

**RANCANG BANGUN SISTEM TRANSAKSI TIKET  
PESAWAT BERBASIS APLIKASI DESKTOP GUI DENGAN  
BAHASA C DAN DATABASE MYSQL**

**PROJECT AKHIR  
PRAKTIKUM DASAR PEMROGRAMAN**



**Dosen :**

Norma Ningsih, S.ST., M.T.

**Oleh :**

Muqsith Barru Pamungkas - 2423600035

Riski Gana Prasetya - 2423600053

**TEKNOLOGI REKAYASA INTERNET  
DEPARTEMEN TEKNIK ELEKTRO  
POLITEKNIK ELEKTRONIKA NEGERI  
SURABAYA  
2024**

# 1 Tujuan

- Merancang sistem penjualan dan pembelian tiket pesawat menggunakan bahasa C berbasis GUI (GTK) dengan MySQL sebagai database

## 2 Pendahuluan

### 2.1 Latar Belakang

Sistem Pembelian dan Penjualan Tiket Pesawat "Myber" yang dibuat ini merupakan sebuah sistem transaksi/e-commerce berbasis aplikasi desktop (GUI) yang dibuat dengan menggunakan bahasa C dan GTK atau GIMP Toolkit sebagai framework untuk membangun tampilan, serta mengintegrasikan sebuah database menggunakan MySQL.

Sistem yang dirancang memiliki dua fungsi utama yaitu untuk penjualan dan pembelian tiket pesawat. Sehingga aplikasi ini memiliki dua tipe pengguna, yaitu admin sebagai penjual dan memiliki akses untuk membuat atau mengupdate detail tiket yang dijual, lalu terdapat pengguna atau customer yang berada di sisi pembeli. Dari sisi pengguna atau customer sendiri dapat melakukan register atau pendaftaran akun.

Dalam pengembangan aplikasi ini, kami melakukan development di lingkungan/environment Linux, menggunakan distro Ubuntu 22.04.4 LTS WSL (kernel 5.15.146.1-microsoft-standard-WSL2) dan Arch Linux (kernel: 6.9.3-zen1-1-zen). Kami menggunakan Makefile (make) untuk melakukan otomatisasi dalam mengompilasi kode dengan gcc.

### 2.2 Bahasa C

Bahasa C oleh Dennis M. Ritchie pada tahun 1972 di laboratorium Bell ditemukan. Pengembangan BPCL (Basic Combined Program Language) yang dibuat oleh Dr. Martin Richard yang selanjutnya dikembangkan oleh Ken Thompson dan diseleksi dengan Language B. Dari ketertarikan Dennis pada penerjemah bahasa B, kemudian dikembangkan menjadi sebuah kompiler yang disebut C. Bahasa pemrograman C merupakan sebuah bahasa pemrograman tingkat menengah yang relatif mudah untuk dipelajari, selain bahasa yang mudah untuk dipelajari bahasa C juga memiliki kemampuan performa yang tinggi serta bahasa C mempunyai sifat portable dalam hal ini bahasa C yang ditulis dalam satu komputer bisa dipindahkan ke komputer lain tanpa mengotak-atikinya, tanpa muncul kerumitan dalam modifikasinya. Dalam pengaplikasiannya bahasa C sering digunakan dalam pembuatan berbagai aplikasi seperti sistem operasi, antivirus, hingga compiler bahasa pemrograman, bahasa C ini juga merupakan induk dari beberapa bahasa seperti bahasa C++, C-Sharp, dan Java. Karena sifat bahasa C yang mudah dipahami, bahasa C menjadi bahasa pemrograman yang paling sering digunakan.

### 2.3 GIMP Toolkit (GTK)

GUI atau singkatan dari Graphical User Interface yang memungkinkan pengguna untuk berinteraksi dengan perangkat keras komputer serta memudahkan dalam mengoperasikan sebuah sistem operasi (user friendly). GUI adalah sarana penghubung antara pengguna (user) dengan aplikasi yang digunakan. Salah satu jenis Toolkit GUI yang populer adalah GIMP Toolkit atau GTK. GTK atau dikenal juga GIMP Toolkit adalah

software yang digunakan untuk membangun sebuah tampilan aplikasi(GUI) lintas platform yang pertama kali dikembangkan oleh GNU Image Manipulation Program (GIMP). ToolKit ini menyediakan berbagai macam widget yang sangat membantu dalam pembuatan aplikasi dan mampu support langsung berbagai bahasa seperti c,c++,dan pythpon

## 2.4 MySQL

MySQL, yang dibaca MY-ES-KYOO-EL, adalah sebuah sistem manajemen database relasional (RDBMS) yang bersifat open source dan dibangun berdasarkan Bahasa Pertanyaan Struktural (SQL). Dengan kata lain, MySQL adalah sebuah program yang menggunakan bahasa pemrograman SQL untuk membuat dan mengelola berbagai jenis data yang ada pada database di server. Tiga orang Swedia bernama David Axmark, Allan Larsson, dan Michael Widenius adalah pendiri MySQL, yang dikembangkan pada tahun 1994 oleh perusahaan asal Swedia bernama MySQL AB. Pada tanggal 23 Mei 1995, versi stabil pertama MySQL muncul setelah kurang lebih satu tahun pengembangan.

Pada tahun 2008, Sun Microsystems membeli hak milik MySQL dari Oracle, raksasa teknologi Amerika. Singkatnya, Oracle memiliki hak cipta MySQL. MySQL saat ini adalah salah satu pilihan database yang paling populer untuk berbagai tujuan, seperti membuat dan mengelola database, penyimpanan data, mengelola transaksi e-commerce, pencatatan data, dan, yang paling umum, sebagai database untuk website.

## 3 Instalasi dan Tampilan Aplikasi

### 3.1 Pra-instalasi dan kompilasi

Berikut ini package-package yang dibutuhkan sebelum melakukan kompilasi :

- `gcc` untuk instalasi gcc (GNU Compiler Collection).
- `libgtk-3-dev` untuk instalasi GTK-3.
- `mysql-server` untuk instalasi MySQL server.
- `default-libmysqlclient-dev` untuk client library C MySQL.
- `build-essential` (opsional) untuk instalasi paket gcc sekaligus make di Ubuntu, merupakan meta-package yang berisi berbagai package yang berguna untuk meng-compile program.

### 3.2 Proses Kompilasi

Dalam melakukan kompilasi, pastikan semua package yang dibutuhkan telah terinstal, selanjutnya adalah melakukan kompilasi dengan menggunakan make.

Makefile merupakan sebuah file yang berisi sekumpulan perintah atau set yang dapat digunakan menggunakan make untuk melakukan otomatisasi agar mencapai tujuan tertentu, dalam hal ini yaitu melakukan kompilasi. Berikut ini isi dari Makefile yang kami buat untuk mempermudah proses kompilasi.

A terminal window titled 'myber: zsh — Konsole <2>' showing the contents of a Makefile. The file defines variables for the program name (main), source files (mysql\_func), compiler (gcc), and various flags and libraries. It also includes rules for building the program and cleaning the build directory.

```
myber: zsh — Konsole <2>
New Tab Split View
Copy Paste Find...
> cat Makefile
PROG = main
PROG1 = mysql_func

CC = gcc
CFLAGS = `pkg-config --cflags gtk+-3.0`
LIBS = `pkg-config --libs gtk+-3.0`
MYSQL1 = mysql_config --cflags
MYSQL2 = mysql_config --libs

${PROG}: ${PROG}.c
    ${CC} -Wall ${CFLAGS} -o ${PROG} ${PROG1}.c ${PROG}.c ${LIBS} ${MYSQL1} ${MYSQL2}

clean:
    rm ${PROG}
```

Figure 1: Makefile

Selanjutnya untuk melakukan kompilasi, gunakan perintah :

make

A terminal window showing the output of the 'make' command. It displays the compilation command using gcc, including the source files and various flags and libraries.

```
myber: zsh — Konsole <2>
New Tab Split View
Copy Paste Find...
> make
gcc -Wall `pkg-config --cflags gtk+-3.0` -o main mysql_func.c main.c `pkg-config --libs gtk+-3.0` `mysql_config --cflags` `mysql_config --libs`
```

Figure 2: Make

Selanjutnya file binary program akan dihasilkan dari proses kompilasi, untuk menjalankan gunakan perintah :

./main

A terminal window showing the output of the 'ls' command, which lists the files in the current directory. Below that, the command './main' is entered, indicating the execution of the compiled program.

```
myber: zsh — Konsole <2>
New Tab Split View
Copy Paste Find...
> ls
assets docs main main.c Makefile mysql_func.c mysql_func.h userinfo.h

./main
```

Figure 3: File program hasil kompilasi

### 3.3 Welcome Screen

Berikut ini adalah tampilan ketika aplikasi pertama kali dibuka



Figure 4: File program hasil kompilasi

### 3.4 Admin - Halaman Login

Selanjutnya adalah halaman login admin, dengan tampilan sebagai berikut:

The image shows the Admin Login page. The page has a dark background with the text "MYBER MELAYANI DENGAN SETULUS HATI" and "Login-as Admin". Below this, there is a prompt "Silahkan Login Sebagai Admin" and a form with fields for "Username:" and "Password:", a "Login" button, and a "Kembali ke halaman awal" button.

