**LAPORAN TUGAS AKHIR MATA KULIAH**

**PRAKTIKUM ALGORITMA DAN PEMROGRAMAN**

****

Oleh:

I Ketut Santa Wijaya (2008561028)

Hans Rio Alfredo Palla (2008561030)

I Gusti Ngurah Bagus Arimbawa (2008561032)

Ngakan Putu Widyasprana (2008561033)

Nanda Nikola (2008561037)

Dosen Pengampu:

Dr. Ngurah Agus Sanjaya ER, S.Kom., M.Kom.

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS UDAYANA**

**2021**

# KATA PENGANTAR

Puja dan puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Esa, karena atas limpahan berkat dan rahmat-Nya sajalah laporan dengan judul Laporan Tugas Akhir Mata Kuliah Praktikum Algoritma Dan Pemrograman ini dapat diselesaikan sesuai dengan waktu yang sudah ditentukan.

Laporan kegiatan ini disusun guna melengkapi selesainya kegiatan Tugas Akhir pada mata kuliah Algoritma dan Pemrograman dalam Program Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Udayana. Selain itu, penulis juga menyampaikan ucapan terima kasih sebesar-besarnya kepada Dr. Ngurah Agus Sanjaya ER, S.Kom., M.Kom. selaku dosen mata kuliah Algoritma dan Pemrogram sekaligus penanggung jawab dalam kegiatan tugas akhir ini, selain itu penulis juga ingin menyampaikan rasa terima kasih khususnya kepada kakak tingkat selaku asisten dosen dan juga penyelanggara dalam kegiatan Tugas Akhir ini.

Penulis menyadari laporan ini masih jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan penulis terima demi kesempurnaan tugas laporan ini. Penulis juga berharap semoga penugsan serta laporan ini dapat bermanfaat bagi setiap para pembaca serta bisa menjadi sumber belajar bagi orang lain juga. Untuk perhatiannya, penulis ucapkan terima kasih yang sebesar-besarnya.

Denpasar, 24 Mei 2021

Penulis

# 

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc72852403)

[DAFTAR ISI ii](#_Toc72852404)

[DAFTAR GAMBAR iv](#_Toc72852405)

[DAFTAR TABEL v](#_Toc72852406)

BAB I [PENDAHULUAN 1](#_Toc72852408)

[1.1 Latar Belakang 1](#_Toc72852409)

BAB II [LANDASAN TEORI 2](#_Toc72852411)

[2.1 Bahasa pemrograman C 2](#_Toc72852412)

[2.2 Tipe Data 3](#_Toc72852413)

[2.3 Control flow 3](#_Toc72852414)

[2.4 Pointer 4](#_Toc72852415)

[2.5 Operasi string 5](#_Toc72852416)

[2.6 Operasi file 6](#_Toc72852417)

[2.7 Array atau larik 7](#_Toc72852418)

BAB III [DESAIN DAN METODE 8](#_Toc72852420)

[3.1 Diagram alir 8](#_Toc72852421)

[3.2 Metode 8](#_Toc72852422)

[3.2.1 Struktur Dasar bahasa pemrograman C 9](#_Toc72852423)

[3.2.2 Penyeleksian Kondisi dan Perulangan 10](#_Toc72852424)

[3.2.3 Array 11](#_Toc72852425)

[3.2.4 Pointer, Fungsi dan Operasi File 12](#_Toc72852426)

BAB IV [HASIL DAN IMPLEMENTASI 15](#_Toc72852428)

[4.1 Sintaks program 15](#_Toc72852429)

[4.2 Contoh output program 25](#_Toc72852430)

[4.3 Penjelasan singkat program 28](#_Toc72852431)

BAB V [PENUTUP 29](#_Toc72852433)

[5.1 Kesimpulan 29](#_Toc72852434)

[5.2 Saran 30](#_Toc72852435)

[DAFTAR PUSTAKA 31](#_Toc72852436)

# DAFTAR GAMBAR

[Gambar 3. 1 Diagram alir program 8](#_Toc72852382)

[Gambar 3. 2 Pengunaan *file header* 9](#_Toc72852383)

