox_tujuan.Text; // mengeset var negara_tujuan ke textBox_tujuan.Text
    string tanggal_berangkat = $"{dateTime_keberangkatan.Text}, {maskedTextBox_waktu_berangkat.Text}"; // var tanggal_berangkat di set
    menjadi gabungan dateTime_keberangkatan.Text dengan maskedTextBox_waktu_berangkat.Text
    string input_berangkat = maskedTextBox_waktu_berangkat.Text; // var input_berangkat diset ke maskedTextBox_waktu_berangkat.Text
    string kelas_layanan = ""; // membuat var kelas_layanan dengan tidak ada isinya
    string layanan = ""; // var layanan dengan isi kosong
    string hotel_pilihan = textBox_nama_hotel_pilihan.Text; // set var hotel_pilihan ke textBox_nama_hotel_pilihan.Text
    string penerbangan_pilihan = comboBox_penerbangan.Text; // set var penerbangan_pilihan ke comboBox_penerbangan.Text
    string informasi_tambahan = textBox_informasi_tambahan.Text; //set var informasi_tambahan ke textBox_informasi_tambahan.Text

    if (string.IsNullOrWhiteSpace(nama)) //mendeteksi apakah variable "nama" kosong
    {
        MessageBox.Show("Nama kontak tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid
        menjadi true
        textBox_nama.Focus(); //focus ke textBox_nama
        return; // berhentikan program
    }
    if (!IsValidEmail(email)) // mendeteksi apakah variable "mail" adalah email yang valid
    {
        MessageBox.Show("Email tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi
        true
        textBox_email.Focus(); // fokus ke textBox_email
        return; // berhentikan program
    }
    if (comboBox_nomor_negara.SelectedIndex == -1) // jika pengguna tidak memilih comboBox_nomor_negara
    {
        MessageBox.Show("Nomor negara tidak dipilih", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid
        menjadi true
        comboBox_nomor_negara.Focus(); // fokus ke comboBox_nomor_negara
        return; // berhentikan program
    }
    if (int.TryParse(textBox_nomor_telepon.Text, out int nomor_telepon)) // mendeteksi kalau textBox_nomor_telepon.Text adalah angka,
    dan dimasukkan ke var nomor_telepon
    {
        nomor_negara += $", {nomor_telepon}"; // menggabungkan nomor_negara dengan nomor telepon
    }
    else // jika textBox_nomor_telepon.Text tidak angka
    {
        MessageBox.Show("Nomor telepon tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid
        menjadi true
        textBox_nomor_telepon.Focus(); // fokus ke textBox_nomor_telepon
        return; //berhentikan program
    }
    if (string.IsNullOrWhiteSpace(negara_keberangkatan)) // jika var negara_keberangkatan kosong
    {
        MessageBox.Show("Negara keberangkatan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid
        menjadi true
        textBox_keberangkatan.Focus(); // fokus ke textBox_keberangkatan
        return; // berhentikan program
    }
```

```

        }
        if (string.IsNullOrWhiteSpace(negara_keberangkatan)) // jika var negara_keberangkatan kosong
        {
            MessageBox.Show("Negara keberangkatan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi true
            txtBox_keberangkatan.Focus(); // fokus ke txtBox_keberangkatan
            return; // berhentikan program
        }
        if (string.IsNullOrWhiteSpace(negara_tujuan)) // jika negara_tujuan kosong
        {
            MessageBox.Show("Negara tujuan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi true
            txtBox_tujuan.Focus(); // fokus ke txtBox_tujuan
            return; // berhentikan program
        }
        // Pastikan user sudah isi lengkap (misal "14:30")
        if (!maskedTextBox_waktu_berangkat.MaskCompleted) // jika maskedTextBox_waktu_berangkat tidak diisi penuh
        {
            MessageBox.Show("Input waktu berangkat belum lengkap! Format: HH:mm",
                "Validasi Waktu", MessageBoxButtons.OK, MessageBoxIcon.Warning); // message box warning ditampilkan
            maskedTextBox_waktu_berangkat.Focus(); // fokus ke maskedTextBox_waktu_berangkat
            invalid = true; // invalid menjadi true
            return; // berhentikan program
        }
        // Coba parse input jadi DateTime
        if (DateTime.TryParseExact(input_berangkat, "HH:mm",
            System.Globalization.CultureInfo.InvariantCulture,
            System.Globalization.DateTimeStyles.None,
            out DateTime waktu))
        {
        }
        else
        {
            // ✗ Jika tidak valid
            MessageBox.Show("Waktu berangkat tidak valid! Gunakan format HH:mm (contoh: 14:30)", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // message box warning ditampilkan
            maskedTextBox_waktu_berangkat.Focus(); // fokus ke maskedTextBox_waktu_berangkat
            invalid = true; // invalid menjadi true
            return; // berhentikan program
        }
        if (comboBox_kelas_layanan.SelectedIndex == -1) // jika comboBox_kelas_layanan tidak dipilih
        {
            MessageBox.Show("Kelas layanan tidak dipilih", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi true
            comboBox_kelas_layanan.Focus(); // fokus ke comboBox_kelas_layanan;
            return; // program dihentikan
        }
        switch (comboBox_kelas_layanan.SelectedIndex) // membuat pilihan dari comboBox_kelas_layanan.SelectedIndex
        {
            case 0:
                kelas_layanan = "Ekonomi"; // kelas layanan di ubah
                break;
            case 1:
                kelas_layanan = "Ekonomi Premium"; // kelas layanan di ubah
                break;
            case 2:
                kelas_layanan = "Bisnis"; // kelas layanan di ubah
                break;
            case 3:
                kelas_layanan = "Kelas Utama (First Class)"; // kelas layanan di ubah
                break;
            default:
                break;
        }
        if(checkBox_tiket_maskapai.Checked) // jika checkBox_tiket_maskapai dipilih
        {
            layanan += "Tiket Maskapai Penerbangan"; // menambahkan pilihan ke var layanan
        }
        if (checkBox_rent_car.Checked) // jika checkBox_rent_car dipilih

```

```

    {
        layanan += ", Rental Mobil"; // menambahkan pilihan ke var layanan
    }
    if (checkBox_akkomodasi_hotel.Checked) // jika checkBox_akkomodasi_hotel dipilih
    {
        layanan += ", Akomodasi Hotel"; // menambahkan pilihan ke var layanan
    }
    if (string.IsNullOrWhiteSpace(hotel_pilihan) && checkBox_akkomodasi_hotel.Checked) // jika var hotel_pilihan kosong namun
    checkBox_akkomodasi_hotel dipilih
    {
        MessageBox.Show("Nama hotel untuk akomodasi tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid =
        true; // invalid menjadi true
        textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
        return; // berhentikan program
    }
    if (!string.IsNullOrEmpty(hotel_pilihan) && !checkBox_akkomodasi_hotel.Checked) // jika var hotel_pilihan tidak kosong namun
    checkBox_akkomodasi_hotel tidak dipilih
    {
        textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
        MessageBox.Show("Anda tidak mencantang layanan Akomodasi Hotel", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid =
        true; // invalid menjadi true
        return; // berhentikan program
    }
    if (string.IsNullOrWhiteSpace(penerbangan_pilihan)) // jika var penerbangan_pilihan kosong
    {
        MessageBox.Show("Penerbangan pilihan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; //
        invalid menjadi true
        comboBox_penerbangan.Focus(); // fokus ke comboBox_penerbangan
        return; // berhentikan program
    }
    int harga = 0; // mendefinisikan awal harga menjadi 0
    int total_harga = Convert.ToInt32(label_total_harga.Text); // mengisi var total_harga menjadi nilai dari label total harga yang
    diconvert ke int
    // tiket default (ekonomi, ekonomi premium, bisnis, kelas atas)
    if (comboBox_kelas_layanan.SelectedIndex == 0) // jika comboBox_kelas_layanan dipilih index 0
    {
        harga = hargaTiket[0]; // var harga diubah menyesuaikan dengan array hargaTiket index 0
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 1) // jika tidak dan comboBox_kelas_layanan dipilih index 1
    {
        harga = hargaTiket[1]; // var harga diubah menyesuaikan dengan array hargaTiket index 1
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 2) // jika tidak dan comboBox_kelas_layanan dipilih index 2
    {
        harga = hargaTiket[2]; // var harga diubah menyesuaikan dengan array hargaTiket index 2
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 3) // jika tidak dan comboBox_kelas_layanan dipilih index 3
    {
        harga = hargaTiket[3]; // var harga diubah menyesuaikan dengan array hargaTiket index 3
    }
    else // tiket tambahan
    {
        harga = hargaTiket[comboBox_kelas_layanan.SelectedIndex]; // var harga diubah menyesuaikan dengan array hargaTiket index
        sesuai index pilihan
    }
    if (invalid == false) // jika nilai invalid tidak true
    {
        dataGrid_ticket.Rows.Add(nama, ttl, email, nomor_negara, negara_keberangkatan, negara_tujuan, tanggal_berangkat,
        kelas_layanan, layanan, hotel_pilihan, penerbangan_pilihan, informasi_tambahan, harga); // tambahkan var tadi ke dataGrid_ticket.Rows
        dengan urutan yang pas
    }
}

total_harga += harga; // pengisian dan penambahan nilai var total_harga dengan var harga
label_total_harga.Text = $"{total_harga}"; // mengubah label_total_harga.Text sesuai dengan harga
}
}

```

2. Hitung Kembalian

```

● ● ●

private void button1_Click(object sender, EventArgs e) // ketika tombol hitung kembalian di klik
{
    if (double.TryParse(textBox_uang_pelanggan.Text, out double uang_pelanggan)) // jika textBox_uang_pelanggan.Text adalah angka,
    dimasukkan ke var uang_pelanggan
    {
        double total = uang_pelanggan - Convert.ToInt32(label_total_harga.Text); // set var total (total kembalian) dengan uang
        pelanggan - label_total_harga.Text yang sudah diubah ke int
        if (!string.IsNullOrEmpty(textBox_biaya_admin.Text)) // jika textBox_biaya_admin.Text tidak kosong
        {
            if (double.TryParse(textBox_biaya_admin.Text, out double biaya_admin)) // jika textBox_biaya_admin.Text adalah angka,
            masukkan ke var biaya_admin
            {
                total -= biaya_admin; // total dikurang biaya admin
            }
            else MessageBox.Show("Biaya admin tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error
            ditampilkan
        }
        lbl_kembalian.Text = $"Rp. {total}"; // mengganti lbl_kembalian.Text dengan total
        if (total < 0) MessageBox.Show("Uang pelanggan tidak cukup", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // //
        messagebox error ditampilkan
    }
    // jika textBox_uang_pelanggan.Text bukan angka
    else MessageBox.Show("Uang pelanggan tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error
    ditampilkan
}

```

3. Lihat Informasi Tambahan Tiket yang Dipesan

```
● ● ●

private void button_informasi_tambahan_Click_1(object sender, EventArgs e) // ketika tombol informasi tambahan di klik
{
    if (dataGridView_ticket.CurrentRow != null) // jika row dari dataGridView_ticket tidak ada yang dipilih.
    {
        lbl_informasi.Text = ""; // dataGridView_ticket dikosongkan terlebih dahulu
        lbl_informasi.Text += "# Informasi Pribadi";
        lbl_informasi.Text += "\n * Nama Kontak: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Tanggal lahir: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_TTL"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * E-mail: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * No. Telp: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Nomor_Telp"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells

        lbl_informasi.Text += "\n\n# Detail Perjalanan";
        lbl_informasi.Text += "\n * Negara Keberangkatan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? ""; // ambil data
        dari dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Negara Tujuan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? ""; // ambil
        data dari dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Kelas Layanan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Layanan Lainnya: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Layanan"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells

        lbl_informasi.Text += "\n\n# Rincian Perjalanan";
        if(!string.IsNullOrEmpty(dataGridView_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString()))
        {
            lbl_informasi.Text += "\n * Hotel Pilihan: ";
            lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString() ?? ""; // ambil data dari
            dataGridView_ticket.CurrentRow.Cells
        }
        lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Penerbangan"].Value?.ToString() ?? ""; // ambil data dari
        dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Informasi Tambahan: ";
        lbl_informasi.Text += dataGridView_ticket.CurrentRow.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? ""; // ambil data
        dari dataGridView_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Harga Tiket: ";
        lbl_informasi.Text += "Rp. " + FormatMoney(Convert.ToInt32(dataGridView_ticket.CurrentRow.Cells["Total_Harga"].Value)); // ambil
        data dari dataGridView_ticket.CurrentRow.Cells, lalu di convert ke int dan diformat ke money,
    }
    else
    {
        MessageBox.Show("Pilih baris tiket yang ingin anda lihat informasi tambahannya", "Error", MessageBoxButtons.OK,
        MessageBoxIcon.Error); // messagebox error ditampilkan
    }
}
```

4. Jadikan Tiket Aktif Tiket yang Dipesan