Figure 5: Halaman Login Admin

### 3.5 Admin - Tampilan Awal

Ini merupakan tampilan awal atau welcome screen Admin

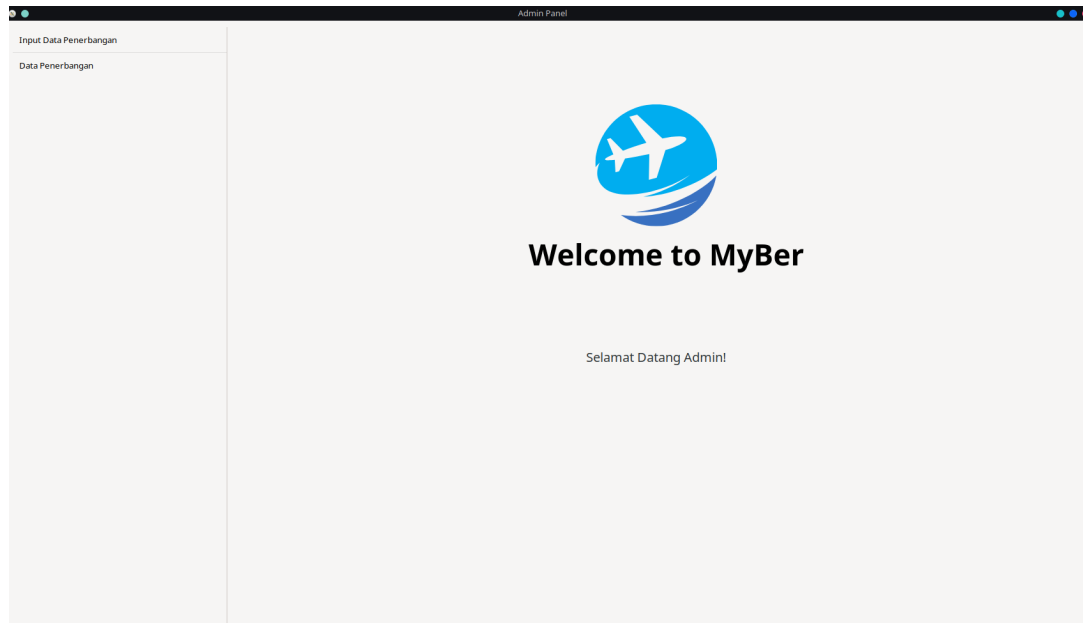


Figure 6: Welcome Admin

### 3.6 Admin - Input Data Pembelian

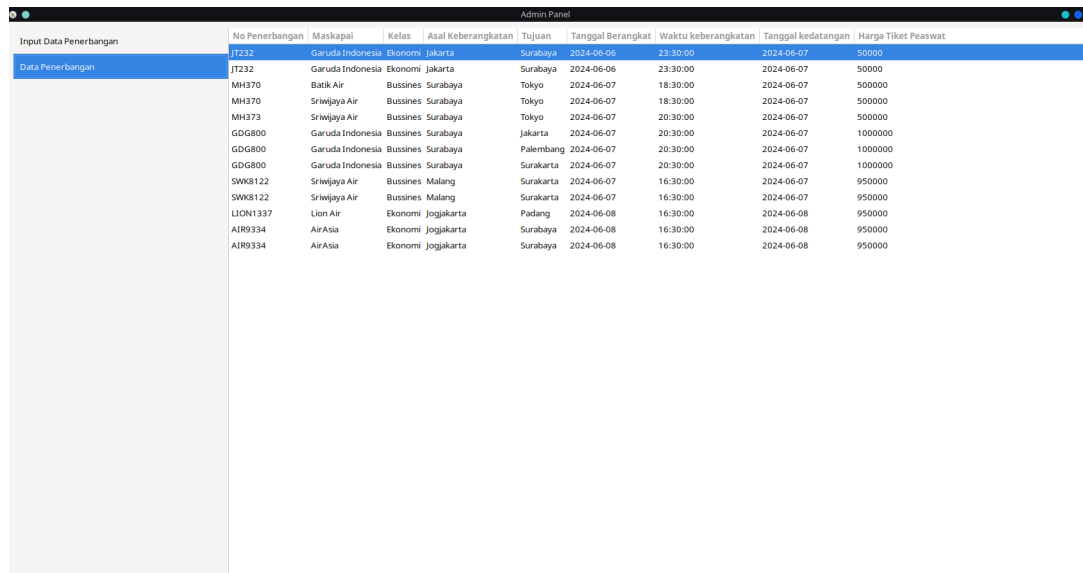
Ini merupakan halaman untuk input data pembelian tiket oleh Admin

A screenshot of the "Input Data Penerbangan" form within the "Admin Panel". The sidebar on the left contains the same menu items as in Figure 6. The form itself is a light gray rectangle with various input fields. At the top right of the form area, it says "No Penerbangan". The fields include: "GDGAN1337" (flight number), "Garuda Indonesia" (selected from a dropdown for "Nama Maskapai"), "Business" (selected from a dropdown for "Jenis Kelas Pesawat"), "Sidoarjo" (selected from a dropdown for "Asal Keberangkatan Pesawat"), "Denpasar" (selected from a dropdown for "Tujuan Kedatangan Pesawat"), "2024-06-06" (selected from a dropdown for "Jadwal Penerbangan:"), "17" (selected from a dropdown for "Jam Penerbangan:"), "2024-06-06" (selected from a dropdown for "Jadwal Kedatangan:"), "18" (selected from a dropdown for "Jam Kedatangan:"), and "250000" (entered in a text field for "Harga"). A "Simpan" button is located at the bottom right of the form. The window has standard macOS-style window controls in the top-left corner.

Figure 7: Halaman Input Data Admin

### 3.7 Admin - Data Penerbangan

Ini merupakan halaman data penerbangan yang sebelumnya diinputkan oleh Admin



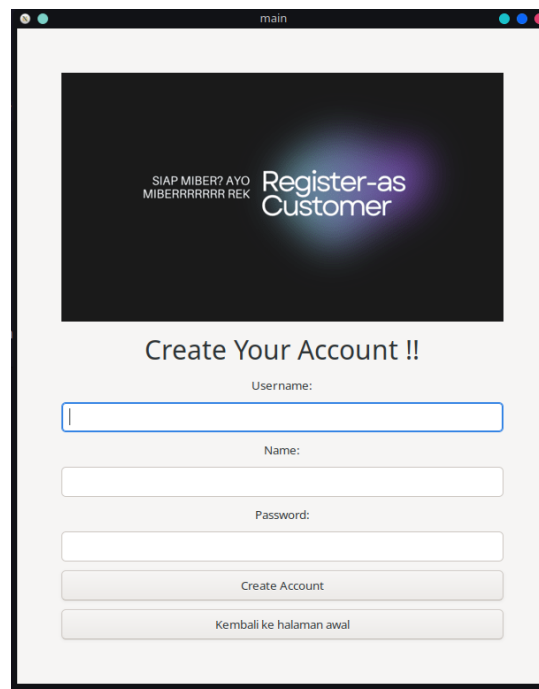
The screenshot shows a web application window titled 'Admin Panel'. On the left is a sidebar with 'Input Data Penerbangan' and 'Data Penerbangan' (selected). The main area displays a table with flight data. The table has 9 columns: No Penerbangan, Maskapai, Kelas, Asal Keberangkatan, Tujuan, Tanggal Berangkat, Waktu keberangkatan, Tanggal kedatangan, and Harga Tiket Pesawat. The data includes flights from Garuda Indonesia, Batik Air, Sriwijaya Air, and Lion Air to various destinations like Surabaya, Tokyo, Jakarta, Palembang, and Padang.

No Penerbangan	Maskapai	Kelas	Asal Keberangkatan	Tujuan	Tanggal Berangkat	Waktu keberangkatan	Tanggal kedatangan	Harga Tiket Pesawat
JT232	Garuda Indonesia	Ekonomi	Jakarta	Surabaya	2024-06-06	23:30:00	2024-06-07	500000
MH370	Batik Air	Bussines	Surabaya	Tokyo	2024-06-07	18:30:00	2024-06-07	500000
MH370	Sriwijaya Air	Bussines	Surabaya	Tokyo	2024-06-07	18:30:00	2024-06-07	500000
MH373	Sriwijaya Air	Bussines	Surabaya	Tokyo	2024-06-07	20:30:00	2024-06-07	500000
GDG800	Garuda Indonesia	Bussines	Surabaya	Jakarta	2024-06-07	20:30:00	2024-06-07	1000000
GDG800	Garuda Indonesia	Bussines	Surabaya	Palembang	2024-06-07	20:30:00	2024-06-07	1000000
GDG800	Garuda Indonesia	Bussines	Surabaya	Surakarta	2024-06-07	20:30:00	2024-06-07	1000000
SWK8122	Sriwijaya Air	Bussines	Malang	Surakarta	2024-06-07	16:30:00	2024-06-07	950000
SWK8122	Sriwijaya Air	Bussines	Malang	Surakarta	2024-06-07	16:30:00	2024-06-07	950000
LION1337	Lion Air	Ekonomi	Jogjakarta	Padang	2024-06-08	16:30:00	2024-06-08	950000
AIK9334	Air Asia	Ekonomi	Jogjakarta	Surabaya	2024-06-08	16:30:00	2024-06-08	950000
AIK9334	Air Asia	Ekonomi	Jogjakarta	Surabaya	2024-06-08	16:30:00	2024-06-08	950000

Figure 8: Halaman Input Data Admin

### 3.8 Customer - Halaman Register

Berikut merupakan tampilan halaman register untuk pengguna/customer

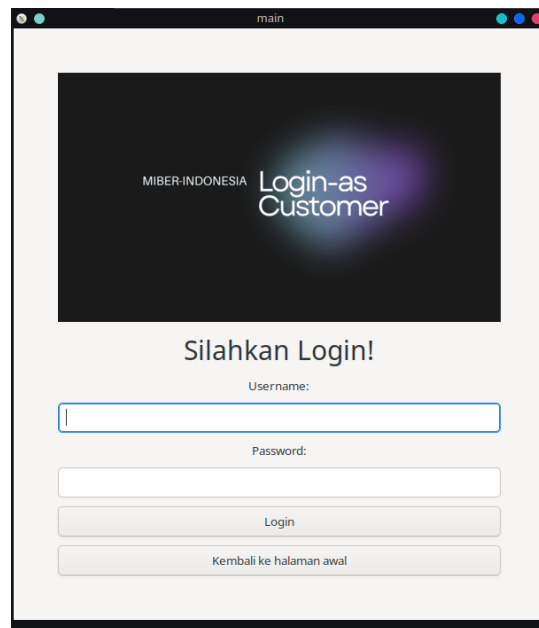


The screenshot shows a web application window titled 'main'. The page has a dark header with the text 'SIAP MIBER? AYO MIBERRRRRRR REK' and 'Register-as Customer'. Below the header is a large black box with the text 'Create Your Account !!'. Underneath is a form with fields for 'Username:', 'Name:', and 'Password:'. There are two buttons at the bottom: 'Create Account' and 'Kembali ke halaman awal'.