[Gambar 3. 3 Fungsi utama atau *main* 9](#_Toc72852384)

[Gambar 3. 4 Pengunaan perulangan *while* 11](#_Toc72852385)

[Gambar 3. 5 Penggunaan *array* 12](#_Toc72852386)

[Gambar 4. 1 Tampilan menu utama 25](#_Toc72852387)

[Gambar 4. 2 Tampilan sesi input 25](#_Toc72852388)

[Gambar 4. 3 Tampilan menu admin 26](#_Toc72852389)

[Gambar 4. 4 Menu admin pilihan 1 26](#_Toc72852390)

[Gambar 4. 5 Tampilan pengguna membeli tiket 27](#_Toc72852391)

[Gambar 4. 6 Tampilan proses pengguna membeli tiket 27](#_Toc72852392)

# 

# DAFTAR TABEL

[Tabel 4. 1 Sintaks program 25](#_Toc72852360)

# BAB I

# PENDAHULUAN

## Latar Belakang

Seiring dengan berkembangnya kehidupan manusia yang semakin kompleks, teknologi komputer pun juga semakin mengembangkan sayapnya ke berbagai banyak hal, mulai dari kepentingan komersial, hiburan, pemerintahan bahkan kepentingan militer. Hal ini ditandai dengan semakin banyaknya teknologi seperti aplikasi atau program-program komputer yang digunakan. Kalau misalkan seseorang mencari atau sekedar berselancar di internet, disana bisa ditemukan banyak sekali aplikasi-aplikasi seperti aplikasi modifikasi data maupun *video game*.

Kemudian, jika membahas tentang pembuatan sebuah aplikasi ataupun program-program komputer, pembahasannya tidak akan lepas dari yang namanya algoritma, data digital serta pemrograman. Ada begitu banyak metode, konsep-konsep serta materi pembelajaran yang bisa dipelajari untuk mencapai titik pemrograrman yang lebih tinggi. Selain itu, juga banyak disediakan pilihan teknologi ataupun bahasa pemrograman yang bisa dipilih sesuai dengan selera masing-masing. Namun tetap, kreativitas adalah hal penting yang dibutuhkan jika dikaitkan dengan kegiatan produksi.

Karena itulah, kegiatan Tugas Akhir ini dilaksanakan guna melatih mahasiswa untuk dapat lebih lagi mengasah kemampuan mereka serta menggunakkan sisi kreativitas mereka dalam membuat suatu aplikasi program yang bisa bermanfaat bagi seseorang ataupun banyak orang. Seperti yang sudah dikatakan sebelumnya, bahwa program yang dibuat bisa saja berhubungan dengan kepentingan komersial ataupun bisnis. Dan salah satu contoh yang bisa diambil adalah manajemen tiket, entah itu tiket bioskop, pameran, wahana dan masih banyak lagi. Hal ini yang menginspirasi penulis untuk mengambil permasalah tersebut menjadi ide di dalam kegiatan Tugas Akhir ini.

# BAB II

# LANDASAN TEORI

## 2.1 Bahasa pemrograman C

Membahas tentang program ataupun pemrograman pasti tidak lepas dari yang namanya bahasa pemrograman. Bahasa pemrograman sendiri berfungsi sama seperti bahasa pada umumnya, yaitu untuk berkomunikasi. Namun bahasa pemrograman dibuat untuk memudahkan interaksi antara komputer dengan manusia, agar komputer dapat menjalankan instruksi yang tepat dan sesuai dengan keinginan dari sang manusia.

Ada begitu banyak jenis bahasa pemrograman yang tersebar ke seluruh dunia. Setiap bahasa pemrograman memiliki karakter dan sintaks uniknya masing-masing, kelebihan dan kekurangannya masing-masing, serta tujuan khusus dari bahasa pemrograman tersebut. Dan salah satu bahasa pemrograman yang sudah dikenal banyak orang adalah bahasa pemrograman C.