```
● ● ●

private void button_aktifkan_tiket_Click(object sender, EventArgs e)
{
    if (dataGridView_ticket.CurrentRow == null) // jika dataGridView_ticket tidak ada baris yang dipilih
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
        return; // program diberhentikan
    }

    string koneksi = "server=localhost;user id=root;password=;database=katas"; // menulis database mysql yang akan dihubungkan
    using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
    {
        conn.Open(); // buka mysql

        string query = @"INSERT INTO tiket
        (@nama_kontak, @tanggal_lahir, @email, @no_telp, @negara_keberangkatan, @negara_tujuan,
        @tanggal_waktu_keberangkatan, @kelas_layanan, @layanan_lainnya, @hotel_pilihan,
        @penerbangan_pilihan, @informasi_tambahan, @harga_tiket)
        VALUES
        (@nama_kontak, @tanggal_lahir, @email, @no_telp, @negara_keberangkatan, @negara_tujuan,
        @tanggal_waktu_keberangkatan, @kelas_layanan, @layanan_lainnya, @hotel_pilihan,
        @penerbangan_pilihan, @informasi_tambahan, @harga_tiket)"; // perintah query sql, (INSERT INTO, VALUES, dll.) untuk menambah baris baru sesuai parameter

        MySqlCommand cmd = new MySqlCommand(query, conn); // perintah query

        DataGridViewRow r = dataGridView_ticket.CurrentRow;
        cmd.Parameters.AddWithValue("@nama_kontak", r.Cells["Column_Nama_Kontak"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@tanggal_lahir", r.Cells["Column_TTL"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@email", r.Cells["Column_Email"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@no_telp", r.Cells["Column_Nomor_Telp"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@negara_keberangkatan", r.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@negara_tujuan", r.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@tanggal_waktu_keberangkatan", r.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? "");

        cmd.Parameters.AddWithValue("@kelas_layanan", r.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@layanan_lainnya", r.Cells["Column_Layanan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@hotel_pilihan", r.Cells["Column_Nama_Hotel"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@penerbangan_pilihan", r.Cells["Column_Penerbangan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@informasi_tambahan", r.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? "");
        cmd.Parameters.AddWithValue("@harga_tiket", r.Cells["Total_Harga"].Value);

        // menambahkan nilai parameter sesuai nilai

        cmd.ExecuteNonQuery();
        // di perintah di eksekusi

        MessageBox.Show("Baris berhasil dimasukkan ke tabel tiket.", "Succes", MessageBoxButtons.OK, MessageBoxIcon.Information); // messagebox berhasil
    }

    label_total_harga.Text = $"{Convert.ToInt32(label_total_harga.Text) -
    Convert.ToInt32(dataGridView_ticket.CurrentRow.Cells["Total_Harga"].Value)}"; // mengurangi nilai total harga tiket
    dataGridView_ticket.Rows.RemoveAt(dataGridView_ticket.CurrentRow.Index); // menghapus baris tiket di datagridview
    lbl_informasi.Text = ""; // membersihkan label informasi selengkapnya
}
```

5. Hapus Tiket Yang Dipesan

```
● ● ●

private void button_hapus_tiket_Click(object sender, EventArgs e)
{
    if (dataGridView_ticket.CurrentRow != null) // jika pengguna tidak memilih baris dari dataGridView_ticket
    {
        int total_harga = Convert.ToInt32(label_total_harga.Text); // isi var total harga dengan label_total_harga.Text yang diubah menjadi int
        total_harga -= Convert.ToInt32(dataGridView_ticket.CurrentRow.Cells["Total_Harga"].Value); // var total harga lalu di kurang nilai dari sel Total_Harga pada dataGridView_ticket row saat ini yang sudah diubah ke int
        label_total_harga.Text = $"{total_harga}"; // set label_total_harga.Text ke label_total_harga.Text
        dataGridView_ticket.Rows.RemoveAt(dataGridView_ticket.CurrentRow.Index); // menghapus baris yang dipilih
        lbl_informasi.Text = ""; // mengosongkan lbl_informasi.Text

    }
    else
    {
        MessageBox.Show("Pilih baris tiket yang ingin dihapus.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // message box error di tampilkan
    }
}
```

6. Verifikasi Tiket

```
private void pictureBox2_Click(object sender, EventArgs e) // ketika gambar tombolverifikasi diklik
{
    if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text)) // jika textBox_tiket_id.Text kosong
    {
        MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // msgbox error
        textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
        lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
        return; // berhentikan program
    }
    if (double.TryParse(textBox_tiket_id.Text, out double id)) // jika textBox_tiket_id.Text adalah double masukkan ke var id
    {
        string koneksi = "server=localhost;user id=root;password=;database=katasii"; // server, userid, password, dan nama database
        using (MySqlConnection conn = new MySqlConnection(koneksi)) // koneksi database
        {
            conn.Open(); // buka database

            string query = "SELECT * FROM tiket WHERE id = @id"; // perintah sql
            MySqlCommand cmd = new MySqlCommand(query, conn); // var command sql
            cmd.Parameters.AddWithValue("@id", id); // menambahkan parameter ke perintah sql

            MySqlDataReader rd = cmd.ExecuteReader(); // eksekusi cmd sql

            if (rd.Read()) // jika terbaca
            {
                // Tiket ditemukan - tampilkan ke label
                MessageBox.Show("Status: Tiket Aktif", "Sukses", MessageBoxButtons.OK, MessageBoxIcon.Information);

                lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
                lbl_informasi.Text += "# Tiket ID: ";
                lbl_informasi.Text += rd["id"].ToString() ?? ""; // isi dari kolom databases
                lbl_informasi.Text += "\nNm Informasi Pribadi";
                lbl_informasi.Text += "\n * Name Kontak: ";
                lbl_informasi.Text += rd["nama_kontak"].ToString() ?? ""; // isi dari kolom databases
                lbl_informasi.Text += "\n * Tanggal lahir: ";
                lbl.informasi.Text += rd["tanggal_lahir"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * E-mail: ";
                lbl.informasi.Text += rd["email"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * No. Telp: ";
                lbl.informasi.Text += rd["no_telp"].ToString() ?? ""; // isi dari kolom databases

                lbl_informasi.Text += "\n\n# Detail Perjalanan";
                lbl_informasi.Text += "\n * Negara Keberangkatan: ";
                lbl_informasi.Text += rd["negara_keberangkatan"].ToString() ?? ""; // isi dari kolom databases
                lbl_informasi.Text += "\n * Negara Tujuan: ";
                lbl.informasi.Text += rd["negara_tujuan"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
                lbl.informasi.Text += rd["tanggal_waktu_keberangkatan"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * Kelas Layanan: ";
                lbl.informasi.Text += rd["kelas_layanan"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * Layanan Lainnya: ";
                lbl.informasi.Text += rd["layanan_lainnya"].ToString() ?? ""; // isi dari kolom databases

                lbl_informasi.Text += "\n\n# Rincian Perjalanan";
                if (!string.IsNullOrEmpty(rd["hotel_pilihan"].ToString())) // jika kolom hotel_pilihan dari database kosong
                {
                    lbl_informasi.Text += "\n * Hotel Pilihan: ";
                    lbl_informasi.Text += rd["hotel_pilihan"].ToString() ?? ""; // isi dari kolom databases
                }
                lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
                lbl.informasi.Text += rd["penerbangan_pilihan"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * Informasi Tambahan: ";
                lbl.informasi.Text += rd["informasi_tambahan"].ToString() ?? ""; // isi dari kolom databases
                lbl.informasi.Text += "\n * Harga Tiket: ";
                lbl.informasi.Text += "Rp. " + FormatMoney(Convert.ToInt32(rd["harga_tiket"])); // isi dari kolom databases lalu
format ke money

            }
            else
            {
                MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
                textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
            }
        }
    }
    else
    {
        MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
        return; // berhentikan program
    }
}
```

7. Pengisian Nama Tiket dan Nama Penerbangan

```
● ● ●

string[] namaTiket = new string[] { "Ekonomi", "Ekonomi Premium", "Bisnis", "Kelas Utama (First Class)" }; // menginisialisasi nama dari tiket
int[] hargaTiket = new int[] { 500000, 1000000, 1500000, 2000000 }; // menginisialisasi harga dari tiket
string[] namaPenerbangan = new string[] { "Citilink", "Lion Air", "Batik Air", "AirAsia" };