Figure 9: Halaman Register Customer

### 3.9 Customer - Halaman Login

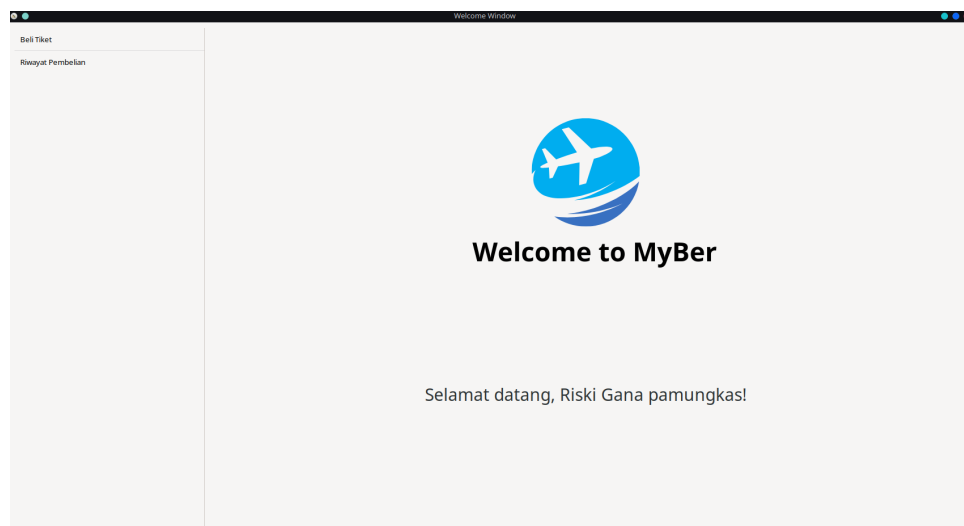
Ini merupakan tampilan halaman login untuk pengguna/customer



**Figure 10:** Halaman Login Customer

### 3.10 Customer - Tampilan Awal

Berikut adalah tampilan ketika user selesai login



**Figure 11:** Halaman Utama User

### 3.11 Customer - Halaman Pembelian Tiket

Ini merupakan tampilan halaman pembelian tiket



No Penerbangan	Maskapai	Kelas	Asal Keberangkatan	Tujuan	Tanggal Berangkat	Waktu keberangkatan	Tanggal kedatangan	Waktu kedatangan	Harga Tiket Pesawat
JT232	Garuda Indonesia	Ekonomi	Jakarta	Surabaya	2024-06-06	23:30:00	2024-06-07	00:03:00	50000
JT232	Garuda Indonesia	Ekonomi	Jakarta	Surabaya	2024-06-06	23:30:00	2024-06-07	00:03:00	50000
MH370	Batik Air	Business	Surabaya	Tokyo	2024-06-07	18:30:00	2024-06-07	21:52:00	500000
MH370	Sriwijaya Air	Business	Surabaya	Tokyo	2024-06-07	18:30:00	2024-06-07	21:52:00	500000
MH373	Sriwijaya Air	Business	Surabaya	Tokyo	2024-06-07	20:30:00	2024-06-07	23:52:00	500000
GDG800	Garuda Indonesia	Business	Surabaya	Jakarta	2024-06-07	20:30:00	2024-06-07	23:52:00	1000000
GDG800	Garuda Indonesia	Business	Surabaya	Palembang	2024-06-07	20:30:00	2024-06-07	23:52:00	1000000
GDG800	Garuda Indonesia	Business	Surabaya	Surakarta	2024-06-07	20:30:00	2024-06-07	23:52:00	1000000
SWK3122	Sriwijaya Air	Business	Malang	Surakarta	2024-06-07	16:30:00	2024-06-07	20:52:00	950000
SWK3122	Sriwijaya Air	Business	Malang	Surakarta	2024-06-07	16:30:00	2024-06-07	20:52:00	950000
LION1337	Lion Air	Ekonomi	Jogjakarta	Padang	2024-06-08	16:30:00	2024-06-08	20:52:00	950000
AIR3334	AirAsia	Ekonomi	Jogjakarta	Surabaya	2024-06-08	16:30:00	2024-06-08	20:52:00	950000
AIR3334	AirAsia	Ekonomi	Jogjakarta	Surabaya	2024-06-08	16:30:00	2024-06-08	20:52:00	950000

Figure 12: Halaman Pembelian Tiket

### 3.12 Customer - Riwayat Pembelian Tiket

Ini merupakan tampilan dari riwayat pembelian tiket oleh Customer

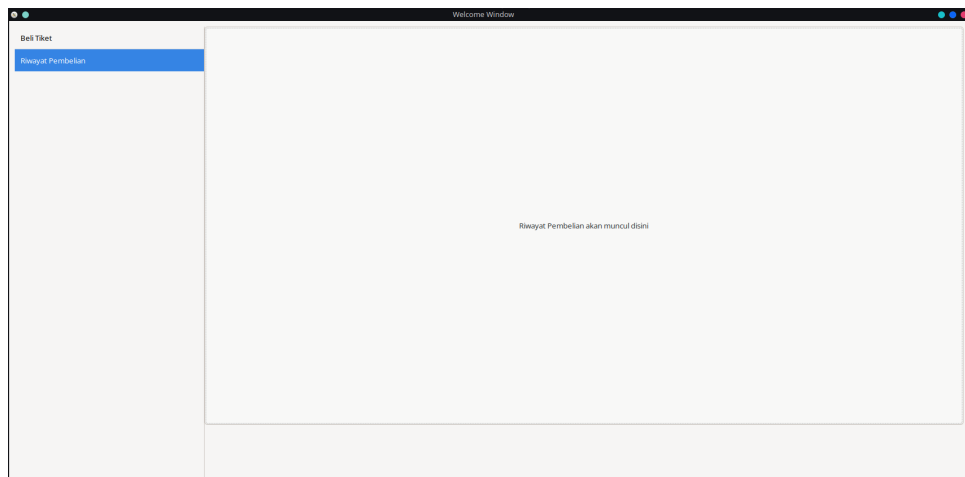


Figure 13: Halaman Pembelian Tiket

## 4 Kode Program

Kode program dari aplikasi kami lampirkan dalam folder myber beserta file myber-dev.zip, serta pada repository github berikut :

- <https://github.com/indonumberone/myber>

## 5 Analisa

### 5.1 Alur Program

Setelah melakukan kompilasi dan compiler menghasilkan sebuah file main. Selanjutnya adalah melakukan eksekusi file binary tersebut. Keseluruhan alur dari aplikasi ini adalah sebagai berikut.

#### 5.1.1 Welcome Screen

Welcome screen merupakan tampilan awal ketika menjalankan aplikasi ini.

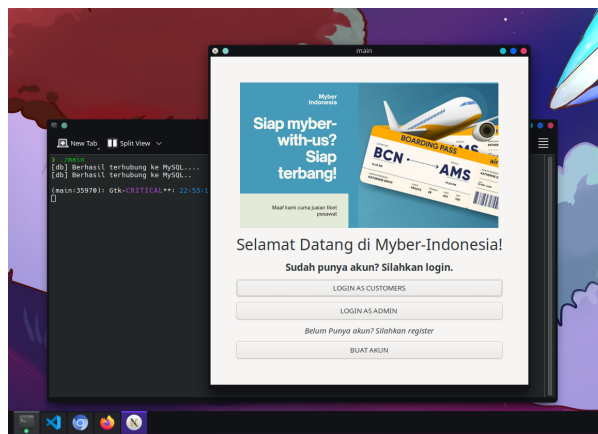


Figure 14: Welcome Screen

Pengguna dalam hal ini ada dua tipe pengguna yaitu Admin dan Customer. Dari sisi Customer, terdapat dua buah fungsi pada bagian ini yaitu untuk Register dan Login. Sedangkan dari sisi Admin hanya bisa melakukan login, karena kredensial Admin sudah dibuat secara terpisah pada database.