Meskipun bahasa pemrograman yang satu ini sudah tidak terlalu tenar penggunaannya dalam bidang-bidang professional seperti kepentingan komersial atau pembuatan aplikasi modern. Namun bahasa pemrograman C masih sering digunakan sebagai pos pertama belajar tentang pemrograman sebelum mempelajari bahasa pemrograman yang lebih kompleks. Hal ini sejalan dengan begitu banyaknya bahasa pemrograman lain yang ternyata bahasa pemrograman tersebut merupakan “keturunan” dari bahasa pemrograman C sendiri, diantaranya seperti bahasa pemrograman C++, C#, PHP maupun Python.

Bahasa Pemrograman C termasuk ke dalam Bahasa Tingkat-Menengah, semakin tinggi tingkat suatu bahasa, maka akan semakin mudah dipelajari oleh Manusia. Bahasa Pemrograman C diciptakan oleh seorang bernama Dennis Ritchie pada tahun 1972. Bahasa Pemrograman C bisa digunakan untuk membuat berbagai jenis aplikasi (*general-purpose programming language*), tapi Bahasa Pemrograman C paling cocok digunakan untuk membuat aplikasi yang berhubungan dengan Sistem Operasi dan Perangkat Keras, salah satu contohnya adalah Kernel-kernel pada Linux yang banyak dibangun menggunakkan bahasa pemrograman C.

## 2.2 Tipe Data

Tipe Data digunakan untuk memberitahu sebuah *compiler* bagaimana sebuah nilai dalam suatu variabel ataupun fungsi diproses. Dalam bahasa pemrograman C sendiri ada banyak jenis tipe data. Tipe data dalam bahasa pemrograman C dibagi menjadi 4, yaitu tipe data dasar*,* tipe data turunan*,* tipe data bentukan (*enum*) dan tipe data *void*.

Untuk yang tipe data dasar,dibagi lagi menjadi tipe integer dan tipe *float*. Tipe data integer terdiri dari *char, unsigned char, signed char, int, unsigned int, short, unsigned short, long* dan *unsigned long*. Sedangkan tipe data *float* terdiri dari *float, double* dan *long double*. Selain tipe data dasar, terdapat juga tipe data turunan.

Tipe data Turunan adalah tipe data dasar yang digabungkan atau dikelompokkan. Hanya ada 3 tipe data turunan di dalam bahasa pemrograman C, yatu *Array*¸ *Structure/ Struct* biasa dikenal dengan *record* dan terakhir tipe data *Pointer.* Tipe data *enum* atau bentukan adalah tipe data yang dibuat sendiri oleh seorang programmer, sedangkan tipe data *void* adalah tipe data yang memberitahukan bahwa tidak ada data, tipe data yang satu ini biasanya digunakandalam sebuah fungsi yang tidak memerlukan sebuah nilai *return*.

## 2.3 Control flow

*Control flow* jika diartikan ke dalam bahasa Indonesia berarti mengontrol alur atau aliran. Alur atau aliran yang dikontrol bukanlah aliran air ataupun aliran benda secara fisik, tapi aliran atau alur yang dimaksud adalah bagaimana cara berjalannya suatu program atau aplikasi. *Control flow* sendiri dibagi menjadi dua tipe lagi yaitu percabangan dan perulangan.

Percabangan dalam dunia pemrograman artinya memilih cabang-cabang tertentu dalam suatu program, jadi hasil akhir program akan berbeda jika cabang yang dipilih juga berbeda. Dalam penentuan atau pemilihan cabang sendiri, komputer memerlukan sebuah kondisi yang ditaruh di setiap cabang, apabila suatu kondisi tertentu terpenuhi maka alur program di dalamnya yang akan dijalankan.

Percabangan sendiri dibutuhkan karena komputer tidak bisa semata-mata melakukan sebuah pertimbangan seperti pikiran manusia. Kemudian pada umumnya, terdapat tiga jenis percabangan, yaitu percabangan *if*, *if else* dan *if else-if*. Tentu saja tiap jenis percabangan tersebut memiliki karakter uniknya masing-masing, dan suatu percabangan tentu saja akan lebih baik dari percabangan yang lain, tergantung dari permasalahan yang ada.