private void Form1_Load(object sender, EventArgs e) // ketika form di load
{
    for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di array naamaa tiket
    {
        comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); // menambahkan pilihan ke comboBox_kelas_layanan.Items dengan array nama dan harga tiket index (i)
    }
    if(namaPenerbangan.Length > 0) // jika array namaPenerbangan item nya lebih dari 0
    {
        int totalnamaPenerbangan = 0; // set jumlah awal 0
        do // lakukan se kali
        {
            comboBox_penerbangan.Items.Add($"{namaPenerbangan[totalnamaPenerbangan]}"); // tambahkan pilihan di comboBox_penerbangan.Items sesuai index dari array namaPenerbangan
            totalnamaPenerbangan++; // tambah 1
        }
        while (totalnamaPenerbangan < namaPenerbangan.Length); // jika masih kurang dari jumlah nama penerbangan ulangi lagi
    }
}
```

B. Fungsi Masing-masing Bagian

1. Halaman Menu

Halaman menu berfungsi untuk memberikan pilihan menu yang akan digunakan. Pada halaman menu ini, pengguna dapat memilih, apakah ingin membuat tiket baru, melihat daftar tiket aktif, melihat daftar tiket mati, verifikasi tiket, dan keluar dari aplikasi.

2. Buat Tiket Baru

Bagian buat tiket baru berfungsi untuk membuat tiket baru dengan formulir yang sudah tersedia dan akan memberikan koreksi otomatis oleh sistem. Tiket baru yang ditambahkan dapat dijadikan tiket aktif atau dihapus, akumulasi tiket juga akan terhitung sehingga menampilkan total harga dari tiket yang ditambahkan, serta penghitungan kembalian pelanggan. Selain itu, juga dapat untuk menampilkan informasi tambahan dari tiket yang dipesan.

3. Daftar Tiket Aktif

Bagian daftar tiket aktif berfungsi untuk menampilkan daftar tiket yang ada pada tabel tiket basis data MySql *localhost*. Menonaktifkan tiket, menghapus tiket, melihat informasi tambahan tiket aktif, serta mencetak tiket.

4. Daftar Tiket Mati

Bagian daftar tiket mati berfungsi untuk menampilkan daftar tiket yang ada pada tabel tiket mati basis data MySql *localhost*. Mengaktifkan kembali tiket, menghapus tiket, melihat informasi tambahan tiket mati.

5. Verifikasi Tiket

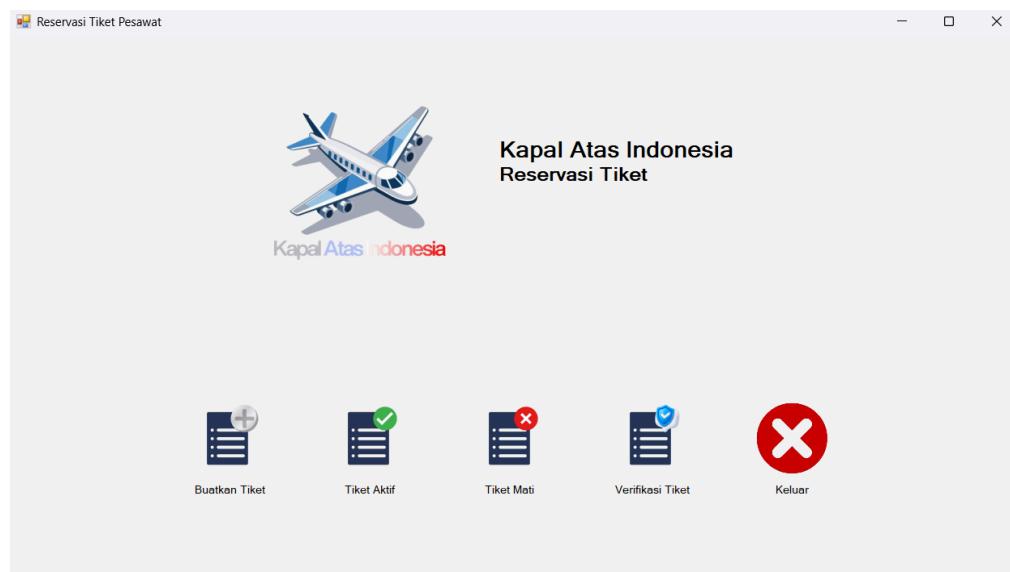
Bagian verifikasi tiket berfungsi untuk memeriksa status tiket berdasarkan identitas tiket. Tiket yang valid adalah tiket yang ada dalam tabel tiket aktif basis data *MySQL localhost* dan akan langsung menampilkan informasi lengkap dari tiket, serta tiket yang valid dapat dimatikan atau dihapus.

C. Hasil dan Pembahasan

Aplikasi ini berhasil dijalankan dengan baik tanpa kendala, fitur yang dirancang sebelumnya telah sepenuhnya terwujud yang dapat digunakan. MySQL juga berhasil dihubungkan dengan *localhost* yang menyimpan data tiket secara permanen. Namun, apabila *localhost* tidak terhubung ke MySQL, lebih tepatnya basis data “katasi”, fitur tiket aktif, tiket mati, dan verifikasi tiket tidak dapat diakses.

D. Tampilan Output Program

1. Halaman Menu



2. Buat Tiket Baru

Formulir Tiket

Kapal Atas Indonesia
Reservasi Tiket

Informasi Pribadi	Informasi Selengkapnya
Nama Kontak	Nama Kontak
Alfa Rizqi	Tanggal Lahir
Wednesday, October 20, 2010	E-mail
rizqi258745@student.smkn1kandeman.sch.id	
Nomor Negara	
+62 - Indonesia	
Nomor Telepon	
124512	

Detail Perjalanan	# Detail Perjalanan
Negara Keberangkatan	- Negara Keberangkatan: Indonesia
Indonesia	- Negara Tujuan: United States of America
Negara Tujuan	- Tanggal dan Waktu Keberangkatan: Thursday, November 27, 2025, 23:00
United States of America	- Kelas Layanan: Kelas Utama (First Class)
Tujuan	- Layanan Lainnya: Tiket Maskapai Penerbangan, Rental Mobil, Akomodasi Hotel
Tanggal dan Waktu Keberangkatan	
Thursday, November 27, 2025, 23:00	
Kelas layanan	
Kelas Utama (First Class) Rp. 2.000.000	
Silakan centang layanan yang pelanggan butuhkan	
<input checked="" type="checkbox"/> Tiket Maskapai Penerbangan	
<input checked="" type="checkbox"/> Rental Mobil	
<input checked="" type="checkbox"/> Akomodasi Hotel	

Rincian Perjalanan	# Rincian Perjalanan
Nama hotel pilihan	- Hotel Pilihan: Los Angles Hotel
Los Angles Hotel	- Penerbangan Pilihan: Lion Air
Penerbangan pilihan	- Informasi Tambahan: Tidak ada
Lion Air	- Harga Tiket: Rp. 2.000.000
Informasi tambahan	
Tidak ada	

Jadikan Tiket Aktif Hapus

Bantuan Kasir

Jumlah Rp. 2.000.000
harga tiket

Jumlah uang 3000000
pelanggan

Jumlah biaya 2500
admin

Kembalian Rp. 997500

Hitung

Lihat informasi

Menu

Tiket Aktif Tiket Mati Verifikasi Tiket Menu

3. Daftar Tiket Aktif

Daftar Tiket Aktif

ID	Nama Kontak	Tanggal Lahir	E-mail	Nomor Telepon	Negara Keberangkatan
40	Alfa Rizqi	Wednesday, Oct... rizqi258745@st... +62 - Indonesia,... Indonesia			

Informasi Selengkapnya
Tiket ID: 40
Informasi Pribadi
- Nama Kontak: Alfa Rizqi
- Tanggal lahir: Wednesday, October 20, 2010
- E-mail: rizqi258745@student.smkn1kandeman.sch.id
- No. Telp: +62 - Indonesia, 124512
Detail Perjalanan
- Negara Keberangkatan: Indonesia
- Negara Tujuan: United States of America
- Tanggal dan Waktu Keberangkatan: Thursday, November 27, 2025, 23:00
- Kelas Layanan: Kelas Utama (First Class)
- Layanan Lainnya: Tiket Maskapai Penerbangan, Rental Mobil, Akomodasi Hotel
Rincian Perjalanan
- Hotel Pilihan: Los Angles Hotel
- Penerbangan Pilihan: Lion Air
- Informasi Tambahan: Tidak ada
- Harga Tiket: Rp. 2.000.000

Menu

Refresh Nonaktifkan Hapus Tiket Mati Tiket Baru Baca Selengkapnya Cetak Menu

4. Daftar Tiket Mati

The screenshot shows a Windows application window titled "Daftar Tiket Mati". On the left is a table with columns: ID, Name Kontak, Tanggal Lahir, E-mail, Nomor Telepon, Negara Keberangkatan, and Negara Tujuan. The table contains two rows of data. Row 32 is for "Alif Alfian" and row 40 is for "Alfa Rizqi". On the right side, there is a panel titled "Informasi Selengkapnya" which displays detailed travel information for ticket ID 40. This includes the passenger's details, travel dates, destinations, and a breakdown of services like accommodation and rental cars. At the bottom of the window is a toolbar with icons for Refresh, Aktifkan, Hapus, Tiket Aktif, Tiket Baru, and Menu.

5. Verifikasi Tiket

The screenshot shows a Windows application window titled "Verifikasi Tiket". On the left, there is a search interface with a logo for "Kapal Atas Indonesia Reservasi Tiket" and a search bar labeled "Masukkan ID Tiket" containing the value "40". Below the search bar is a button labeled "Periksa" with a magnifying glass icon. On the right, there is a panel titled "Informasi Selengkapnya" showing the same detailed travel information as the previous window for ticket ID 40. At the bottom of the window is a toolbar with icons for Nonaktifkan, Hapus, Tiket Mati, Tiket Aktif, Tiket Baru, and Menu.

E. Hasil Uji Coba

Sebagai penyempurnaan, dilakukan beberapa uji coba antara lain:

1. Uji Coba Auto Koreksi

Uji coba dilakukan dengan sengaja tidak mengisi semua formulir dan penulisan email yang salah pada pembuatan tiket baru. Uji coba ini berhasil dan sistem berjalan dengan baik.

The screenshot shows the 'Formulir Tiket' application window. On the left, there's a form for 'Informasi Pribadi' with fields for Name Contact (Alfa), Birth Date (Tuesday, November 25, 2025), Email (asdad), Country (Indonesia), and Phone Number. Below it is the 'Detail Perjalanan' section with fields for Country of Origin, Destination, Date (Tuesday, November 25, 2025), Time, and Travel Class. Underneath is the 'Rincian Perjalanan' section with fields for Hotel and Flight Selection, and Additional Information. At the bottom are 'Kirim Formulir' and 'Bersihkan Formulir' buttons. In the center, a modal dialog box titled 'Error' displays the message 'Email tidak valid' (Email is invalid) with an 'OK' button. To the right, there's a 'Informasi Selengkapnya' panel with a table for contact information and a 'Bantuan Kasir' section with cashier details. At the bottom right are navigation buttons for 'Tiket Aktif', 'Tiket Mati', 'Verifikasi Tiket', and 'Menu'. A status bar at the bottom shows 'Lihat informasi'.

2. Uji Coba Keselarasan Daftar Tiket Aktif dan Daftar Tiket Mati

Uji coba dilakukan dengan membandingkan keselarasan antara baris yang ada pada tabel basis data MySQL *localhost* dengan *data grid view* daftar tiket aktif dan daftar tiket mati. Uji coba ini berhasil dan sistem berjalan dengan baik.

a. Keselarasan daftar tiket aktif

ID	Nama Kontak	Tanggal Lahir	E-mail	Nomor Telepon	Negara Keberangkatan
40	Alfa Rizqi	Wednesday, Oc...	rizqi258745@student.smkn1kandeman.sch.id	+62 - Indonesia	Indonesia

	id	nama_kontak	tanggal_lahir	email	no_telp	negara_keberangkatan	negara_tujuan	tanggal_waktu_keberangkatan
<input type="checkbox"/>	40	Alfa Rizqi	Wednesday, October 20, 2010	rizqi258745@student.smkn1kandeman.sch.id	+62 - 124512	Indonesia	United States of America	Thursday, November 27, 2025, 23:00

b. Keselarasan daftar tiket mati

ID	Nama Kontak	Tanggal Lahir	E-mail	Nomor Telepon	Negara Keberangkatan
32	Alif Alifan	Saturday, May 8...	alifkolip@gmail....	+62 - Indonesia,...	Indonesia

id	nama_kontak	tanggal_lahir	email	no_telp	negara_keberangkatan	negara_tujuan	tanggal_waktu_keberangkatan	kelas_layanan	layanan_lainnya	hotel_pilihan	pen
32	Alif Alifan	Saturday, May 8, 2010	alifkolip@gmail.com	+62 - 8562391928	Indonesia	Indonesia	Kamboja	Saturday, November 29, 2025, 04:00	Kelas Utama (First Class)	, Akomodasi Hotel	Alfa Hotel **** Garn

3. Penyesuaian Tiket yang Dicetak

Uji coba dilakukan dengan memasukkan *e-mail* yang panjang pada formulir pembuatan, diharapkan tulisan dapat menyesuaikan dengan ruang yang sempit. Uji coba ini berhasil dan sistem berjalan dengan lancar.

TIKET PESAWAT - KAPAL ATAS INDONESIA		
NAMA Alfa Rizqi	WAKTU MENGUDARA Thursday, November 27, 2025, 23:00	PENERBANGAN Lion Air
E-MAIL rizqi258745@student.smkn1kandeman.sch.id	DARI Indonesia	TIKET ID 40
	MENUJU United States of America	KELAS Kelas Utama (First Class)

BAB III

LAMPIRAN PROGRAM

A. Kode Lengkap

1. FormMenu.cs

```
● ● ●

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 Resevasi_Tiket_Pesawat
{
    public partial class FormMenu : Form
    {
        public FormMenu()
        {
            InitializeComponent();
        }

        private void pictureBox1_Click(object sender, EventArgs e) // jika gambar formulir baru diklik.
        {
            FormFormulir ff = new FormFormulir();
            this.Hide(); // tutup form saat ini
            ff.Show(); // tampilkan form daftar tiket aktif
        }

        private void pictureBox2_Click_1(object sender, EventArgs e) // jika gambar daftar tiket aktif diklik.
        {
            FormTiketAktif fta = new FormTiketAktif();
            this.Hide(); // tutup form saat ini
            fta.Show(); // tampilkan form daftar tiket aktif
        }

        private void pictureBox3_Click(object sender, EventArgs e) // jika gambar daftar tiket mati diklik.
        {
            FormTiketMati fm = new FormTiketMati();
            this.Hide(); // tutup form saat ini
            fm.Show(); // tampilkan form daftar tiket mati
        }

        private void pictureBox4_Click(object sender, EventArgs e) // jika gambar verifikasi tiket diklik.
        {
            FormVerifikasi fv = new FormVerifikasi();
            this.Hide(); // tutup form saat ini
            fv.Show(); // tampilkan form verifikasi tiket
        }

        private void pictureBox5_Click(object sender, EventArgs e) // jika gambar close aplikasi diklik.
        {
            if (MessageBox.Show("Keluar dari Aplikasi?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
            {
                Close();
            }
        }

        private void label5_Click(object sender, EventArgs e)
        {

        }

        private void FormMenu_Load(object sender, EventArgs e)
        {

        }
    }
}
```

2. FormFormulir.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Text.RegularExpressions;
using System.Threading;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient; // tambahan NuGet Packages, untuk include MySQL

namespace Reservasi_Tiket_Pesawat
{
    public partial class FormFormulir : Form
    {
        string[] namaTiket = new string[] { "Ekonomi", "Ekonomi Premium", "Bisnis", "Kelas Utama (First Class)" }; // menginisialisasi
        // nama dari tiket
        int[] hargaTiket = new int[] { 500000, 1000000, 1500000, 2000000 }; // menginisialisasi harga dari tiket
        string[] namaPenerbangan = new string[] { "Citilink", "Lion Air", "Batik Air", "AirAsia" };

        public FormFormulir()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e) // ketika form di load
        {
            for (int i = 0; i < namaTiket.Length; i++) // melakukan pengulangan sebanyak total item di array naamaa tiket
            {
                comboBox_kelas_layanan.Items.Add($"{namaTiket[i]} | Rp. {FormatMoney(hargaTiket[i])}"); // menambahkan pilihan ke
                comboBox_kelas_layanan.Items // dengan array nama dan harga tiket index (i)
            }
            if (namaPenerbangan.Length > 0) // jika array namaPenerbangan item nya lebih dari 0
            {
                int totalnamaPenerbangan = 0; // set jumlah awal 0
                do // lakukan sekali
                {
                    comboBox_penerbangan.Items.Add($"{namaPenerbangan[totalnamaPenerbangan]}"); // tambahkan pilihan di
                    comboBox_penerbangan.Items // sesuai index dari array namaPenerbangan
                    totalnamaPenerbangan++; // tambah 1
                }
                while (totalnamaPenerbangan < namaPenerbangan.Length); // jika masih kurang dari jumlah nama penerbangan ulangi lagi
            }
        }

        public static string FormatRupiah(decimal angka) // fungsi format angka ke rupiah
        {
            return angka.ToString("Rp.", new System.Globalization.CultureInfo("id-ID")); // menyesuaikan dengan mata uang indonesia
        }

        public static string FormatMoney(decimal angka) // fungsi dari angka ke digit uang
        {
            return angka.ToString("#,0").Replace(",", "."); // mengganti nilai menjadi ada koma dari setiap 3 digit lalu diubah lagi
            // menjadi titik.
        }

        private void textBox3_TextChanged(object sender, EventArgs e)
        {
            if (!checkBox_akomodasi_hotel.Checked && !string.IsNullOrEmpty(textBox_nama_hotel_pilihan.Text)) // memeriksa apakah
            pengguna memilih akomodasi hotel
            {
                textBox_nama_hotel_pilihan.Clear(); // membersihkan textBox_nama_hotel_pilihan
                MessageBox.Show("Anda tidak memilih layanan Akomodasi Hotel", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                // message box error ditampilkan
            }
        }

        private bool IsValidEmail(string email) // fungsi untuk mendeteksi apakah suatu string adalah email yang valid
        {
            if (string.IsNullOrWhiteSpace(email)) // jika email kosong
                return false;

            // Pola regex sederhana dan umum digunakan
            string pattern = @"^([a-zA-Z0-9_\.-]+@[a-zA-Z0-9\.-]+\.[a-zA-Z]{2,})$"; // mendeteksi apakah ada char '@' dan char '.'

            return Regex.IsMatch(email, pattern); // jika ada lanjut return, nilai true
        }

        private void btn_kirim_formulir_Click(object sender, EventArgs e) // jika tombol kirim di klik
        {
            bool invalid = false; // variable bool, untuk menjadi saklar, apakah invalid atau tidak

            string nama = textBox_nama.Text; // variable nama diset ke textBox_nama.Text