Kode dari halaman ini terdapat dalam fungsi utama yaitu main pada file main.c

```
115 int main(int argc, char *argv[])
116 {
117     // Create login button
118     login_customers_button = gtk_button_new_with_label("LOGIN AS CUSTOMERS");
119     g_signal_connect(G_OBJECT(login_customers_button), "clicked", G_CALLBACK(login_as_cu), NULL);
120     gtk_box_pack_start(GTK_BOX(vbox), login_customers_button, TRUE, TRUE, 0);
121     login_admin_button = gtk_button_new_with_label("LOGIN AS ADMIN");
122     g_signal_connect(G_OBJECT(login_admin_button), "clicked", G_CALLBACK(login_as_admin), NULL);
123     gtk_box_pack_start(GTK_BOX(vbox), login_admin_button, TRUE, TRUE, 0);
124
125     create_label = gtk_label_new("Belum Punya akun? Silahkan register");
126     gtk_box_pack_start(GTK_BOX(vbox), create_label, TRUE, TRUE, 0);
127     PangoAttrList *attr_list_info_lagi = pango_attr_list_new();
128     PangoAttribute *attr_size_info_lagi = pango_attr_size_new_absolute(15 * PANGO_SCALE);
129     PangoAttribute *attr_weight_info_lagi = pango_attr_style_new(PANGO_STYLE_ITALIC);
130     pango_attr_list_insert(attr_list_info_lagi, attr_size_info_lagi);
131     pango_attr_list_insert(attr_list_info_lagi, attr_weight_info_lagi);
132     gtk_label_set_attributes(GTK_LABEL(create_label), attr_list_info_lagi);
133     pango_attr_list_unref(attr_list_info_lagi);
134     gtk_box_pack_start(GTK_BOX(vbox), create_label, TRUE, TRUE, 0);
135
136     create_akun = gtk_button_new_with_label("BUAT AKUN");
137     g_signal_connect(G_OBJECT(create_akun), "clicked", G_CALLBACK(create_user_go), intin);
138     gtk_box_pack_start(GTK_BOX(vbox), create_akun, TRUE, TRUE, 0);
139
140     gtk_widget_show_all(intine_window);
141     gtk_main();
142
143     return 0;
144 }
```

Figure 15: Sebagian Kode Program untuk Welcome Screen

Kode tersebut memanggil beberapa entitas dari GTK seperti membuat window beserta memanggil GtkWidget yang merupakan base class dari widget yang ada pada GTK. Dari GtkWidget kemudian bisa dibuat beberapa elemen lain yang nantinya akan membentuk user interface dari aplikasi ini.

Contohnya adalah dalam pembuatan teks atau tulisan, teks dalam GTK disebut sebagai label. Untuk membuat label pertama yang dilakukan adalah memanggil GtkLabel lalu menggunakan Constructor new dari GtkLabel untuk membuat sebuah label baru.

```
username_label = gtk_label_new("Username:");
username_entry = gtk_entry_new();
gtk_box_pack_start(GTK_BOX(vbox), username_label, TRUE, TRUE, 0);
gtk_box_pack_start(GTK_BOX(vbox), username_entry, TRUE, TRUE, 0);

password_label = gtk_label_new("Password:");
password_entry = gtk_entry_new();
gtk_entry_set_visibility(GTK_ENTRY(password_entry), FALSE);
gtk_box_pack_start(GTK_BOX(vbox), password_label, TRUE, TRUE, 0);
gtk_box_pack_start(GTK_BOX(vbox), password_entry, TRUE, TRUE, 0);

login_button = gtk_button_new_with_label("Login");
g_signal_connect(G_OBJECT(login_button), "clicked", G_CALLBACK(login_process_user), customers_login_wi);
gtk_box_pack_start(GTK_BOX(vbox), login_button, TRUE, TRUE, 0);

back_button = gtk_button_new_with_label("Kembali ke halaman awal");
g_signal_connect(G_OBJECT(back_button), "clicked", G_CALLBACK(go_back_to_home), customers_login_window);
gtk_box_pack_start(GTK_BOX(vbox), back_button, TRUE, TRUE, 0);

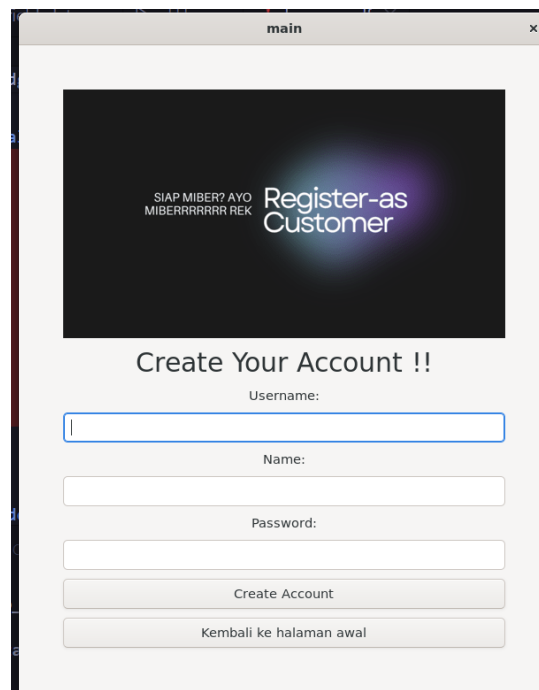
gtk_widget_show_all(customers_login_window);
```

**Figure 16:** GtkLabel

Sebagai contoh, diatas merupakan kode program dari fungsi `create_login_user_window()` Teks-teks seperti Username, Password, Login dan Kembali ke halaman awal dibuat dengan menggunakan GtkLabel. Dimana dalam membuat Constructor adalah dengan memanggil fungsi `gtk_label_new("isi_teks");`. Setelah itu adalah menampilkannya ke dalam window dengan memanggil fungsi `gtk_widget_show_all(window_saat_ini)` Dokumentasi dari GtkLabel yang lebih lengkap terdapat pada halaman <https://docs.gtk.org/gtk3/ctor.Widget.new.html>

## 5.2 Halaman Register

Pada Halaman Register digunakan user untuk membuat akun aplikasi MyBer.



**Figure 17:** Halaman Register

Pada halaman ini jika user ingin membuat sebuah akun maka wajib mengisi semua text input yang disediakan, jika user belum menginputkan semua text input yang digunakan maka akan alert atau warning yang muncul, seperti berikut ini:

main

SIAP MIBER? AYO MIBERRRRRR REK Register-as Customer

USERNAME WAJIB DIISI!!!

Username:

Name:

Password:

Create Account

Kembali ke halaman awal

**Figure 18:** Username kosong

Pada gambar diatas terjadi ketika user tidak menginputkan username

main

SIAP MIBER? AYO MIBERRRRRR REK Register-as Customer

NAMA WAJIB DIISI!!!

Username:

Name:

Password:

Create Account

Kembali ke halaman awal

**Figure 19:** nama kosong

Pada gambar diatas terjadi ketika user tidak menginputkan nama

**Figure 20:** password kosong

Pada gambar diatas terjadi ketika user tidak menginputkan password

**Figure 21:** username telah dipakai

Pada gambar diatas terjadi ketika username yang diinputkan oleh user telah dipakai oleh userlain.

Berikut ini adalah tampilan kode dari halaman register

```

void create_customer_register_window(GtkWidget *parent_window)
{
    // dikek 1 banner gambar
    image = gtk_image_new_from_file("assets/register_cust.png");
    gtk_box_pack_start(GTK_BOX(vbox), image, TRUE, TRUE, 0);

    banner_label = gtk_label_new("Create Your Account !!");
    PangoAttrList *attr_list = pango_attr_list_new();
    PangoAttribute *attr_size = pango_attr_size_new_absolute(30 * PANGO_SCALE);
    pango_attr_list_insert(attr_list, attr_size);
    gtk_label_set_attributes(GTK_LABEL(banner_label), attr_list);
    pango_attr_list_unref(attr_list);
    gtk_box_pack_start(GTK_BOX(vbox), banner_label, TRUE, TRUE, 0);

    username_label = gtk_label_new("Username:");
    username_entry = gtk_entry_new();
    gtk_box_pack_start(GTK_BOX(vbox), username_label, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(vbox), username_entry, TRUE, TRUE, 0);
    name_label = gtk_label_new("Name:");
    name_entry = gtk_entry_new();
    gtk_box_pack_start(GTK_BOX(vbox), name_label, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(vbox), name_entry, TRUE, TRUE, 0);
    password_label = gtk_label_new("Password:");
    password_entry = gtk_entry_new();
    gtk_entry_set_visibility(GTK_ENTRY(password_entry), FALSE);
    gtk_box_pack_start(GTK_BOX(vbox), password_label, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(vbox), password_entry, TRUE, TRUE, 0);

    login_button = gtk_button_new_with_label("Create Account");
    g_signal_connect(G_OBJECT(login_button), "clicked", G_CALLBACK(create_user), customer_register_window);
    gtk_box_pack_start(GTK_BOX(vbox), login_button, TRUE, TRUE, 0);

    back_button = gtk_button_new_with_label("Kembali ke halaman awal");
    g_signal_connect(G_OBJECT(back_button), "clicked", G_CALLBACK(go_back_to_home), customer_register_window);
    gtk_box_pack_start(GTK_BOX(vbox), back_button, TRUE, TRUE, 0);

    gtk_widget_show_all(customer_register_window);
}

```

**Figure 22:** Kode halaman register dari gtk

Dari kode diatas dibuat sebuah window bernama `customer_register_window()` yang digunakan sebagai window pada halaman register. kemudian dibuat sebuah box dengan nama vbox dengan orientasi vertikal yang dimasukan ke dalam container window, dibuat box dengan orientasi vertikal ini bertujuan agar setiap ada widget baru yang nantinya akan ditambahkan ke dalam vbox ini akan tertata secara vertikal menurun.

kemudian variabel yang berisi fungsi gtk entry new digunakan untuk mengambil teks yang diinputkan oleh user, kemudian jika user menekan button create akun maka fungsi callback dari button akan mengarahkan ke fungsi create user

```

main: create user(GtkWidget *gpointer data)
{
    void create_user(GtkWidget *widget, gpointer data)
    {
        const gchar *name = gtk_entry_get_text(GTK_ENTRY(name_entry));
        const gchar *password = gtk_entry_get_text(GTK_ENTRY(password_entry));
        printf("size e user %d", strlen(username));
        printf("size e nama %d", strlen(name));
        printf("size e pass %d", strlen(password));
        g_print("Username: %s\n", username);
        g_print("Name: %s\n", name);
        g_print("Password: %s\n", password);
        if (strlen(username) == 0) ...
        else if (strlen(name) == 0) ...
        else if (strlen(password) == 0) ...
        else
        {
            int hasile = register_user(username, name, password);
            if (hasile == 0)
            {
                gchar *alert_block = g_strdup_printf("Berhasil register, silahkan login!");
                gtk_label_set_text(GTK_LABEL(banner_label), alert_block);
                g_free(alert_block);
                create_login_user_window(current_window);
            }
            else if (hasile == 1)
            {
                gchar *alert_block = g_strdup_printf("Username sudah digunakan");
                gtk_label_set_text(GTK_LABEL(banner_label), alert_block);
                g_free(alert_block);
                gtk_widget_show_all(customer_register_window);
            }
        }
    }
}