Setelah membahas singkat tentang percabangan, sekarang beralih ke perulangan. Perulangan adalah sebuah metode yang digunakan untuk mengulang suatu perintah dengan otomatis. Perulangan dibutuhkan karena jika perulangan sudah mencapai jumlah yang bisa dibilang masif, maka manusia akan kelelahan jika menuliskan perintah satu-satu persatu, karena itulah otomatisasi berupa perulangan ini sangatlah diperlukan. Terdapat tiga jenis perulangan yang sangat umum digunakan, yaitu perulangan *for*, *while* dan *do-while*.

## 2.4 Pointer

Dalam sebuah program, pasti tidak lepas dengan yang namanya variabel dan tipe data. Kemudian disaat seseorang membuat sebuah variabel, maka komputer akan meminta tempat di dalam memori untuk menempatkan data yang ada di dalam variabel tersebut.

Kemudian ada variabel *pointer* atau artinya penunjuk. Jika pada umumnya sebuah variabel menyimpan tipe-tipe data seperti angka, huruf ataupun karakter lainnya. Variabel *pointer* ditujukan untuk menyimpan alamat dari variabel lain yang ada di dalam program tersebut. Memang disaat mula-mula mengenal *pointer*, pengguna akan dipertemukan berbagai macam kebingungan dari karakter pengunaan *pointer* ataupun pekerjaan apa yang cocok dengan *pointer* itu sendiri.

Namun dalam bahasa pemrograman C, *pointer* akan sangat dibutuhkan karena program yang biasanya dibuat dengan bahasa pemrograman C lebih mengedepankan membuat program dengan struktur modular, jadi suatu fungsi akan dipisahkan akan dibedakan tempatnya dengan fungsi utama dan apabila dibutuhkan hanya perlu dipanggil saja. Dan variabel *pointer* akan sangat dibutuhkan disini, karena variabel atau variabel yang bertipe data *pointer* ini akan berhubungan dengan konsep *call by value* ataupun *call by reference*. Singkatnya *call by value* adalah pemanggilan nilai dari suatu variabel yang berbeda fungsi, kemudian *call by reference* adalah pemanggilan alamat dari suatu variabel lain yang tentu saja bertempat di fungsi yang lain.

## 2.5 Operasi string

Sebuah tipe data *string* sebenarnya merupakan sebuah *array of character* atau himpunan karakter. Kemudian bahasa pemrograman C sendiri sudah menyediakan beberapa fungsi khusus yang memiliki kegunaan dalam mengoperasikan *string*. Salah satu *file header* yang menampung fungsi-fungsi tersebut adalah *file header* “<string.h>”. Berikut adalah fungsi atau fitur-fitur yang ada di dalam *file header* tersebut.

1. Strcpy

Fungsi ini digunakan untuk menyalin sebuah nilai dari variabel dengan tipe data *string* ke dalam variabel yang lain.

1. Strcmp

Fungsi ini digunakan untuk membandingkan dua buah *string*, termasuk *uppercase* dan *lowercase* akan dianggap berbeda.

1. Stcmpi

fungsi operator *string* yang mirip dengan strcmp, namun strcmpi tidak memerhatikan *string* tersebut merupakan huruf kapital atau tidak, atau bisa dibilang “*not case sensitive”*.

1. Strcat

fungsi operator *string* yang digunakan untuk menggabungkan 2 buah *string*.

1. Strlen

Fungsi yang digunakan untuk menghitung jumlah karakter di dalam sebuah variabel *string* atau bisa dibilangan mengukur panjangnya variabel *string* tersebut.

1. Strupr

fungsi yang digunakan untuk mengubah setiap huruf yang merupakan *lowercase* menjadi *uppercase*, namun jika huruf tersebut sudah merupakan *uppercase*, maka huruf tersebut tidak akan diubah.

1. Strlower

fungsi ini akan mengubah huruf yang tadinya uppercase menjadi *lowercase*¸ jika huruf tersebut sudah merupakan *lowercase*, maka tidak akan diubah.