```

```

string ttl = dateTime_ttl.Text; // variable ttl diset ke dateTime_ttl.Text
string email = textBox_email.Text; // variable email diset ke textBox_email.Text
string nomor_negara = combo_nomor_negara.SelectedItem.ToString(); // mengeset variable nomor_negara ke
combo_nomor_negara.SelectedItem yang diubah dulu menjadi string.
string negara_keberangkatan = textBox_keberangkatan.Text; // mengeset var negara_keberangkatan ke textBox_keberangkatan.Text
string negara_tujuan = textBox_tujuan.Text; // mengeset var negara_tujuan ke textBox_tujuan.Text
string tanggal_berangkat = $"{dateTime_keberangkatan.Text}, {maskedTextBox_waktu_berangkat.Text}"; // var
tanggal_berangkat di set menjadi gabungan dateTime_keberangkatan.Text dengan maskedTextBox_waktu_berangkat.Text
string input_berangkat = maskedTextBox_waktu_berangkat.Text; // var input_berangkat diset ke
maskedTextBox_waktu_berangkat.Text
string kelas_layanan = ""; // membuat var kelas_layanan dengan tidak ada isinya
string layanan = ""; // var layanan dengan isi kosong
string hotel_pilihan = textBox_nama_hotel_pilihan.Text; // set var hotel_pilihan ke textBox_nama_hotel_pilihan.Text
string penerbangan_pilihan = comboBox_penerbangan.Text; // set var penerbangan_pilihan ke comboBox_penerbangan.Text
string informasi_tambahan = textBox_informasi_tambahan.Text; //set var informasi_tambahan ke
textBox_informasi_tambahan.Text

if (string.IsNullOrWhiteSpace(nama)) // mendeteksi apakah variable "nama" kosong
{
    MessageBox.Show("Nama kontak tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid menjadi true
    textBox_nama.Focus(); // focus ke textBox_nama
    return; // berhentikan program
}
if (!IsValidEmail(email)) // mendeteksi apakah variable "mail" adalah email yang valid
{
    MessageBox.Show("Email tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; // invalid
menjadi true
    textBox_email.Focus(); // fokus ke textBox_email
    return; // berhentikan program
}
if (combo_nomor_negara.SelectedIndex == -1) // jika pengguna tidak memilih combo_nomor_negara
{
    MessageBox.Show("Nomor negara tidak dipilih", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; //
invalid menjadi true
    combo_nomor_negara.Focus(); // fokus ke combo_nomor_negara
    return; // berhentikan program
}
if (double.TryParse(textBox_nomor_telepon.Text, out double nomor_telepon)) // mendeteksi kalau textBox_nomor_telepon.Text
adalah angka, dan dimasukkan ke var nomor_telepon
{
    nomor_negara += $", {nomor_telepon}"; // menggabungkan nomor_negara dengan nomor telepon
}
else // jika textBox_nomor_telepon.Text tidak angka
{
    MessageBox.Show("Nomor telepon tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; //
invalid menjadi true
    textBox_nomor_telepon.Focus(); // fokus ke textBox_nomor_telepon
    return; // berhentikan program
}
if (string.IsNullOrWhiteSpace(negara_keberangkatan)) // jika var negara_keberangkatan kosong
{
    MessageBox.Show("Negara keberangkatan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid =
true; // invalid menjadi true
    textBox_keberangkatan.Focus(); // fokus ke textBox_keberangkatan
    return; // berhentikan program
}
if (string.IsNullOrWhiteSpace(negara_tujuan)) // jika negara_tujuan kosong
{
    MessageBox.Show("Negara tujuan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true; //
invalid menjadi true
    textBox_tujuan.Focus(); // fokus ke textBox_tujuan
    return; // berhentikan program
}
// Pastikan user sudah isi lengkap (misal "14:30")
if (!maskedTextBox_waktu_berangkat.MaskCompleted) // jika maskedTextBox_waktu_berangkat tidak diisi penuh
{
    MessageBox.Show("Input waktu berangkat belum lengkap! Format: HH:mm",
                    "Validasi Waktu", MessageBoxButtons.OK, MessageBoxIcon.Warning); // message box warning ditampilkan
    maskedTextBox_waktu_berangkat.Focus(); // fokus ke maskedTextBox_waktu_berangkat
    invalid = true; // invalid menjadi true
    return; // berhentikan program
}
// Coba parse input jadi DateTime
if (DateTime.TryParseExact(input_berangkat, "HH:mm",
                           System.Globalization.CultureInfo.InvariantCulture,
                           System.Globalization.DateTimeStyles.None,
                           out DateTime waktu))
{
}
else
{
    // ✗ Jika tidak valid
    MessageBox.Show("Waktu berangkat tidak valid! Gunakan format HH:mm (contoh: 14:30)",
                    "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // message box warning ditampilkan
    maskedTextBox_waktu_berangkat.Focus(); // fokus ke maskedTextBox_waktu_berangkat
    invalid = true; // invalid menjadi true
    return; // berhentikan program
}
if (comboBox_kelas_layanan.SelectedIndex == -1) // jika comboBox_kelas_layanan tidak dipilih
{
    MessageBox.Show("Kelas layanan tidak dipilih", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid = true;
}
// invalid menjadi true

```

```

        comboBox_kelas_layanan.Focus(); // fokus ke comboBox_kelas_layanan
        return; // program dihentikan
    }
    switch (comboBox_kelas_layanan.SelectedIndex) // membuat pilihan dari comboBox_kelas_layanan.SelectedIndex
    {
        case 0:
            kelas_layanan = "Ekonomi"; // kelas layanan di ubah
            break;
        case 1:
            kelas_layanan = "Ekonomi Premium"; // kelas layanan di ubah
            break;
        case 2:
            kelas_layanan = "Bisnis"; // kelas layanan di ubah
            break;
        case 3:
            kelas_layanan = "Kelas Utama (First Class)"; // kelas layanan di ubah
            break;
        default:
            break;
    }
    if(checkBox_tiket_maskapai.Checked) // jika checkBox_tiket_maskapai dipilih
    {
        layanan += "Tiket Maskapai Penerbangan"; // menambahkan pilihan ke var layanan
    }
    if (checkBox_rent_car.Checked) // jika checkBox_rent_car dipilih
    {
        layanan += ", Rental Mobil"; // menambahkan pilihan ke var layanan
    }
    if (checkBox_akomodasi_hotel.Checked) // jika checkBox_akomodasi_hotel dipilih
    {
        layanan += ", Akomodasi Hotel"; // menambahkan pilihan ke var layanan
    }
    if (string.IsNullOrWhiteSpace(hotel_pilihan) && checkBox_akomodasi_hotel.Checked) // jika var hotel_pilihan kosong namun
    checkBox_akomodasi_hotel dipilih
    {
        MessageBox.Show("Nama hotel untuk akomodasi tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        invalid = true; // invalid menjadi true
        textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
        return; // berhentikan program
    }
    if (!string.IsNullOrWhiteSpace(hotel_pilihan) && !checkBox_akomodasi_hotel.Checked) // jika var hotel_pilihan tidak kosong
    namun checkBox_akomodasi_hotel tidak dipilih
    {
        textBox_nama_hotel_pilihan.Focus(); // fokus ke textBox_nama_hotel_pilihan
        MessageBox.Show("Anda tidak mengcentang layanan Akomodasi Hotel", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        invalid = true; // invalid menjadi true
        return; // berhentikan program
    }
    if (string.IsNullOrWhiteSpace(penerbangan_pilihan)) // jika var penerbangan_pilihan kosong
    {
        MessageBox.Show("Penerbangan pilihan tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); invalid =
        true; // invalid menjadi true
        comboBox_penerbangan.Focus(); // fokus ke comboBox_penerbangan;
        return; // berhentikan program
    }
    int harga = 0; // mendefinisikan awal harga menjadi 0
    int total_harga = Convert.ToInt32(label_total_harga.Text); // mengisi var total_harga menjadi nilai dari label total harga
    yang diconvert ke int
    // tiket default (ekonomi, ekonomi premium, bisnis, kelas atas)
    if (comboBox_kelas_layanan.SelectedIndex == 0) // jika comboBox_kelas_layanan dipilih index 0
    {
        harga = hargaTiket[0]; // var harga diubah menyesuaikan dengan array hargaTiket index 0
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 1) // jika tidak dan comboBox_kelas_layanan dipilih index 1
    {
        harga = hargaTiket[1]; // var harga diubah menyesuaikan dengan array hargaTiket index 1
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 2) // jika tidak dan comboBox_kelas_layanan dipilih index 2
    {
        harga = hargaTiket[2]; // var harga diubah menyesuaikan dengan array hargaTiket index 2
    }
    else if (comboBox_kelas_layanan.SelectedIndex == 3) // jika tidak dan comboBox_kelas_layanan dipilih index 3
    {
        harga = hargaTiket[3]; // var harga diubah menyesuaikan dengan array hargaTiket index 3
    }
    else // tiket tambahan
    {
        harga = hargaTiket[comboBox_kelas_layanan.SelectedIndex]; // var harga diubah menyesuaikan dengan array hargaTiket
        index sesuai index pilihan
    }
    if (invalid == false) // jika nilai invalid tidak true
    {
        dataGrid_ticket.Rows.Add(nama, ttl, email, nomor_negara, negara_keberangkatan, negara_tujuan, tanggal_berangkat,
        kelas_layanan, layanan, hotel_pilihan, penerbangan_pilihan, informasi_tambahan, harga); // tambahkan var tadi ke dataGrid_ticket.Rows
        dengan urutan yang pas
        total_harga += harga; // pengisian dan penambahan nilai var total_harga dengan var harga
        label_total_harga.Text = $"{total_harga}"; // mengubah label_total_harga.Text sesuai dengan harga
    }
}

```

```

private void button_clear_form_Click(object sender, EventArgs e)
{
    //membersihkan text, pilihan, dan status ceklis pada semua input.
    textBox_nama.Clear();
    dateTime_ttl.ResetText();
    textBox_email.Clear();
    textBox_nomor_telepon.Clear();
    textBox_keberangkatan.Clear();
    textBox_tujuan.Clear();
    dateTime_keberangkatan.ResetText();
    maskedTextBox_waktu_berangkat.ResetText();
    comboBox_kelas_layanan.ResetText();
    checkBox_tiket_maskapai.Checked = false;
    checkBox_rent_car.Checked = false;
    checkBox_akkomodasi_hotel.Checked = false;
    textBox_nama_hotel_pilihan.Clear();
    comboBox_penerbangan.ResetText();
    textBox_informasi_tambahan.Clear();
    comboBox_kelas_layanan.SelectedIndex = -1;
    comboBox_nomor_negara.SelectedIndex = -1;
    comboBox_penerbangan.SelectedIndex = -1;
}

private void button_hapus_tiket_Click(object sender, EventArgs e)
{
    if (dataGridView_ticket.CurrentRow != null) // jika pengguna tidak memilih baris dari dataGridView_ticket
    {
        int total_harga = Convert.ToInt32(label_total_harga.Text); // isi var total harga dengan label_total_harga.Text yang diubah menjadi int
        total_harga -= Convert.ToInt32(dataGridView_ticket.CurrentRow.Cells["Total_Harga"].Value); // var total harga lalu di kurang nilai dari sel Total_Harga pada dataGridView_ticket row saat int yang sudah diubah ke int
        label_total_harga.Text = $"{total_harga}"; // set label_total_harga.Text ke label_total_harga.Text
        dataGridView_ticket.Rows.RemoveAt(dataGridView_ticket.CurrentRow.Index); // menghapus baris yang dipilih
        lbl_informasi.Text = ""; // mengosongkan lbl_informasi.Text
        textBox_biaya_admin.Text = ""; // mengosongkan textbox biaya admin
        textBox_uang_pelanggan.Text = ""; // mengosongkan textbox uang pelanggan
        lbl_kembalian.Text = "Rp. 0"; // mengeset lbl kembalian text ke setelan awal (Rp. 0).
    }
    else
    {
        MessageBox.Show("Pilih baris tiket yang ingin dihapus.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // message box error di tampilkan
    }
}

private void button_print_Click(object sender, EventArgs e) // ketika button print di klik
{
    PrintPreviewDialog preview = new PrintPreviewDialog(); // membuat var preview
    preview.Document = printDocument1; // set ke preview dari printDocument1
    preview.ShowDialog(); // menunjukka preview
}

private void button1_Click(object sender, EventArgs e) // ketika tombol hitung kembalian di klik
{
    if (double.TryParse(textBox_uang_pelanggan.Text, out double uang_pelanggan)) // jika textBox_uang_pelanggan.Text adalah angka, dimasukkan ke var uang_pelanggan
    {
        double total = uang_pelanggan - Convert.ToInt32(label_total_harga.Text); // set var total (total kembalian) dengan uang pelanggan - label_total_harga.Text yang sudah diubah ke int
        if (!string.IsNullOrEmpty(textBox_biaya_admin.Text)) // jika textBox_biaya_admin.Text tidak kosong
        {
            if (double.TryParse(textBox_biaya_admin.Text, out double biaya_admin)) // jika textBox_biaya_admin.Text adalah angka, masukkan ke var biaya_admin
            {
                total -= biaya_admin; // total dikurang biaya admin
            }
            else MessageBox.Show("Biaya admin tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
        }
        lbl_kembalian.Text = $"Rp. {total}"; // menganti lbl_kembalian.Text dengan total
        if (total < 0) MessageBox.Show("Uang pelanggan tidak cukup", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
    }
    // jika textBox_uang_pelanggan.Text bukan angka
    else MessageBox.Show("Uang pelanggan tidak valid", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
}

private void button_informasi_tambahan_Click_1(object sender, EventArgs e) // ketika tombol informasi tambahan di klik
{
    if (dataGridView_ticket.CurrentRow != null) // jika row dari dataGridView_ticket tidak ada yang dipilih
    {
        lbl_informasi.Text = ""; // dataGridView_ticket diksangkan terlebih dahulu
    }
}