```

**Figure 23:** Kode dari fungsi create user

Dari kode diatas username yang telah diinputkan akan di ambil dan disimpan dalam sebuah variabel seperti username, nama, password, dari variabel tersebut kemudian di cek apakah panjang dari variabel (variabel username, nama, password) sama dengan 0 jika sama dengan 0 maka akan mengganti nilai label yang awalnya berisi Create Your Account!!, pengecekan ini bertujuan agar user yang ingin membuat sebuah akun wajib mengisi text input yang disediakan, jika user telah mengisi semua text input yang digunakan maka variabel tadi akan dijadikan argumen untuk fungsi register user

```

int register_user(char *user, char *nama, char *pw)
{
    char query[256];

    sprintf(query, "INSERT INTO users (username, nama, password) VALUES ('%s', '%s', '%s')", user, nama, pw);
    if (mysql_query(conn, query))
    {
        fprintf(stderr, "mysql_query() failed: %s\n", mysql_error(conn));
        return 1;
    }
    return 0;
}

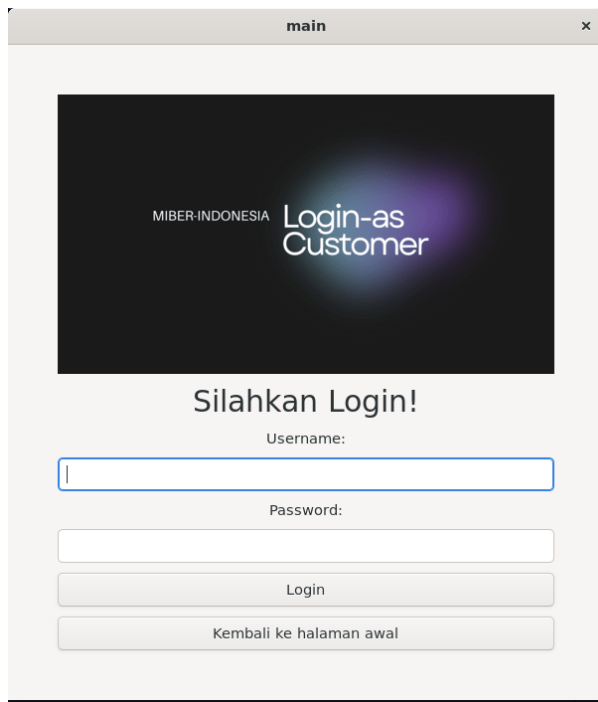
```

**Figure 24:** Kode untuk menginputkan data kedalam mysql

Dari kode diatas digunakan untuk menginputkan kedalam database mysql dengan menggunakan insert into, jika username ada yang sama akan mengembalikan 1 jika username tidak ada yang sama maka akan mengembalikan nilai 0 dari nilai diatas di kembalikan ke fungsi create user dan kemudian di handling jika pengembalian fungsi register user itu 1 maka akan username tersebut telah dipakai , jika 0 maka user berhasil mendaftar dan data berhasil disimpan dalam database mysql, dan akan diredirect ke halaman login.

### 5.3 Halaman Login User

Halaman Login User adalah halaman yang digunakan untuk user masuk kedalam akunya dan menuju ke dashboard tampilan user



**Figure 25:** Tampilan Awal dari Login User

Tampilan diatas adalah tampilan Login User, dari tampilan itu user disuruh untuk mengisi username dan password untuk masuk ke tampilan selanjutnya. Untuk melihat logic pada tampilan ini bekerja kita bisa melihat di dalam fungsi berikut ini.

```

void create_login_user_window(GtkWidget *parent_window)
{
    customers_login_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    g_signal_connect(customers_login_window, "destroy", G_CALLBACK(destroy), NULL);
    gtk_container_set_border_width(GTK_CONTAINER(customers_login_window), WINDOW_SIZE);
    vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 10);
    gtk_container_add(GTK_CONTAINER(customers_login_window), vbox);
    // dikeh i banner gambar
    image = gtk_image_new_from_file("assets/login_cust.png");
    gtk_box_pack_start(GTK_BOX(vbox), image, TRUE, TRUE, 0);

    banner_label = gtk_label_new("Silahkan Login!");
    PangoAttrList *attr_list = pango_attr_list_new();
    PangoAttribute *attr_size = pango_attr_size_new_absolute(30 * PANGO_SCALE);
    pango_attr_list_insert(attr_list, attr_size);
    gtk_label_set_attributes(GTK_LABEL(banner_label), attr_list);
    pango_attr_list_unref(attr_list);
    gtk_box_pack_start(GTK_BOX(vbox), banner_label, TRUE, TRUE, 0);

    username_label = gtk_label_new("Username:");
    username_entry = gtk_entry_new();
    gtk_box_pack_start(GTK_BOX(vbox), username_label, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(vbox), username_entry, TRUE, TRUE, 0);

    password_label = gtk_label_new("Password:");
    password_entry = gtk_entry_new();
    gtk_entry_set_visibility(GTK_ENTRY(password_entry), FALSE);
    gtk_box_pack_start(GTK_BOX(vbox), password_label, TRUE, TRUE, 0);
}

```

Figure 26: Kode fungsi login user pada gtk

Dari kode diatas `gtk_label_new()` digunakan untuk memberikan tampilan teks ke gtk window yang telah dibuat kemudian `gtk_entry_new()` digunakan untuk menangkap text input yang diinputkan oleh user. kemudian `gtk_button_new_with_label()` digunakan untuk membuat sebuah button dengan text ditengah tengah button. ketika button itu di klik maka fungsi callback dari gtk akan memanggil fungsi `login_process_user()`

```

void login_process_user(GtkWidget *widget, gpointer data)
{
    static int try = 3;

    GtkWidget *current_window = GTK_WIDGET(data);
    gtk_widget_hide(customers_login_window);

    const gchar *username = gtk_entry_get_text(GTK_ENTRY(username_entry));
    const gchar *password = gtk_entry_get_text(GTK_ENTRY(password_entry));
    UserDetails user_details;

    if (check_user(username) == 0) ...
    else
    {
        int hasil = login_user(username, password, &user_details);
        g_print("Username: %s\n", username);
        g_print("Password: %s\n", password);
        if (try < 1) ...
        else if (hasil == 0) ...
        else if (hasil == 1) ...

        printf("[[log] Login %d try %d\n", hasil, try);
    }
}

```

Figure 27: Kode dari login process user

Dari kode diatas terdapat sebuah variabel statis dengan tipe data integer yang bernilai 3, kemudian terdapat variabel yang digunakan untuk mendapatkan text input username dan password kemudian dari variabel variabel tersebut dijadikan argumen pada fungsi `check_user()`

```

int check_user(char *user)
{
    char query[256];
    sprintf(query, "SELECT username FROM users WHERE username = '%s'", user);

    if (mysql_query(conn, query))
    {
        fprintf(stderr, "mysql_query() failed: %s\n", mysql_error(conn));
        return -1;
    }

    res = mysql_store_result(conn);
    MYSQL_ROW row = mysql_fetch_row(res);
    int success = (row != NULL);
    mysql_free_result(res);
    return success;
}

```

Figure 28: fungsi check\_user()

fungsi `check_user()` ini digunakan untuk mengecek pada databases apakah ada



username sudah ada pada database belum. jika ada maka lanjut keproses dibawahnya terdapat sebuah fungsi `login_user()`

```
int login_user(char *user, char *pw, UserDetails *user_details)
{
    char query[256];
    sprintf(query, "SELECT username, name, admin FROM users WHERE username = '%s' AND password = '%s' AND blocked = 0");

    if (mysql_query(conn, query)) {
        return -1;
    }

    res = mysql_store_result(conn);
    MYSQL_ROW row = mysql_fetch_row(res);
    int success = (row != NULL);

    if (success)
    {
        strcpy(user_details->username, row[0]);
        strcpy(user_details->name, row[1]);
        user_details->is_admin = atoi(row[2]);
    }

    mysql_free_result(res);
    return success;
}
```