## 2.6 Operasi file

Sesuai namanya, operasi yang satu ini adalah operasi khusus yang digunakan untuk melakukan operasi pada sebuah *file*, entah itu menambahkan data ke dalam suatu *file*, memodifikasi suatu data di dalam sebuah *file*, menghapus data ataupun mencari suatu data tertentu di dalam sebuah *file* serta tujuan lainnya.

Operasi *file* sendiri dibutuhkan karena satu masalah, yaitu sifat tidak permanennya suatu data di dalam sebuah program. Biasanya dalam membuat suatu program, jika terdapat data di dalamnya, maka disaat program telah selesai digunakan, maka semua data di dalam program tersebut akan hilang dengan sendirinya. Maka dari itu muncul sebuah pertanyaan “Bagaimana jika data tersebut dibutuhkan secara permanen?” Jawabannya adalah dengan menggunakkan metode operasi *file* ini.

Sederhananya, operasi *file* akan memasukkan data dari sebuah program ke dalam sebuah *file*, atau bisa juga mengambil sebuah data dari dalam *file* dan menampilkannya kembali dalam sebuah program. Dalam bahasa pemrograman C, operasi *file* dibagi menjadi tiga fungsi utama, yaitu *read* untuk membaca isi dari sebuah *file*, *write* untuk menuliskan data ke dalam sebuah *file* dan *append* untuk menambahkan data ke dalam sebuah *file* pada baris paling terakhir.

## 2.7 Array atau larik

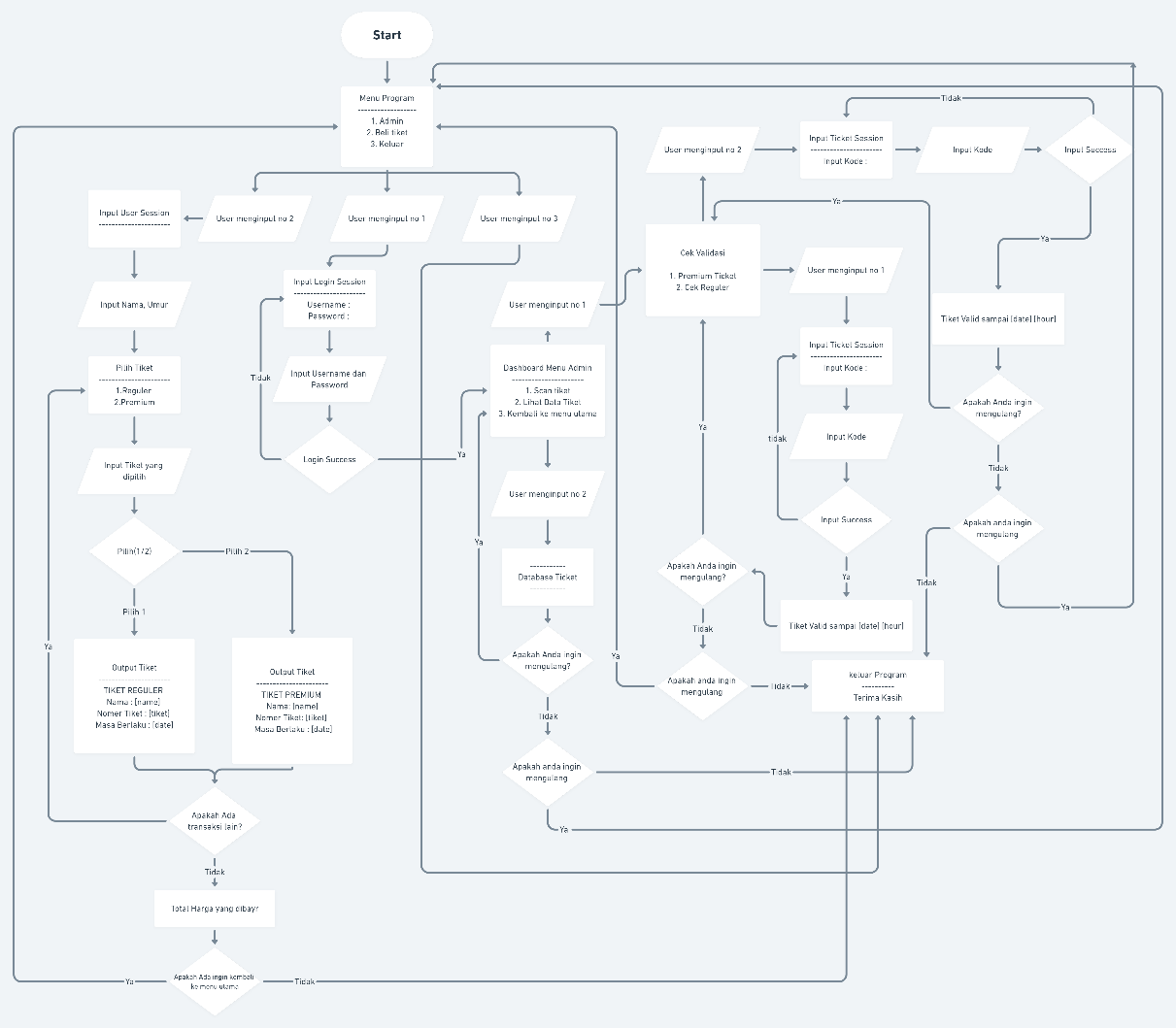
Dalam ilmu komputer, larik yang biasanya disebut dengan *array* adalah sebuah struktur data yang dapat menyimpan satu buah tipe data didalamnya dan setiap data yang ada akan disimpan di dalam suatu blok data, dimana setiap blok data akan diberikan penomoran atau alamat di dalam *array* tersebut, penomoran ini umumnya disebut dengan indeks.