```

```

        lbl_informasi.Text += "# Informasi Pribadi";
        lbl_informasi.Text += "\n * Nama Kontak: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Tanggal lahir: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_TTL"].Value?.ToString() ?? ""; // ambil data dari
dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * E-mail: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString() ?? ""; // ambil data dari
dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * No. Telp: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nomor_Telp"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells

        lbl_informasi.Text += "\n\n# Detail Perjalanan";
        lbl_informasi.Text += "\n * Negara Keberangkatan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Negara Tujuan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Kelas Layanan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Layanan Lainnya: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Layanan"].Value?.ToString() ?? ""; // ambil data dari
dataGrid_ticket.CurrentRow.Cells

        lbl_informasi.Text += "\n\n# Rincian Perjalanan";
        if(!string.IsNullOrEmpty(dataGrid_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString()))
        {
            lbl_informasi.Text += "\n * Hotel Pilihan: ";
            lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        }
        lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Penerbangan"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Informasi Tambahan: ";
        lbl_informasi.Text += dataGrid_ticket.CurrentRow.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? ""; // ambil data
dari dataGrid_ticket.CurrentRow.Cells
        lbl_informasi.Text += "\n * Harga Tiket: ";
        lbl_informasi.Text += "Rp. " + FormatMoney(Convert.ToInt32(dataGrid_ticket.CurrentRow.Cells["Total_Harga"].Value)); // ambil data
dari dataGrid_ticket.CurrentRow.Cells, lalu di convert ke int dan diformat ke money,
        }
        else
        {
            MessageBox.Show("Pilih baris tiket yang ingin anda lihat informasi tambahannya", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error); // messagebox error ditampilkan
        }
    }

    private void printPreviewDialog1_Load(object sender, EventArgs e)
    {

    }

    private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
    {

    }

    private void button_aktifkan_tiket_Click(object sender, EventArgs e)
    {

        if (dataGrid_ticket.CurrentRow == null) // jika dataGrid_ticket tidak ada baris yang dipilih
        {
            MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox
error ditampilkan
            return; // program diberhentikan
        }

        string koneksi = "server=localhost;user id=root;password=;database=katas"; // menulis database mysql yang akan
dihubungkan

        using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
        {
            conn.Open(); // buka mysql

            string query = @"INSERT INTO tiket
(nama_kontak, tanggal_lahir, email, no_telp, negara_keberangkatan, negara_tujuan,
tanggal_waktu_keberangkatan, kelas_layanan, layanan_lainnya, hotel_pilihan,
penerbangan_pilihan, informasi_tambahan, harga_tiket)
VALUES
(@nama_kontak, @tanggal_lahir, @email, @no_telp, @negara_keberangkatan, @negara_tujuan,
@tanggal_waktu_keberangkatan, @kelas_layanan, @layanan_lainnya, @hotel_pilihan,
@penerbangan_pilihan, @informasi_tambahan, @harga_tiket)"; // perintah querry sql, (INSERT INTO, VALUES, dll.) untuk
----- di sini -----"

```

```

menambah baris baru sesuai parameter

MySqlCommand cmd = new MySqlCommand(query, conn); // perintah query

DataGridviewRow r = dataGrid_ticket.CurrentRow;
cmd.Parameters.AddWithValue("@nama_kontak", r.Cells["Column_Nama_Kontak"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@tanggal_lahir", r.Cells["Column_TTL"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@email", r.Cells["Column_Email"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@negara_keberangkatan", r.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@negara_tujuan", r.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@tanggal_waktu_keberangkatan", r.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@kelas_layanan", r.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@layanan_lainnya", r.Cells["Column_Layanan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@hotel_pilihan", r.Cells["Column_Nama_Hotel"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@penerbangan_pilihan", r.Cells["Column_Penerbangan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@informasi_tambahan", r.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? "");
cmd.Parameters.AddWithValue("@harga_tiket", r.Cells["Total_Harga"].Value);

// menambahkan nilai parameter sesuai nilai

cmd.ExecuteNonQuery();
// di perintah di eksekusi

MessageBox.Show("Baris berhasil dimasukkan ke tabel tiket.", "Succes", MessageBoxButtons.OK,
MessageBoxIcon.Information); // messagebox berhasil
}

label_total_harga.Text = $"{Convert.ToInt32(label_total_harga.Text) -
Convert.ToInt32(dataGrid_ticket.CurrentRow.Cells["Total_Harga"].Value)}"; // mengurangi nilai total harga tiket
dataGrid_ticket.Rows.RemoveAt(dataGrid_ticket.CurrentRow.Index); // menghapus baris tiket di datagridview
lbl_informasi.Text = ""; // membersihkan label informasi selengkapnya
textBox_biaya_admin.Text = ""; // mengosongkan textbox biaya admin
textBox_uang_pelanggan.Text = ""; // mengosongkan textbox uang pelanggan
lbl_kembalian.Text = "Rp. 0"; // mengeset lbl kembalian text ke setelan awal (Rp. 0).
}
}

private void pictureBox2_Click(object sender, EventArgs e)
{
    FormTiketAktif fta = new FormTiketAktif();
    this.Hide(); // tutup form saat ini
    fta.Show(); // tampilkan form daftar tiket aktif
}

private void pictureBox3_Click(object sender, EventArgs e)
{
    FormTiketMati fm = new FormTiketMati();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form daftar tiket mati
}

private void pictureBox4_Click(object sender, EventArgs e)
{
    FormVerifikasi fv = new FormVerifikasi();
    this.Hide(); // tutup form saat ini
    fv.Show(); // tampilkan form verifikasi tiket
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    FormMenu fm = new FormMenu();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form menu
}

private void label8_Click(object sender, EventArgs e)
{
}

private void label10_Click(object sender, EventArgs e)
{
}

private void label11_Click(object sender, EventArgs e)
{
}

private void label3_Click(object sender, EventArgs e)
{
}

private void label15_Click(object sender, EventArgs e)
{
}

```

```

private void dateTimePicker1_ValueChanged(object sender, EventArgs e)
{
}

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

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

private void dateTimePicker1_ValueChanged_1(object sender, EventArgs e)
{
}

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

private void textBox1_TextChanged(object sender, EventArgs e)
{
}

private void textBox4_TextChanged(object sender, EventArgs e)
{
}

private void groupBox2_Enter(object sender, EventArgs e)
{
}

private void textBox5_TextChanged(object sender, EventArgs e)
{
}

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

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

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

private void dataGridView_ticket_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}

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

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

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

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

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

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

```

```
private void lbl_keberangkatan_Click(object sender, EventArgs e)
{
}

private void dateTimePicker2_ValueChanged(object sender, EventArgs e)
{
}

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

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

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

private void label3_Click_1(object sender, EventArgs e)
{
}

private void label2_Click(object sender, EventArgs e)
{
}

private void label4_Click(object sender, EventArgs e)
{
}

private void maskedTextBox_waktu_berangkat_MaskInputRejected(object sender, MaskInputRejectedEventArgs e)
{
}

private void maskedTextBox_waktu_kembali_MaskInputRejected(object sender, MaskInputRejectedEventArgs e)
{
}

private void listBox13_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
}

private void label2_Click_1(object sender, EventArgs e)
{
}
}
```

3. FormTiketAktif.cs

```
● ● ●

using MySql.Data.MySqlClient; // tambahan NuGet Packages, untuk include MySQL
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;
using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Resevasi_Tiket_Pesawat
{
    public partial class FormTiketAktif : Form
    {
        public FormTiketAktif()
        {
            InitializeComponent();
        }

        private void FormTiketAktif_Load(object sender, EventArgs e)
        {
            TampilkanData(); // refresh datagrid
        }

        private void TampilkanData()
        {
            string koneksi = "server=localhost;user id=root;password=;database=katas"; // menulis database mysql yang akan
dihubungkan
            MySqlConnection conn = new MySqlConnection(koneksi); // hubungkan mysql
            conn.Open(); // buka mysql

            // 1. Hitung total baris dulu
            MySqlCommand countCmd = new MySqlCommand("SELECT COUNT(*) FROM tiket", conn);
            int total = Convert.ToInt32(countCmd.ExecuteScalar());

            // Set progress bar
            progressBar1.Value = 0;
            progressBar1.Maximum = total;
            progressBar1.Visible = true;

            // 2. Ambil data
            string query = "SELECT * FROM tiket"; // perintah untuk memilih baris di tabel tiket
            MySqlCommand cmd = new MySqlCommand(query, conn);
            MySqlDataReader rd = cmd.ExecuteReader();

            dataGridView_active_ticket.Rows.Clear(); // menghapus isi lama
            while (rd.Read())
            {
                // Tambahkan row baru
                dataGridView_active_ticket.Rows.Add(
                    rd["id"].ToString(),
                    rd["nama_kontak"].ToString(),
                    rd["tanggal_lahir"].ToString(),
                    rd["email"].ToString(),
                    rd["no_telp"].ToString(),
                    rd["negara_keberangkatan"].ToString(),
                    rd["negara_tujuan"].ToString(),
                    rd["tanggal_waktu_keberangkatan"].ToString(),
                    rd["kelas_layanan"].ToString(),
                    rd["layanan_lainnya"].ToString(),
                    rd["hotel_pilihan"].ToString(),
                    rd["penerbangan_pilihan"].ToString(),
                    rd["informasi_tambahan"].ToString(),
                    rd["harga_tiket"].ToString()
                );
            }

            if (progressBar1.Value < progressBar1.Maximum)
                progressBar1.Value++; // ambahkan value progress bar setiap memuat baris
        }

        rd.Close();
        conn.Close();

        // 4. Sembunyikan progress bar setelah selesai
        progressBar1.Visible = false;
    }

    private void button_refresh_Click(object sender, EventArgs e)
    {
        TampilkanData();
    }

    private void button_delete_Click(object sender, EventArgs e)
    {
        if (dataGridView_active_ticket.CurrentRow == null) // jika dataGridView_active_ticket tida ada baris yang dipilih
    }
}
```

```

        {
            MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
            return; // program dihentikan
        }

        string id = dataGrid_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString(); // var id di set ke nilai pada kolom Column_ID pada dataGrid_active_ticket.CurrentRow.Cells

        if (MessageBox.Show("Yakin hapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes) // show dialog messagebox warning lalu jika direspon yes
        {
            HapusData(id); // hapus tiket sesuai var id
            TampilkanData(); // Refresh datagrid
            lbl_informasi.Text = ""; // clear informasi selengkapnya
        }
    }

    private void HapusData(string id) // fungsi hapus tiket sesuai id
    {
        string koneksi = "server=localhost;user id=root;password=katas"; // menulis database mysql yang akan dihubungkan
        using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
        {
            conn.Open(); // buka mysql
            string query = "DELETE FROM tiket WHERE id = @id"; // perintah querry untuk hapus pada baris sesuai id
            MySqlCommand cmd = new MySqlCommand(query, conn); // membuat cmd menjadi perintah
            cmd.Parameters.AddWithValue("@id", id); // tambah parameter ID ke perintah
            cmd.ExecuteNonQuery(); // di perintah di eksekusi
        }
    }

    private void PindahkanBaris(string id) // fungsi untuk pindahkan baris ke tabel mati
    {
        string koneksi = "server=localhost;user id=root;password=katas"; // menulis database mysql yang akan dihubungkan
        using (MySqlConnection conn = new MySqlConnection(koneksi)) // hubungkan mysql
        {
            conn.Open(); // buka mysql

            // 1) Insert ke tabel tujuan
            string insertQuery = "INSERT INTO tiket_mati SELECT * FROM tiket WHERE id = @id"; // perintah querry sql, INSERT untuk menambah baris baru ke tabel_mati yang sama seperti baris yang ada di tabel tiket
            MySqlCommand cmdInsert = new MySqlCommand(insertQuery, conn); // membuat cmdinsert dari querry
            cmdInsert.Parameters.AddWithValue("@id", id); // penambahan parameter id baris
            cmdInsert.ExecuteNonQuery(); // perintah di eksekusi

            // 2) Hapus dari tabel_asal
            string deleteQuery = "DELETE FROM tiket WHERE id = @id"; // perintah querry dihapus baris sebelumnya dari tabel tiket
            MySqlCommand cmdDelete = new MySqlCommand(deleteQuery, conn); // cmddelete dibuat dari query
            cmdDelete.Parameters.AddWithValue("@id", id); // parameter id ditambahkan ke cmd query
            cmdDelete.ExecuteNonQuery(); // eksesuki cmd
        }
    }

    private void button_nonaktifkan_Click(object sender, EventArgs e) // fungsi untuk memindahkan tiket ke tiket mati
    {
        if (dataGrid_active_ticket.CurrentRow == null) // jika dataGrid_active_ticket tidak ada baris yang dipilih
        {
            MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            return; // program diberhentikan
        }

        string id = dataGrid_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString(); // definisikan var id sesuai dengan kolom Column_ID pada dataGrid_active_ticket.CurrentRow.Cells

        if (MessageBox.Show("Pindahkan data ini ke Daftar Tiket Mati?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes) // messagebox dialog warning, result jika yes
        {
            PindahkanBaris(id); // jalankan fungsi pindahkan baris
            TampilkanData(); // refreshdatagrid
            lbl_informasi.Text = ""; // clear informasi selengkapnya
        }
    }

    private void button_back_to_form_Click(object sender, EventArgs e)
    {
        FormFormulir ff = new FormFormulir();
        this.Hide(); // tutup form saat ini
        ff.Show(); // tampilkan form buat tiket
    }

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

    private void button_to_tiket_mati_Click(object sender, EventArgs e)
    {
        FormTiketMati fm = new FormTiketMati();
        this.Hide(); // tutup form saat ini
        fm.Show(); // tampilkan form daftar tiket mati
    }