**Figure 29:** fungsi `login_user()`

fungsi `login_user()` membutuhkan 3 argumen, pada argumen yang ke satu dan dua digunakan untuk menginputkan username dan password kemudian untuk argumen ke 3 adalah sebuah struct yang digunakan nantinya untuk menyimpan username nama dan isAdmin. kemudian terdapat percabangan if didalam if yang pertama digunakan untuk mengecek apakah user ini keblokir atau tidak. jika ke blokir maka akan menampilkan alert pada GtkLabel kemudian kemudian pada else if dibawahnya mengecek jika hasil `== 0` jika iya berarti password yang dimasukanoleh si user salah dan try yang bernilai 3 akan di kurangi 1. kemudian pada else if dibawahnya digunaka untuk mengecek apakah hasl `==1` jika iya maka windows selanjunya akan di tampilkan.

```
int blocked(char *user)
{
    char query[256];
    sprintf(query, "UPDATE users SET blocked =1 ,blocked_until = CURRENT_TIMESTAMP() + INTERVAL 1 MINUTE WHERE username = '%s'");

    if (mysql_query(conn, query))
    {
        fprintf(stderr, "mysql_query() failed: %s\n", mysql_error(conn));
        mysql_close(conn);
        return -1;
    }

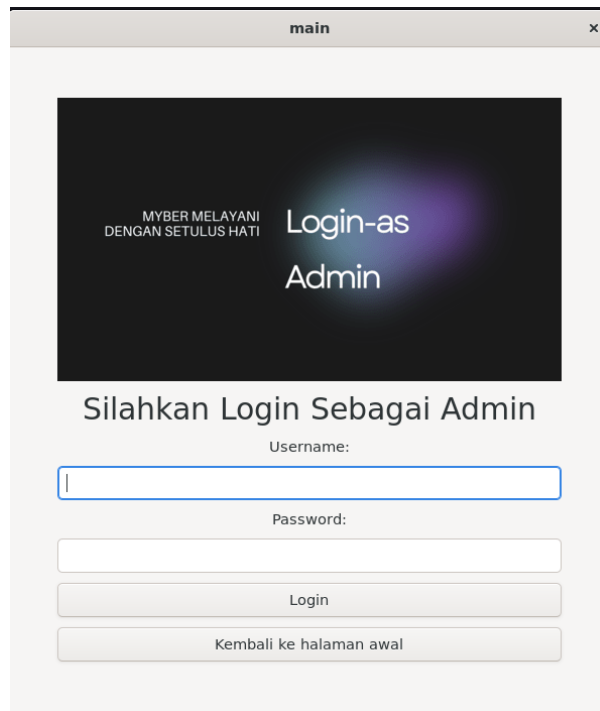
    return 0;
}
```

**Figure 30:** fungsi `blocked()`

fungsi `blocked()` digunakan untuk memblokir sementara jika user lebih dari 3x gagal dalam memasukan password.

## 5.4 Login Admin

Halaman ini adalah halaman untuk login admin



main x

MYBER MELAYANI  
DENGAN SETULUS HATI

Login-as  
Admin

Silahkan Login Sebagai Admin

Username:

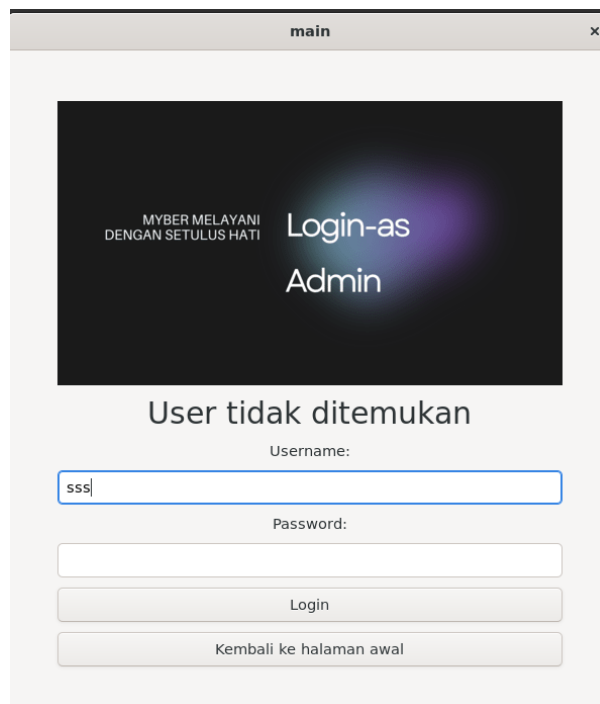
Password:

Login

Kembali ke halaman awal

**Figure 31:** Tampilan Login Admin

Pada Halaman login ini digunakan admin untuk login ke halaman dashboard admin. pada halaman ini admin diperintahkan untuk memasukkan username dan password untuk login jika username yang di inputkan salah maka akan menampilkan username/ password salah seperti berikut



main x

MYBER MELAYANI  
DENGAN SETULUS HATI

Login-as  
Admin

User tidak ditemukan

Username:

sss

Password:

Login

Kembali ke halaman awal

**Figure 32:** Username atau Password salah

```

64 void create_login_admin_window(GtkWidget *parent_window)
65 {
66     GtkWidget *vbox;
67     GtkWidget *username_label;
68     GtkWidget *password_label;
69     GtkWidget *login_button;
70     GtkWidget *back_button;
71     GtkWidget *image;
72
73     // printf("toplah %s\n", parent_window);
74     admin_login_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
75     g_signal_connect(admin_login_window, "destroy", G_CALLBACK(destroy), NULL);
76     gtk_container_set_border_width(GTK_CONTAINER(admin_login_window), WINDOW_SIZE);
77     vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 0);
78     gtk_container_add(GTK_CONTAINER(admin_login_window), vbox);
79     // dikeh i banner gambar
80     image = gtk_image_new_from_file("assets/login_admin.png");
81     gtk_box_pack_start(GTK_BOX(vbox), image, TRUE, TRUE, 0);
82
83     banner_label = gtk_label_new("Silahkan login Sebagai Admin");
84     PangoAttrList *attr_list = pango_attr_list_new();
85     PangoAttribute *attr_size = pango_attr_size_new_absolute(30 * PANGO_SCALE);
86     pango_attr_list_insert(attr_list, attr_size);
87     gtk_label_set_attributes(GTK_LABEL(banner_label), attr_list);
88     pango_attr_list_unref(attr_list);
89     gtk_box_pack_start(GTK_BOX(vbox), banner_label, TRUE, TRUE, 0);
90
91     username_label = gtk_label_new("Username:");
92     username_entry = gtk_entry_new();
93     gtk_box_pack_start(GTK_BOX(vbox), username_label, TRUE, TRUE, 0);

```

**Figure 33:** Tampilan Login Admin

Pada kode diatas adalah kode untuk membuat window untuk ditampilkan, cara kerjanya sama seperti login user yaitu membuat window terlebih dahulu kemudian membuat box vertikal kemudian box vertikal itu dimasukan kedalam window untuk menampilkan teks pada gtk dapat menggunakan fungsi dari gtk itu sendiri yaitu dengan fungsi `gtk_label_new()` kemudian untuk memberikan ruang teks input pada gtk dapat menggunakan fungsi `gtk_entry_new()`, kemudian untuk membuat sebuah button pada gtk dapat menggunakan fungsi `gtk_entry_new()` dari gtk widget yang sudah dibuat tadi kemudian dimasukan kedalam box vertikal tadi yang sudah dibuat untuk membuat button ketika ditekan melakukan sebuah tindakan dapat menggunakan sebuah call yang telah disediakan gtk, dalam call back ini akan mengarahkan kefungsi `login_process_admin`

```

104 void login_process_admin(GtkWidget *widget, gpointer data)
105 {
106     GtkWidget *current_window = GTK_WIDGET(data);
107
108     const gchar *username = gtk_entry_get_text(GTK_ENTRY(username_entry));
109     const gchar *password = gtk_entry_get_text(GTK_ENTRY(password_entry));
110
111     if (check_user(username) == 0)
112     {
113         gchar *alert_warning = g_strdup_printf("User tidak ditemukan");
114         gtk_label_set_text(GTK_LABEL(banner_label), alert_warning);
115         g_free(alert_warning);
116
117         gtk_widget_show_all(current_window);
118     }
119     else
120     {
121         int hasil = login_admin(username, password);
122         if (hasil == 0)
123         {
124             gtk_widget_show_all(admin_login_window);
125         }
126         else if (hasil == 1)
127         {
128             create_welcome_admin_window(admin_login_window);
129         }
130     }
131 }

```

**Figure 34:** fungsi `login_process_admin`

Fungsi ini adalah fungsi yang dipanggil oleh callback button login, dalam fungsi ini digunakan untuk masuk kedalam dashboard admin. didalam fungsi ini terdapat fungsi check user yang digunakan untuk mengecek username apakah sudah ada didalam database. jika ada maka fungsi `login_admin()` akan dieksekusi

```

int login_admin(char *user, char *pw)
{
    char query[256];
    sprintf(query, "SELECT username FROM users WHERE username = '%s' AND password = '%s' AND admin = 1", user, pw);

    if (mysql_query(conn, query))
    {
        fprintf(stderr, "mysql_query() failed: %s\n", mysql_error(conn));
        return -1;
    }

    res = mysql_store_result(conn);
    MYSQL_ROW row = mysql_fetch_row(res);
    int success = (row != NULL);
    mysql_free_result(res);
    return success;
}

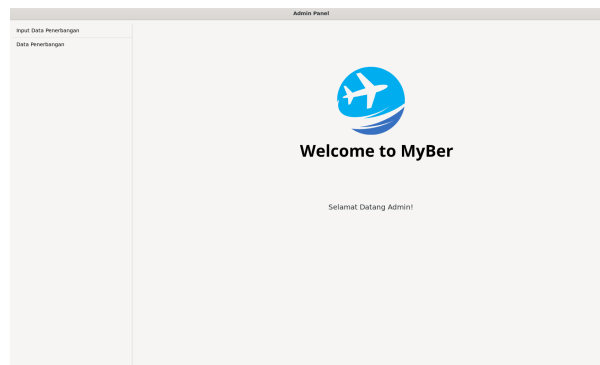
```

**Figure 35:** fungsi login\_admin()

Dalam fungsi login\_admin() ini digunakan untuk mencari username dan password serta isAdmin 1 dari database mysql jika pada database ada dan username password dan isAdmin bernilai 1 maka akan diarahkan ke dashboard admin

## 5.5 Dashboard Admin

Ini adalah tampilan dashboard admin



**Figure 36:** tampilan awal dashboard admin

Pada tampilan dashboard admin ini terdapat 2 menu pada side bar menu pertama adalah menu yang digunakan untuk menginput data kedalam mysql, berikut ini adalah tampilannya

**Figure 37:** tampilan input data

Pada tampilan ini digunakan oleh admin untuk melakukan input database penerbangan pesawat kemudian untuk menu kedua dari dashboard admin ini adalah menu data penerbangan, berikut ini adalah tampilannya