```

```

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

private void printDocument1_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)
{
}

private void printPreviewDialog1_Load(object sender, EventArgs e)
{
}

private void printDocument1_PrintPage_1(object sender, System.Drawing.Printing.PrintPageEventArgs e) // saat printDocument1 dibuka halamannya
{
    if (dataGridView_active_ticket.CurrentRow != null) // jika dataGridView_active_ticket tidak ada baris yang dipilih
    {

        Graphics g = e.Graphics; // membuat graphics baru
        Font headerFont = new Font("Arial", 18, FontStyle.Bold); // membuat headerFont
        Font labelFont = new Font("Arial", 10, FontStyle.Bold); // membuat label font
        Font valueFont = new Font("Arial", 10); // membuat nilai font
        Brush blueBrush = new SolidBrush(Color.FromArgb(0, 102, 204)); // membuat brush biru
        Brush blackBrush = Brushes.Black; // membuat brush hitam

        int startX = 50; // set nilai x awal ke 50
        int startY = 60; // set nilai y awal ke 60
        int width = 750; // set panjang graphic ke 750
        int height = 250; // set lebar graphic ke 250

        // --- Background garis biru atas ---
        g.FillRectangle(blueBrush, startX, startY, width, 40); // buat persegi panjang biru
        g.DrawString("TIKET PESAWAT - KAPAL ATAS INDONESIA", headerFont, Brushes.White, startX + 15, startY + 5); // tulis teks

        // --- Garis border tiket ---
        g.DrawRectangle(Pens.Black, startX, startY, width, height); // garis lurus untuk border dengan warna hitam

        int leftColX = startX + 20; // align left
        int midColX = startX + 195; // align mid
        int rightColX = startX + 500; // align right
        int lineY = startY + 70; // garis ketinggian

        // --- Info penumpang ---
        g.DrawString("NAMA", labelFont, blueBrush, leftColX, lineY); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString(), valueFont, blackBrush, leftColX, lineY + 18); // ambil data dari kolom datagridview pada current row

        g.DrawString("E-MAIL", labelFont, blueBrush, leftColX, lineY + 50); // tulis teks
        string email = dataGridView_active_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString();
        int max = 20;
        if (email.Length > max)
        {
            for (int i = max; i < email.Length; i += max + 1)
            {
                email = email.Insert(i, "\n");
            }
        }
        g.DrawString(email, valueFont, blackBrush, leftColX, lineY + 68); // ambil data dari kolom datagridview pada current row

        // --- Info keberangkatan ---
        g.DrawString("WAKTU MENGUDARA", labelFont, blueBrush, midColX, lineY); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString(), valueFont, blackBrush, midColX, lineY + 18); // ambil data dari kolom datagridview pada current row

        g.DrawString("DARI", labelFont, blueBrush, midColX, lineY + 50); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_Negara_Keberangkatan"].Value?.ToString(), valueFont, blackBrush, midColX, lineY + 68); // ambil data dari kolom datagridview pada current row

        g.DrawString("MENUJU", labelFont, blueBrush, midColX, lineY + 100); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_Negara_Tujuan"].Value?.ToString(), valueFont, blackBrush, midColX, lineY + 118); // ambil data dari kolom datagridview pada current row

        // --- Info penerbangan ---
        g.DrawString("PENERBANGAN", labelFont, blueBrush, rightColX, lineY); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_Penerbangan"].Value?.ToString(), valueFont, blackBrush, rightColX, lineY + 18); // ambil data dari kolom datagridview pada current row

        g.DrawString("TIKET ID", labelFont, blueBrush, rightColX, lineY + 50); // tulis teks
        g.DrawString(dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value?.ToString(), valueFont, blackBrush, rightColX, lineY + 68); // ambil data dari kolom datagridview pada current row
    }
}

```

```

        g.DrawString("KELAS", labelFont, blueBrush, rightColX, lineY + 100); // tulis teks
        g.DrawString(dataGrid_active_ticket.CurrentRow.Cells["Column_Kelas_Layanan"].Value?.ToString(), valueFont, blackBrush,
rightColX, lineY + 118); // ambil data dari kolom datagridview pada current row
    }
    else
    {
        MessageBox.Show("Pilih baris tiket yang ingin anda cetak", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // messagebox error ditampilkan
    }
}

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

private void pictureBox2_Click(object sender, EventArgs e) // ketika gambar print di klik
{
    if (dataGrid_active_ticket.CurrentRow != null) // jika dataGrid_active_ticket.CurrentRow dipilih
    {
        PrintPreviewDialog preview = new PrintPreviewDialog(); // membuat var preview
        preview.Document = printDocument1; // set ke preview dari printDocument1
        preview.ShowDialog(); // menunjukkan preview
    }
    else // jika tidak ada row/baris yang dipilih.
    {
        MessageBox.Show("Pilih baris tiket yang ingin anda cetak", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
//messagebox error
    }
}

private void pictureBox1_Click(object sender, EventArgs e)
{
    if (dataGrid_active_ticket.CurrentRow != null)
    {
        lbl_informasi.Text = "";
        lbl_informasi.Text += "# Tiket ID: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_ID"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n\n# Informasi Pribadi";
        lbl_informasi.Text += "\n * Nama Kontak: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Tanggal lahir: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_TTL"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * E-mail: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * No. Telp: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Nomor_Telp"].Value?.ToString() ?? "";

        lbl_informasi.Text += "\n\n# Detail Perjalanan";
        lbl_informasi.Text += "\n * Negara Keberangkatan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Negara Tujuan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? "";

        lbl_informasi.Text += "\n * Kelas Layanan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Layanan Lainnya: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Layanan"].Value?.ToString() ?? "";

        lbl_informasi.Text += "\n\n# Rincian Perjalanan";
        if (!string.IsNullOrEmpty(dataGrid_active_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString()))
        {
            lbl_informasi.Text += "\n * Hotel Pilihan: ";
            lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString() ?? "";
        }
        lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Penerbangan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Informasi Tambahan: ";
        lbl_informasi.Text += dataGrid_active_ticket.CurrentRow.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Harga Tiket: ";
        lbl_informasi.Text += "Rp. " +
FormatMoney(Convert.ToInt32(dataGrid_active_ticket.CurrentRow.Cells["Total_Harga"].Value));
        }
        else
        {
            MessageBox.Show("Pilih baris tiket yang ingin anda lihat informasi tambahannya", "Error", MessageBoxButtons.OK,
MessageBoxIcon.Error);
        }
    }

    public static string FormatMoney(int angka)
    {
        return angka.ToString("#,0").Replace(".", ",");
    }

    private void pictureBox6_Click(object sender, EventArgs e)
    {
        TampilkanData(); //refresh
    }
}

```

```

private void pictureBox3_Click(object sender, EventArgs e) // pindahkan ke daftar mati
{
    if (dataGrid_active_ticket.CurrentRow == null)
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string id = dataGrid_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();

    if (MessageBox.Show("Pindahkan data ini ke Daftar Tiket Mati?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        PindahkanBaris(id);
        TampilkanData(); // refresh datagrid
        lbl_informasi.Text = ""; // clear informasi selengkapnya
    }
}

private void pictureBox4_Click(object sender, EventArgs e) // hapus tiket
{
    if (dataGrid_active_ticket.CurrentRow == null)
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string id = dataGrid_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();

    if (MessageBox.Show("Yakin hapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        HapusData(id);
        TampilkanData(); // Refresh datagrid
        lbl_informasi.Text = ""; // clear informasi selengkapnya
    }
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    FormTiketMati fm = new FormTiketMati();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form daftar tiket mati
}

private void pictureBox7_Click(object sender, EventArgs e)
{
    FormFormulir ff = new FormFormulir();
    this.Hide(); // tutup form saat ini
    ff.Show(); // tampilkan form buat tiket
}

private void pictureBox8_Click(object sender, EventArgs e)
{
    FormMenu fm = new FormMenu();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form menu
}
}

```

4. FormTiketMati.cs

```
● ● ●

using MySql.Data.MySqlClient; // tambahan NuGet Packages, untuk include MySQL
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;

// Program hampir mirip seperti dengan FormTiketAktif, hanya saja
// disini kita yang awalnya dari aktif ke mati, sekarang dari mati ke aktif.

namespace Resevasi_Tiket_Pesawat
{
    public partial class FormTiketMati : Form
    {
        public FormTiketMati()
        {
            InitializeComponent();
        }

        private void button_to_tiket_aktif_Click(object sender, EventArgs e)
        {
            FormTiketAktif fta = new FormTiketAktif();
            this.Hide(); // tutup form saat ini
            fta.Show(); // tampilkan form daftar tiket aktif
        }

        private void FormTiketMati_Load(object sender, EventArgs e)
        {
            TampilkanData(); // refresh datagrid
        }

        private void TampilkanData()
        {
            string koneksi = "server=localhost;user id=root;password=;database=katas";
            MySqlConnection conn = new MySqlConnection(koneksi);
            conn.Open();

            // 1. Hitung total baris dulu
            MySqlCommand countCmd = new MySqlCommand("SELECT COUNT(*) FROM tiket_mati", conn);
            int total = Convert.ToInt32(countCmd.ExecuteScalar());

            // Set progress bar
            progressBar1.Value = 0;
            progressBar1.Maximum = total;
            progressBar1.Visible = true;

            // 2. Ambil data
            string query = "SELECT * FROM tiket_mati";
            MySqlCommand cmd = new MySqlCommand(query, conn);
            MySqlDataReader rd = cmd.ExecuteReader();

            dataGridView_active_ticket.Rows.Clear(); // menghapus isi lama
            while (rd.Read())
            {
                // Tambahkan row baru
                dataGridView_active_ticket.Rows.Add(
                    rd["id"].ToString(),
                    rd["nama_kontak"].ToString(),
                    rd["tanggal_lahir"].ToString(),
                    rd["email"].ToString(),
                    rd["no_telp"].ToString(),
                    rd["negara_keberangkatan"].ToString(),
                    rd["negara_tujuan"].ToString(),
                    rd["tanggal_waktu_keberangkatan"].ToString(),
                    rd["kelas_layanan"].ToString(),
                    rd["layanan_lainnya"].ToString(),
                    rd["hotel_pilihan"].ToString(),
                    rd["penerbangan_pilihan"].ToString(),
                    rd["informasi_tambahan"].ToString(),
                    rd["harga_tiket"].ToString()
                );
            }

            // 3. Update progress setiap baris
            if (progressBar1.Value < progressBar1.Maximum)
                progressBar1.Value++;
        }

        rd.Close();
        conn.Close();

        // 4. Sembunyikan progress bar setelah selesai
        progressBar1.Visible = false;
    }
}
```

```

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

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

private void HapusData(string id)
{
    string koneksi = "server=localhost;user id=root;password=;database=katas";
    using (MySqlConnection conn = new MySqlConnection(koneksi))
    {
        conn.Open();
        string query = "DELETE FROM tiket_mati WHERE id = @id";
        MySqlCommand cmd = new MySqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@id", id);
        cmd.ExecuteNonQuery();
    }
}
private void PindahkanBaris(string id)
{
    string koneksi = "server=localhost;user id=root;password=;database=katas";
    using (MySqlConnection conn = new MySqlConnection(koneksi))
    {
        conn.Open();

        // 1) Insert ke tabel tujuan
        string insertQuery = "INSERT INTO tiket SELECT * FROM tiket_mati WHERE id = @id";
        MySqlCommand cmdInsert = new MySqlCommand(insertQuery, conn);
        cmdInsert.Parameters.AddWithValue("@id", id);
        cmdInsert.ExecuteNonQuery();

        // 2) Hapus dari tabel asal
        string deleteQuery = "DELETE FROM tiket_mati WHERE id = @id";
        MySqlCommand cmdDelete = new MySqlCommand(deleteQuery, conn);
        cmdDelete.Parameters.AddWithValue("@id", id);
        cmdDelete.ExecuteNonQuery();
    }
}
private void button_nonaktifkan_Click(object sender, EventArgs e)
{
}

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

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

private void button_refresh_Click_1(object sender, EventArgs e)
{
    TampilkanData();
}

private void button_aktifkan_Click(object sender, EventArgs e)
{
    if (dataGridView_active_ticket.CurrentRow == null)
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string id = dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();

    if (MessageBox.Show("Pindahkan data ini ke Daftar Tiket Aktif?", "Konfirmasi", MessageBoxButtons.YesNo,
    MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        PindahkanBaris(id);
        TampilkanData(); // refresh datagrid
        lbl_informasi.Text = ""; // clear informasi selengkapnya
    }
}

private void button_delete_Click_1(object sender, EventArgs e)
{
    if (dataGridView_active_ticket.CurrentRow == null)
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string id = dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();
}

```

```

        if (MessageBox.Show("Yakin hapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        HapusData(id);
        TampilkanData(); // Refresh datagrid
        lbl_informasi.Text = "";// clear informasi selengkapnya
    }
}

private void button_back_to_form_Click_1(object sender, EventArgs e)
{
    FormFormulir ff = new FormFormulir();
    this.Hide(); // tutup form saat ini
    ff.Show(); // tampilkan form buat tiket
}

private void button_informasi_tambahan_Click_1(object sender, EventArgs e)
{
}

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

private void pictureBox1_Click(object sender, EventArgs e)
{
    if (dataGridView_active_ticket.CurrentRow != null)
    {
        lbl_informasi.Text = "";
        lbl_informasi.Text += "# Tiket ID: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n# Informasi Pribadi";
        lbl_informasi.Text += "\n * Nama Kontak: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Nama_Kontak"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Tanggal lahir: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_TTL"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * E-mail: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Email"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * No. Telp: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Nomor_Telp"].Value?.ToString() ?? "";

        lbl_informasi.Text += "\n\n# Detail Perjalanan";
        lbl_informasi.Text += "\n * Negara Keberangkatan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Negara_Keberangkatan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Negara Tujuan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Negara_Tujuan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Tanggal_Waktu_Berangkat"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Kelas Layanan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Kelas_Layanan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Layanan Lainnya: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Layanan"].Value?.ToString() ?? "";

        lbl_informasi.Text += "\n\n# Rincian Perjalanan";
        if (!string.IsNullOrEmpty(dataGridView_active_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString()))
        {
            lbl_informasi.Text += "\n * Hotel Pilihan: ";
            lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Nama_Hotel"].Value?.ToString() ?? "";
        }
        lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Penerbangan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Informasi Tambahan: ";
        lbl_informasi.Text += dataGridView_active_ticket.CurrentRow.Cells["Column_Informasi_Tambahan"].Value?.ToString() ?? "";
        lbl_informasi.Text += "\n * Harga Tiket: ";
        lbl_informasi.Text += "Rp. " + FormatMoney(Convert.ToInt32(dataGridView_active_ticket.CurrentRow.Cells["Total_Harga"].Value));
        }
        else
        {
            MessageBox.Show("Pilih baris tiket yang ingin anda lihat informasi tambahannya", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    public static string FormatMoney(int angka)
    {
        return angka.ToString("#,0").Replace(",",".");
    }

    private void pictureBox6_Click(object sender, EventArgs e)
    {
        TampilkanData();
    }

    private void pictureBox3_Click(object sender, EventArgs e)
    {
        if (dataGridView_active_ticket.CurrentRow == null)
        {
            MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

        return;
    }

    string id = dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();

    if (MessageBox.Show("Pindahkan data ini ke Daftar Tiket Aktif?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        PindahkanBaris(id);
        TampilkanData(); // refresh datagrid
        lbl_informasi.Text = ""; // clear informasi selengkapnya
    }
}

private void pictureBox4_Click(object sender, EventArgs e)
{
    if (dataGridView_active_ticket.CurrentRow == null)
    {
        MessageBox.Show("Tidak ada baris yang dipilih.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return;
    }

    string id = dataGridView_active_ticket.CurrentRow.Cells["Column_ID"].Value.ToString();

    if (MessageBox.Show("Yakin hapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Warning) == DialogResult.Yes)
    {
        HapusData(id);
        TampilkanData(); // Refresh datagrid
        lbl_informasi.Text = ""; // clear informasi selengkapnya
    }
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    FormTiketAktif fta = new FormTiketAktif();
    this.Hide(); // tutup form saat ini
    fta.Show(); // tampilkan form daftar tiket aktif
}

private void pictureBox7_Click(object sender, EventArgs e)
{
    FormFormulir ff = new FormFormulir();
    this.Hide(); // tutup form saat ini
    ff.Show(); // tampilkan form buat tiket
}

private void pictureBox8_Click(object sender, EventArgs e)
{
    FormMenu fm = new FormMenu();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form menu
}

private void dataGridView_active_ticket_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
}
}

```

5. FormVerifikasi.cs

```
● ● ●

using MySql.Data.MySqlClient;
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 Resevasi_Tiket_Pesawat
{
    public partial class FormVerifikasi : Form
    {
        public FormVerifikasi()
        {
            InitializeComponent();
        }

        private void button_verifikasi_tiket_Click(object sender, EventArgs e)
        {

        }

        private void button1_Click(object sender, EventArgs e)
        {
            FormFormulir ff = new FormFormulir();
            this.Hide(); // tutup form saat ini
            ff.Show(); // tampilkan form buat tiket
        }

        private void HapusData(string id)
        {
            string koneksi = "server=localhost;user id=root;password=;database=katas";
            using (MySqlConnection conn = new MySqlConnection(koneksi))
            {
                conn.Open();
                string query = "DELETE FROM tiket WHERE id = @id";
                MySqlCommand cmd = new MySqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@id", id);
                cmd.ExecuteNonQuery();
            }
        }

        private void PindahkanBaris(string id)
        {
            string koneksi = "server=localhost;user id=root;password=;database=katas";
            using (MySqlConnection conn = new MySqlConnection(koneksi))
            {
                conn.Open();

                // 1) Insert ke tabel tujuan
                string insertQuery = "INSERT INTO tiket_mati SELECT * FROM tiket WHERE id = @id";
                MySqlCommand cmdInsert = new MySqlCommand(insertQuery, conn);
                cmdInsert.Parameters.AddWithValue("@id", id);
                cmdInsert.ExecuteNonQuery();

                // 2) Hapus dari tabel asal
                string deleteQuery = "DELETE FROM tiket WHERE id = @id";
                MySqlCommand cmdDelete = new MySqlCommand(deleteQuery, conn);
                cmdDelete.Parameters.AddWithValue("@id", id);
                cmdDelete.ExecuteNonQuery();
            }
        }

        private void button_delete_Click(object sender, EventArgs e)
        {
            if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text))
            {
                MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                textBox_tiket_id.Focus();
                lbl_informasi.Text = "";
                return;
            }
            if (double.TryParse(textBox_tiket_id.Text, out double id))
            {
                string koneksi = "server=localhost;user id=root;password=;database=katas";
                using (MySqlConnection conn = new MySqlConnection(koneksi))
                {
                    conn.Open();

                    string query = "SELECT * FROM tiket WHERE id = @id";
                    MySqlCommand cmd = new MySqlCommand(query, conn);
                    cmd.Parameters.AddWithValue("@id", id);

                    MySqlDataReader rd = cmd.ExecuteReader();

                    if (rd.Read())
                    {
```

```

        // Tiket ditemukan → tampilkan ke label
        if (MessageBox.Show("Yakin menghapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) == DialogResult.Yes)
        {
            HapusData(id.ToString());
            lbl_informasi.Text = ""; // clear informasi selengkapnya
        }

    }
else
{
    MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    lbl_informasi.Text = "";
    textBox_tiket_id.Focus();
}

}

else
{
    MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
    textBox_tiket_id.Focus();
    return;
}
}

private void button_nonaktifkan_Click(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text))
    {
        MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_tiket_id.Focus();
        lbl_informasi.Text = "";
        return;
    }
    if (double.TryParse(textBox_tiket_id.Text, out double id))
    {
        string koneksi = "server=localhost;user id=root;password=;database=katas";
        using (MySqlConnection conn = new MySqlConnection(koneksi))
        {
            conn.Open();

            string query = "SELECT * FROM tiket WHERE id = @id";
            MySqlCommand cmd = new MySqlCommand(query, conn);
            cmd.Parameters.AddWithValue("@id", id);

            MySqlDataReader rd = cmd.ExecuteReader();

            if (rd.Read())
            {
                // Tiket ditemukan → tampilkan ke label
                if (MessageBox.Show("Yakin menonaktifkan data ini?", "Konfirmasi", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) == DialogResult.Yes)
                {
                    PindahkanBaris(id.ToString());
                    lbl_informasi.Text = ""; // clear informasi selengkapnya
                }

            }
            else
            {
                MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                lbl_informasi.Text = "";
                textBox_tiket_id.Focus();
            }
        }
    }
    else
    {
        MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_tiket_id.Focus();
        return;
    }
}

private void button_cek_aktif_tiket_Click(object sender, EventArgs e)
{
    FormTiketAktif fta = new FormTiketAktif();
    this.Hide(); // tutup form saat ini
    fta.Show(); // tampilkan form daftar tiket aktif
}

private void button_menu_tiket_mati_Click(object sender, EventArgs e)
{
    FormTiketMati fm = new FormTiketMati();
}

```

```

        this.Hide(); // tutup form saat ini
        fm.Show(); // tampilkan form daftar tiket mati
    }

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

    private void pictureBox2_Click(object sender, EventArgs e) // ketika gambar tombolverifikasi diklik
    {
        if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text)) // jika textBox_tiket_id.Text kosong
        {
            MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error); // msgbox error
            textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
            lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
            return; // berhentikan program
        }
        if (double.TryParse(textBox_tiket_id.Text, out double id)) // jika textBox_tiket_id.Text adalah double masukkan ke var id
        {
            string koneksi = "server=localhost;user id=root;password=;database=katasi;"; // server, userid, password, dan nama
            database

            using (MySqlConnection conn = new MySqlConnection(koneksi)) // koneksi database
            {
                conn.Open(); // buka database

                string query = "SELECT * FROM tiket WHERE id = @id"; // perintah sql
                MySqlCommand cmd = new MySqlCommand(query, conn); // var command sql
                cmd.Parameters.AddWithValue("@id", id); // menambahkan parameter ke perintah sql

                MySqlDataReader rd = cmd.ExecuteReader(); // eksekusi cmd sql

                if (rd.Read()) // jika terbaca
                {
                    // Tiket ditemukan ~ tampilkan ke label
                    MessageBox.Show("Status: Tiket Aktif", "Succes", MessageBoxButtons.OK, MessageBoxIcon.Information);

                    lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
                    lbl_informasi.Text += "# Tiket ID: ";
                    lbl_informasi.Text += rd["id"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n# Informasi Pribadi";
                    lbl_informasi.Text += "\n * Nama Kontak: ";
                    lbl_informasi.Text += rd["nama_kontak"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Tanggal lahir: ";
                    lbl_informasi.Text += rd["tanggal_lahir"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * E-mail: ";
                    lbl_informasi.Text += rd["email"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * No. Telp: ";
                    lbl_informasi.Text += rd["no_telp"].ToString() ?? ""; // isi dari kolom databases

                    lbl_informasi.Text += "\n\n# Detail Perjalanan";
                    lbl_informasi.Text += "\n * Negara Keberangkatan: ";
                    lbl_informasi.Text += rd["negara_keberangkatan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Negara Tujuan: ";
                    lbl_informasi.Text += rd["negara_tujuan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Tanggal dan Waktu Keberangkatan: ";
                    lbl_informasi.Text += rd["tanggal_waktu_keberangkatan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Kelas Layanan: ";
                    lbl_informasi.Text += rd["kelas_layanan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Layanan Lainnya: ";
                    lbl_informasi.Text += rd["layanan_lainnya"].ToString() ?? ""; // isi dari kolom databases

                    lbl_informasi.Text += "\n\n# Rincian Perjalanan";
                    if (!string.IsNullOrEmpty(rd["hotel_pilihan"].ToString())) // jika kolom hotel_pilihan dari database kosong
                    {
                        lbl_informasi.Text += "\n * Hotel Pilihan: ";
                        lbl_informasi.Text += rd["hotel_pilihan"].ToString() ?? ""; // isi dari kolom databases
                    }
                    lbl_informasi.Text += "\n * Penerbangan Pilihan: ";
                    lbl_informasi.Text += rd["penerbangan_pilihan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Informasi Tambahan: ";
                    lbl_informasi.Text += rd["informasi_tambahan"].ToString() ?? ""; // isi dari kolom databases
                    lbl_informasi.Text += "\n * Harga Tiket: ";
                    lbl_informasi.Text += "Rp. " + FormatMoney(Convert.ToInt32(rd["harga_tiket"])); // isi dari kolom databases
                    lalu format ke money
                }
                else
                {
                    MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    lbl_informasi.Text = ""; // kosongkan lbl_informasi.Text
                    textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
                }
            }
        }
        else
        {
            MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