Admin Panel									
Input Data Penerbangan	No Penerbangan	Maskapai	Kelas	Jenis Keberangkatan	Tujuan	Tanggal Berangkat	Waktu Keberangkatan	Tanggal Kedatangan	Harga Tiket Pesawat
Data Penerbangan	0201	Garda Indonesia	Business	SBI	SLO	2024-08-06	08:00:00	2024-08-06	700000
	0202	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0203	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0204	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0205	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0206	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0207	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0208	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0209	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000
	0210	Garuda	Ekonomi	SBI	SLO	2024-08-06	08:00:00	2024-08-06	300000

Figure 38: tampilan data penerbangan

Pada tampilan ini digunakan untuk menampilkan data penerbangan yang sudah diinputkan kedalam database, untuk kode dari dashboard admin ini dapat dilihat berikut ini

```
void create_welcome_admin_window(GtkWidget *parent_window)
{
    GtkWidget *vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 2);
    GtkWidget *hbox = gtk_box_new(GTK_ORIENTATION_HORIZONTAL, 2);
    GtkWidget *input_data_page = create_input_data_page(welcome_window);
    GtkWidget *history_data_page = show_history_data(welcome_window);
    GtkWidget *stack = gtk_stack_new(GTK_ORIENTATION_VERTICAL, 0);
    gtk_stack_add_named(stack, input_data_page, "inputdata", "Input Data Penerbangan");
    gtk_stack_add_named(stack, history_data_page, "historydata", "Data Penerbangan");
    GtkWidget *welcome_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(welcome_window, "Admin Panel");
    gtk_window_set_default_size(welcome_window, 800, 600);
    gtk_container_add(GTK_CONTAINER(welcome_window), vbox);
    gtk_box_pack_start(GTK_BOX(vbox), hbox, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(hbox), stack, TRUE, TRUE, 0);
    gtk_widget_show_all(welcome_window);
}
```

Figure 39: kode dari dashboard admin

Pada kode diatas window yang telah dibuat kemudian dimasukan sebuah grid, dalam grid tersebut mempunyai 2 widget yaitu sidebar dan stack , kemudian di sidebar terdapat 2 menu yaitu input data penerbangan dan data penerbangan pada menu input data penerbangan ketika di klik akan menampilkan sebuah fungsi widget yaitu create\_input\_data() untuk isi dari fungsi widget ini bisa dilihat dalam gambar dibawah ini.

```
GtkWidget *create_input_data_page(GtkWidget *parent_window)
{
    GtkWidget *vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 2);
    GtkWidget *hbox = gtk_box_new(GTK_ORIENTATION_HORIZONTAL, 2);
    GtkWidget *stack = gtk_stack_new(GTK_ORIENTATION_VERTICAL, 0);
    GtkWidget *input_data_page = create_input_data_page(welcome_window);
    GtkWidget *history_data_page = show_history_data(welcome_window);
    GtkWidget *stack = gtk_stack_new(GTK_ORIENTATION_VERTICAL, 0);
    gtk_stack_add_named(stack, input_data_page, "inputdata", "Input Data Penerbangan");
    gtk_stack_add_named(stack, history_data_page, "historydata", "Data Penerbangan");
    GtkWidget *welcome_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(welcome_window, "Admin Panel");
    gtk_window_set_default_size(welcome_window, 800, 600);
    gtk_container_add(GTK_CONTAINER(welcome_window), vbox);
    gtk_box_pack_start(GTK_BOX(vbox), hbox, TRUE, TRUE, 0);
    gtk_box_pack_start(GTK_BOX(hbox), stack, TRUE, TRUE, 0);
    gtk_widget_show_all(welcome_window);
}
```

Figure 40: widget input data

```
void onSaveButtonClicked(GtkWidget *button, gpointer data) {
    const gchar *no_penerbangan = gtk_entry_get_text(GTK_ENTRY(no_penerbangan_pesawat_entry));
    const gchar *nama_asal = gtk_entry_get_text(GTK_ENTRY(nama_asal_entry));
    const gchar *nama_tujuan = gtk_entry_get_text(GTK_ENTRY(nama_tujuan_entry));
    const gchar *jadwal_berangkat = gtk_button_get_label(GTK_BUTTON(jadwal_keberangkatan_button));
    const gchar *jadwal_datang = gtk_button_get_label(GTK_BUTTON(jadwal_kedatangan_button));
    const gchar *hargaanya = gtk_entry_get_text(GTK_ENTRY(harga_entry));
    gint jam_berangkat = gtk_spin_button_get_value_as_int(GTK_SPIN_BUTTON(jam_keberangkatan));
    gint menit_berangkat = gtk_spin_button_get_value_as_int(GTK_SPIN_BUTTON(minit_keberangkatan));
    gint jam_kedatangannya = gtk_spin_button_get_value_as_int(GTK_SPIN_BUTTON(jam_kedatangan));
    gint menit_kedatangannya = gtk_spin_button_get_value_as_int(GTK_SPIN_BUTTON(minit_kedatangan));
    g_print("Value no %d\t%t\tiki\n", no_penerbangan);
    char jam_keberangkatan[100];
    sprintf(jam_keberangkatan, "%02d:%02d:00", jam_berangkat, menit_berangkat);
    char jam_datang[100];
    sprintf(jam_datang, "%02d:%02d:00", jam_kedatangannya, menit_kedatangannya);
    printf("nilai ne %d\tV\n", strlen(jam_keberangkatan));
    if (strlen(no_penerbangan) == 0) ...
    else if (strlen(entry_maskapai) == 0) ...
    else if (strlen(entry_kelas) == 0) ...
    else if (strlen(nama_asal) == 0) ...
    else if (strlen(nama_tujuan) == 0) ...
    else if (strlen(jadwal_berangkat) == 26) ...
    else if (strlen(jadwal_datang) == 0) ...
    else if (strlen(hargaanya) == 0) ...
    else
    {
        insert_flight_data(no_penerbangan, entry_maskapai, entry_kelas, nama_asal, nama_tujuan, jadwal_berangkat, ja
            _g_print("sukses");
    }
}
```

ini digunakan untuk mengambil data dari inputan  
menginput semua inputan yang dibutuhkan kan ji  
akan di tampilkan, jika user sudah menginputk  
h\_data akan di eksekusi, untuk isi dari fungsi  
gambar dibawah ini

```
void insert_flight_data(char *no_pener, char *maskapai, char *kelas, char *asal, char *tujuan, char *tgl_berangkat, char *tgl_kembali)
{
    char query[256];
    sprintf(query, "INSERT INTO data_penerbangan values(NULL, '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s');",
        no_pener, maskapai, kelas, asal, tujuan, tgl_berangkat, tgl_kembali, " ");
    if (mysql_query(conn, query))
    {
        fprintf(stderr, "mysql_query() failed: %s\n", mysql_error(conn));
    }
}
```

gunakan untuk menginputkan data dari gtk ke database. kemudian dari menu dashboard untuk menampilkan data penerbangan yang sudah diinputkan sebelumnya. nah ini

```

gtkwidget *show_history_data(GtkWidget *parent_windows)
{
    model = gtk_list_store_new(N_COLUMNS,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_STRING,
        G_TYPE_UINT,
        G_TYPE_UINT,
        G_TYPE_STRING,
        G_TYPE_STRING);

    int num_rows = get_jumlah_penerbangan();
    printf("Jd", num_rows);
    FlightDetails flight_details[num_rows];

    int res = get_data_penerbangan(flight_details);

    for (int i = 0; i < num_rows; i++)
    {
        gtk_list_store_insert_with_values(model, NULL, -1,
            FILE_NAME, flight_details[i].no_penerbangan,
            FILE_MASKAPAI, flight_details[i].maskapai,
            FILE_KELAS, flight_details[i].kelas,
            FILE_ASAL, flight_details[i].asal,
            FILE_TUJUAN, flight_details[i].tujuan,
            FILE_TGL_BERANGKAT, flight_details[i].tgl_berangkat

```

Figure 43: fungsi menu show data penerbangan

pada kode diatas window yang sudah dibuat kemudian dimasukan gtk list yang berisi kolom dan baris, untuk nilai dari baris sendiri didapatkan dari mengambil data dari mysql dari fungsi get\_data\_penerbangan isi dari fungsi get\_data\_penerbangan dapat dilihat pada gambar dibawah ini.

```

int get_data_penerbangan(FlightDetails *flight_details)
{
    num_fields = mysql_num_fields(res);
    num_rows = mysql_num_rows(res);

    for (int j = 0; j < num_rows; j++)
    {
        while ((row = mysql_fetch_row(res)) & (error-type) *res
            {
                row = mysql_fetch_row(res);
                // count = 0;

                for (i = 0; i < num_fields; i++)
                {
                    flight_details[j].no_id = atoi(row[0]) ? row[0] : 0;
                    strcpy(flight_details[j].no_penerbangan, row[1] ? row[1] : "NULL");
                    strcpy(flight_details[j].maskapai, row[2] ? row[2] : "NULL");
                    strcpy(flight_details[j].kelas, row[3] ? row[3] : "NULL");
                    strcpy(flight_details[j].asal, row[4] ? row[4] : "NULL");
                    strcpy(flight_details[j].tujuan, row[5] ? row[5] : "NULL");
                    strcpy(flight_details[j].tgl_berangkat, row[6] ? row[6] : "NULL");
                    strcpy(flight_details[j].waktu_keberangkatan, row[7] ? row[7] : "NULL");
                    strcpy(flight_details[j].tgl_datang, row[8] ? row[8] : "NULL");
                    strcpy(flight_details[j].waktu_kedatangan, row[9] ? row[9] : "NULL");
                    flight_details[j].harga = atoi(row[10]) ? row[10] : 0;

                    // printf("Jd", row[i] ? row[i] : "NULL");
                }
                printf("\n");
            }
    }
}