```

```

        textBox_tiket_id.Focus(); // fokus ke textBox_tiket_id
        return; // berhentikan program
    }

    public static string FormatMoney(int angka)
    {
        return angka.ToString("#,0").Replace(",",".");
    }

    private void pictureBox7_Click(object sender, EventArgs e)
    {
        FormFormulir ff = new FormFormulir();
        this.Hide(); // tutup form saat ini
        ff.Show(); // tampilkan form buat tiket
    }

    private void pictureBox3_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text))
        {
            MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            textBox_tiket_id.Focus();
            lbl_informasi.Text = "";
            return;
        }
        if (double.TryParse(textBox_tiket_id.Text, out double id))
        {
            string koneksi = "server=localhost;user id=root;password=;database=katasi;";

            using (MySqlConnection conn = new MySqlConnection(koneksi))
            {
                conn.Open();

                string query = "SELECT * FROM tiket WHERE id = @id";
                MySqlCommand cmd = new MySqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@id", id);

                MySqlDataReader rd = cmd.ExecuteReader();

                if (rd.Read())
                {
                    // Tiket ditemukan - tampilkan ke label
                    if (MessageBox.Show("Yakin menonaktifkan data ini?", "Konfirmasi", MessageBoxButtons.YesNo,
MessageIcon.Warning) == DialogResult.Yes)
                    {
                        PindahkanBaris(id.ToString());
                        lbl_informasi.Text = ""; // clear informasi selengkapnya
                    }
                }
                else
                {
                    MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
                    lbl_informasi.Text = "";
                    textBox_tiket_id.Focus();
                }
            }
        }
        else
        {
            MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            textBox_tiket_id.Focus();
            return;
        }
    }

    private void pictureBox4_Click(object sender, EventArgs e)
    {
        if (string.IsNullOrWhiteSpace(textBox_tiket_id.Text))
        {
            MessageBox.Show("ID Tiket tidak terisi", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            textBox_tiket_id.Focus();
            lbl_informasi.Text = "";
            return;
        }
        if (double.TryParse(textBox_tiket_id.Text, out double id))
        {
            string koneksi = "server=localhost;user id=root;password=;database=katasi;";

            using (MySqlConnection conn = new MySqlConnection(koneksi))
            {
                conn.Open();

                string query = "SELECT * FROM tiket WHERE id = @id";
                MySqlCommand cmd = new MySqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@id", id);

```

```

        MySqlDataReader rd = cmd.ExecuteReader();
        if (rd.Read())
        {
            // Tiket ditemukan ~ tampilkan ke label
            if (MessageBox.Show("Yakin menghapus data ini?", "Konfirmasi", MessageBoxButtons.YesNo,
MessageBoxIcon.Warning) == DialogResult.Yes)
            {
                HapusData(id.ToString());
                lbl_informasi.Text = ""; // clear informasi selengkapnya
            }
        }
        else
        {
            MessageBox.Show("Status: Tiket Tidak Aktif", "Warning", MessageBoxButtons.OK, MessageBoxIcon.Warning);
            lbl_informasi.Text = "";
            textBox_tiket_id.Focus();
        }
    }

    }
    else
    {
        MessageBox.Show("ID Tiket harus angka", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        textBox_tiket_id.Focus();
        return;
    }
}

private void pictureBox6_Click(object sender, EventArgs e)
{
    FormTiketAktif fta = new FormTiketAktif();
    this.Hide(); // tutup form saat ini
    fta.Show(); // tampilkan form daftar tiket aktif
}

private void pictureBox5_Click(object sender, EventArgs e)
{
    FormTiketMati fm = new FormTiketMati();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form daftar tiket mati
}

private void pictureBox8_Click(object sender, EventArgs e)
{
    FormMenu fm = new FormMenu();
    this.Hide(); // tutup form saat ini
    fm.Show(); // tampilkan form menu
}

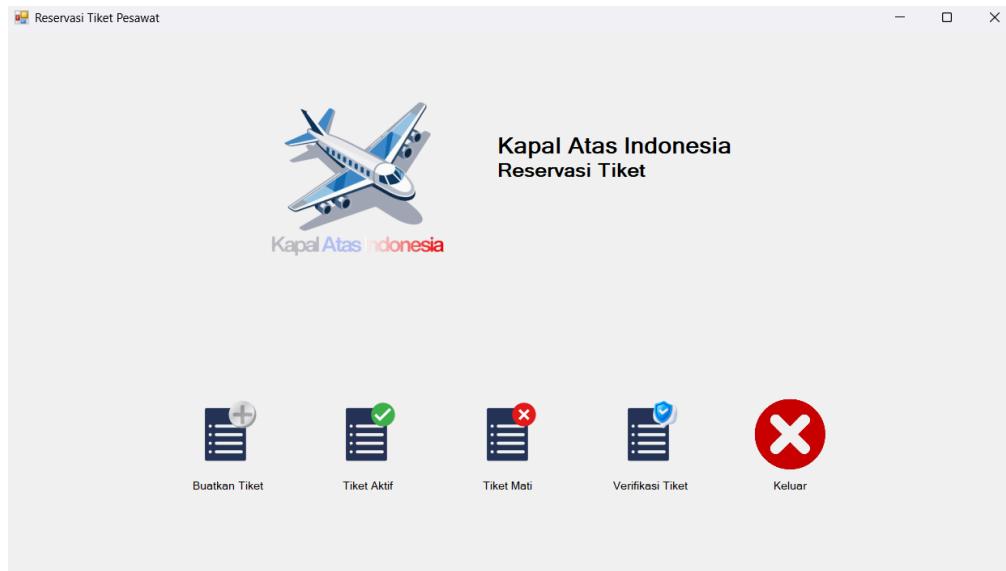
private void lbl_informasi_Click(object sender, EventArgs e)
{
}

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

```

B. Screenshot Program

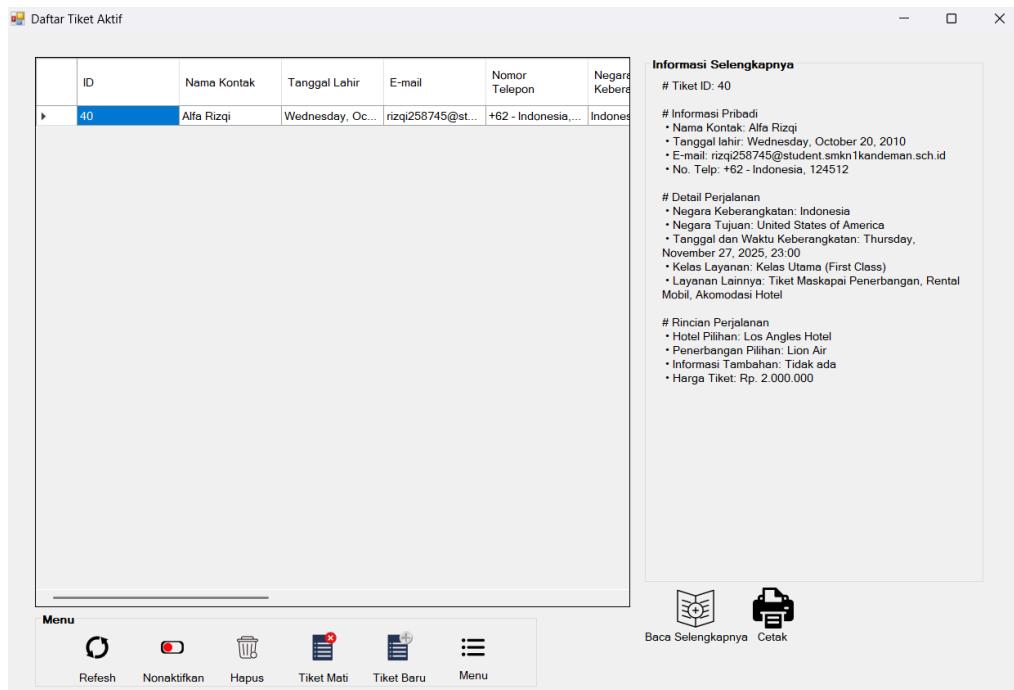
1. Halaman Menu



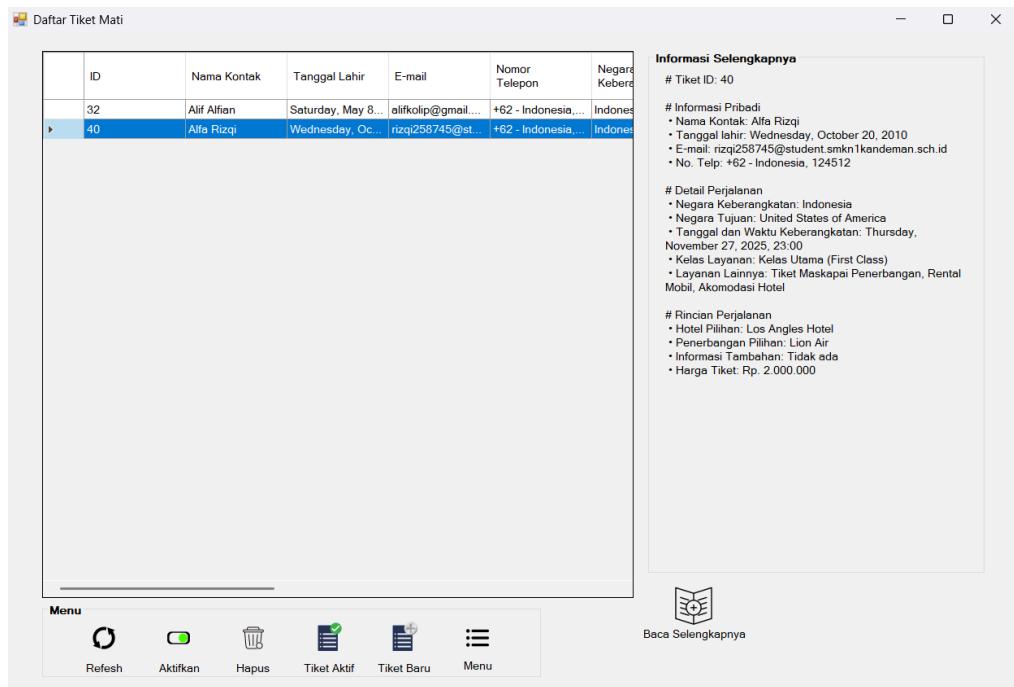
2. Buat Tiket Baru

The screenshot shows the "Formulir Tiket" (Ticket Form) window. The left side contains sections for "Informasi Pribadi" (Personal Information) and "Detail Perjalanan" (Travel Details). The "Informasi Pribadi" section includes fields for Name, Date of Birth, Email, Nationality, and Phone Number. The "Detail Perjalanan" section includes fields for Destination, Date, Time, and Class. A note at the bottom says "Silakan centang layanan yang pelanggan butuhkan" (Please check the services the customer needs) with checkboxes for flight, car rental, and hotel accommodation. The right side contains sections for "Informasi Selengkapnya" (More Information), "Rincian Perjalanan" (Travel Details), "Bantuan Kasir" (Cashier Help), and a "Menu" bar. The "Informasi Selengkapnya" section lists travel details like destination, date, and class. The "Rincian Perjalanan" section lists hotel and flight information. The "Bantuan Kasir" section shows a breakdown of costs: total price (Rp. 2,000,000), amount paid (300,000), amount due (2500), and change given (Rp. 997,500). The "Menu" bar at the bottom includes icons for "Tiket Aktif", "Tiket Mati", "Verifikasi Tiket", and "Menu".

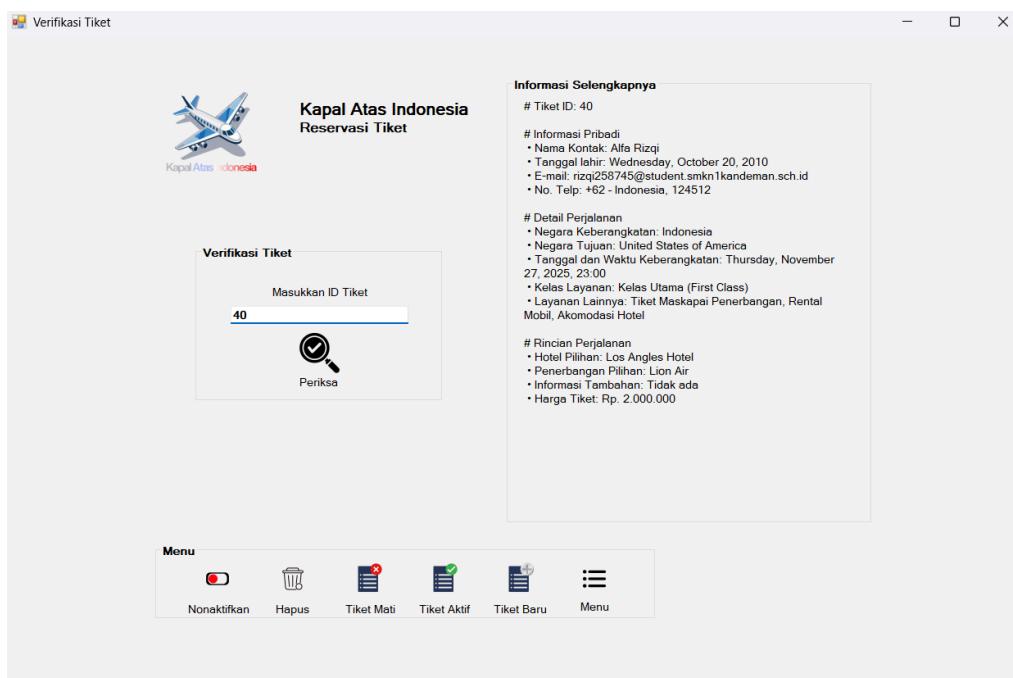
3. Daftar Tiket Aktif



4. Daftar Tiket Mati



5. Verifikasi Tiket



6. Tiket Cetak