```

Figure 44: fungsi get\_data\_penerbangan

Pada kode diatas terdapat parameter struct flightDetails dari struct tersebut digunakan untuk menyimpan data dari database mysql. dan kemudian data dari mysql itu akan dirender dan di tampilkan pada menu ke 2 show data penerbangan

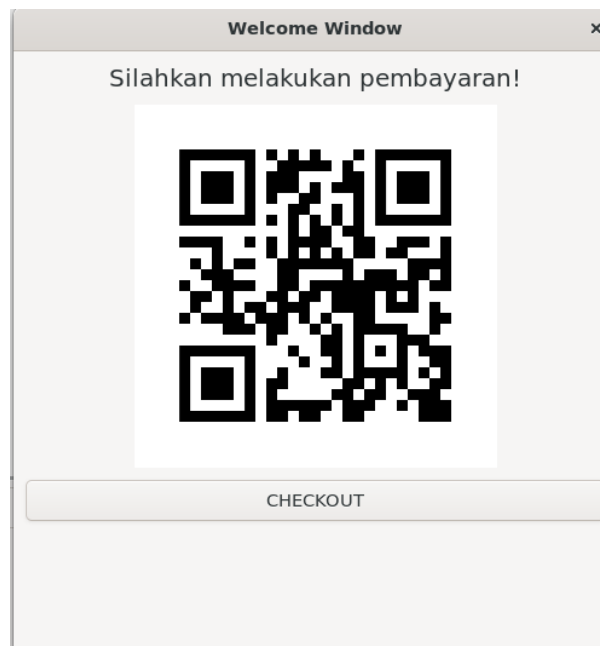
## 5.6 Dashboard User

Ini adalah tampilan dashboard users

Welcome Welcome										
No	Penumpang	Maskapai	Kelas	Jumlah Keberangkatan	Tujuan	Tanggal Keberangkatan	Waktu Keberangkatan	Tanggal Kedatangan	Waktu Kedatangan	Harga Tiket Pesawat
0003	Garuda Indonesia	Business	3Y	SLQ	2024-06-06	06:00:00	2024-06-06	09:00:00	700000	
P101	Garuda Indonesia	Business	3Y	SLQ	2024-06-06	06:00:00	2024-06-06	09:00:00	800000	
P105	Garuda Indonesia	Business	3Y	SLQ	2024-06-06	06:00:00	2024-06-06	09:00:00	800000	
P102	Garuda Indonesia	Business	3Y	SLQ	2024-06-06	06:00:00	2024-06-06	09:00:00	800000	
PA	Garuda Indonesia	Business	3Y	LP	2024-06-18	10:00:00	2024-06-20	23:00:00	7000000	
TOTAL										
CT121	Alhuda	Business	3Y	LP	2024-06-18	10:00:00	2024-06-20	23:00:00	7000000	
W0001	Wings Air	Business	3Y	LP	2024-06-18	10:00:00	2024-06-20	23:00:00	7000000	
GA121	Garuda Indonesia	Business	3Y	LP	2024-06-18	10:00:00	2024-07-01	00:14:00	1200000	

Figure 45: tampilan dashboard user

pada menu dashboard user tersedia menu yang digunakan untuk pembelian tiket pesawat apabila tiket itu dibeli maka akan muncul tampilan checkout seperti berikut ini.



**Figure 46:** tampilan checkout

Dari tampilan ini apabila button checkout ditekan maka akan menghasilkan tampilan seperti ini.



**Figure 47:** tampilan tiket

dari tampilan ini akan mencetak tiket dan nama username, untuk kode dari dashboard ini dapat dilihat dari kode berikut ini.



```

GtkWidget *show_pembelian_data(GtkWidget *parent_window)
{
    GtkWidget *show_pembelian_data;
    GtkWidget *gtk_cell_renderer_text_new();
    GtkWidget *gtk_tree_view_append_column(GTK_TREE_VIEW(view), column);

    column = gtk_tree_view_column_new_with_attributes("Harga Tiket Pesawat",
    gtk_cell_renderer_text_new(),
    "text", FILE_WAKTU_T,
    "background", COLOR,
    NULL);
    gtk_tree_view_append_column(GTK_TREE_VIEW(view), column);

    GtkWidget *button;
    button = gtk_button_new_with_label("BELI TICKET");
    g_signal_connect(button, "clicked", G_CALLBACK(on_button_clicked), NULL);

    GtkTreeSelection *selection;
    selection = gtk_tree_view_get_selection(GTK_TREE_VIEW(view));
    g_signal_connect(selection, "changed", G_CALLBACK(on_selection_changed), NULL);

    GtkWidget *scrollview = gtk_scrolled_window_new(NULL, NULL);
    gtk_container_add(GTK_CONTAINER(scrollview), view);
    // gtk_container_add(GTK_CONTAINER(scrollview), view);
    gtk_widget_set_hexpand(scrollview, TRUE);
    gtk_widget_set_vexpand(scrollview, TRUE);
    gtk_container_add(GTK_CONTAINER(show_history_page), scrollview);
    gtk_container_add(GTK_CONTAINER(show_history_page), button);
}

```

Figure 48: window show pembelian

Dari kode diatas data dari database mysql kemudian diambil dan di renderke widget, proses dari pengambilan data ini sama dengan proses show data penerbangan pada menu admin, yang membedakan disini terdapat tombol beli tike yang mempunyai sebuah callback dengan fungsi `on_button_clicked` untuk kode dari fungsi ini dapat dilihat dari gambar berikut ini.

```

void on_button_clicked(GtkButton *button, gpointer data)
{
    g_print("waktu keberangkatan: %s\n", waktu_keberangh);
    g_print("Tanggal Datang: %s\n", tgl_datang);
    g_print("waktu Kedatangan: %s\n", waktu_kedatangan);
    g_print("Harga: %s\n", harga);

    // g_free(no_penerbangan);
    g_free(maskapai);
    g_free(kelas);
    g_free(asal);
    g_free(tujuan);
    g_free(tgl_keberangkatan);
    g_free(waktu_keberangkatan);
    g_free(tgl_datang);
    g_free(waktu_kedatangan);

    char nomor_penerbangan[256];
    sprintf(nomor_penerbangan, "%s", no_penerbangan);

    create_ticket(current_user.name, nomor_penerbangan);
    create_orders(current_user.name);
}

gtk_box_pack_start(GTK_BOX(vbox), banner_label, FALSE, FALSE, 0);
gtk_box_pack_start(GTK_BOX(vbox), image, FALSE, FALSE, 0);
gtk_box_pack_start(GTK_BOX(vbox), button_checkout, FALSE, FALSE, 0);
gtk_widget_show_all(orders_window);
}

```

Figure 49: fungsi on\_button\_clicked

Dari kode diatas data penerbangan yang dipilih oleh user kemudian di proses dan dimasukan kedalam database,setelah dimasukan kedatabase kemudian diarahkan ke cetak tiket, untuk kodenya dapat dilihat pada gambar dibawah ini

```

void on_checkout_clicked(GtkWidget *widget, gpointer data)
{
    GtkWidget *current_window = GTK_WIDGET(data);
    gtk_widget_hide(current_window);

    tickets_window = gtk_window_new(GTK_WINDOW_TOPLEVEL);
    gtk_window_set_title(GTK_WINDOW(tickets_window), "Welcome Window");
    gtk_window_set_default_size(GTK_WINDOW(tickets_window), 500, 500);
    gtk_container_set_border_width(GTK_CONTAINER(tickets_window), 10);
    // GtkWidget *vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 2);

    GtkWidget *vbox = gtk_box_new(GTK_ORIENTATION_VERTICAL, 10);
    gtk_container_add(GTK_CONTAINER(tickets_window), vbox);

    char notif_order[256];
    sprintf(notif_order, "Silahkan Cetak Tiket Anda!");
    banner_label = gtk_label_new(NULL);
    gtk_label_set_text(GTK_LABEL(banner_label), notif_order);
    PangoAttrList *attr_list = pango_attr_list_new();
    PangoAttribute *attr_size = pango_attr_size_new_absolute(20 * PANGO_SCALE);
    pango_attr_list_insert(attr_list, attr_size);
    gtk_label_set_attributes(GTK_LABEL(banner_label), attr_list);
    pango_attr_list_unref(attr_list);

    GtkWidget *image = gtk_image_new_from_file("assets/tickets.jpg");

    char detail_tiket[256];
    GtkWidget *ticket_label;
    sprintf(detail_tiket, "A/n: %s", current_user.name);
    ticket_label = gtk_label_new(NULL);
    gtk_label_set_text(GTK_LABEL(ticket_label), detail_tiket);
}

```

Figure 50: fungsi on\_checkout\_clicked

pada fungsi `on_checkout_clicked` digunakan untuk mencetak tiket yang telah dibuat dan menampilkan nama user yang membeli

## 6 Kesimpulan

Keseluruhan alur dari program ini merupakan sebuah manajemen data tiket pesawat oleh dua tipe pengguna. Dari sisi Admin bertugas sebagai orang yang menginput data penjualan tiket. Sedangkan dari sisi Customer yang akan membeli tiket yang telah diinputkan oleh Admin sebelumnya. Pada mekanisme Login, Customer bisa melakukan Register dan Login sedangkan Admin hanya bisa melakukan Login karena kredensial dari Admin sendiri sudah dibuatkan dari database. Selain itu, terdapat mekanisme untuk melakukan blocking atau ban sementara apabila pengguna salah memasukkan password-nya lebih dari 3 kali, hal ini dilakukan untuk mengantisipasi adanya fraud atau spam. Lalu terdapat error handling yang lain seperti ketika memasukkan User yang belum ada.