Larik sendiri bisa diumpamakan seperti sebuah penomoran alamat di dalam sebuah perumahan. Setiap kode alamat rumah adalah indeks, dan rumah di alamat tersebut adalah data yang ada di dalamnya. Selian itu, posisi dari setiap rumah sudah tersusun secara rapi dari rumah dengan nomor yang paling kecil sampai dengan rumah yang memiliki nomor paling besar.

# BAB III

# DESAIN DAN METODE

## 3.1 Diagram alir



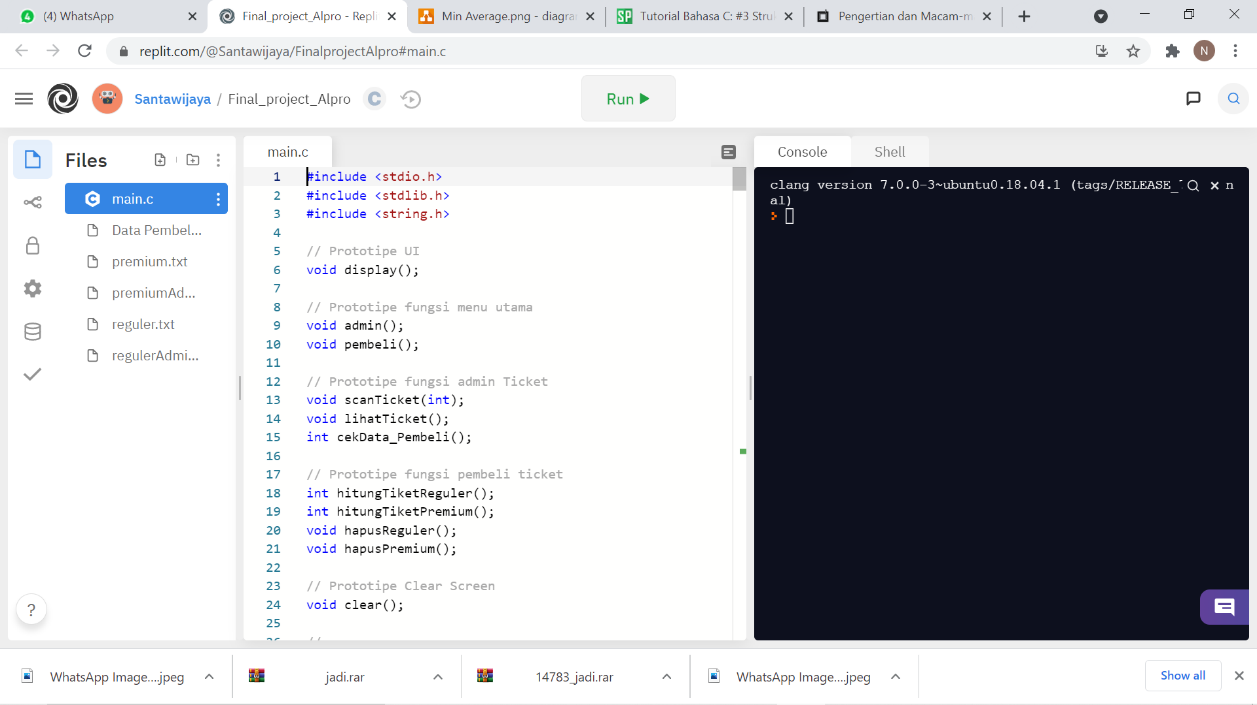
Gambar 3. 1 Diagram alir program

## Metode

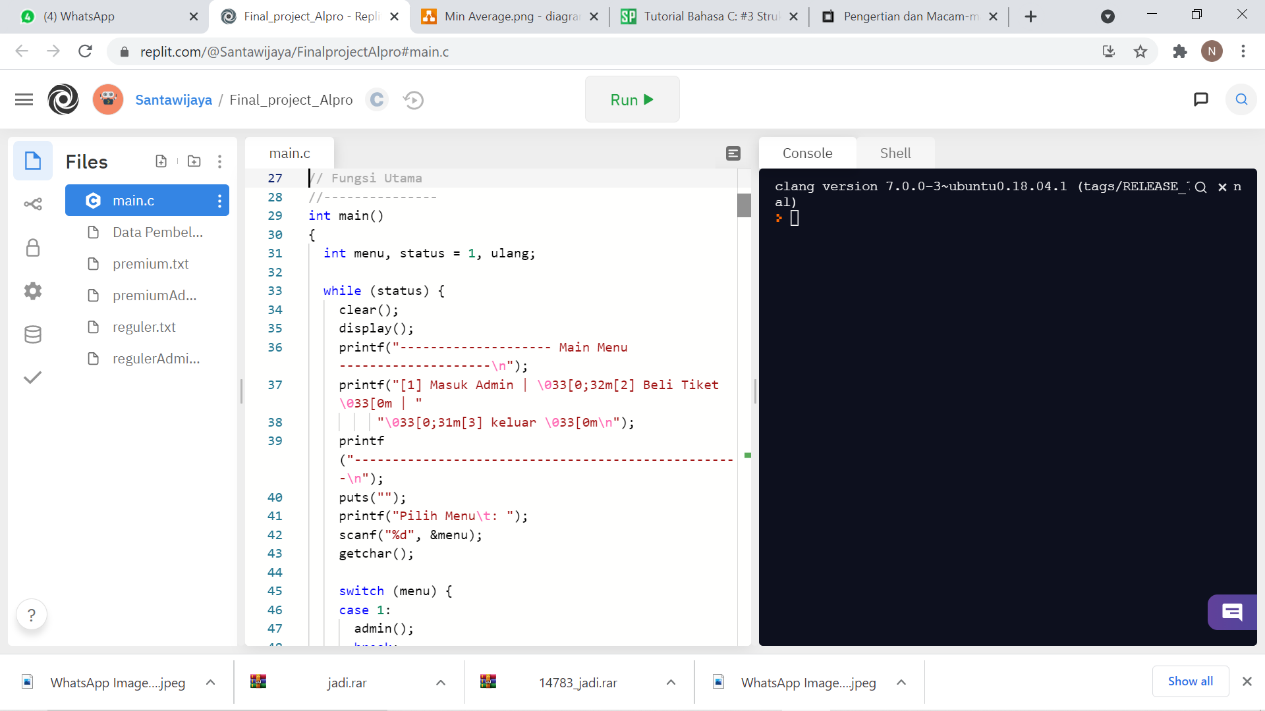
Dari kegiatan Tugas Akhir mata kuliah Praktikum Algoritma dan Pemrograman ini, kami telah mengaplikasikan beberapa materi yang ada di dalam landasan teori ke dalam program atau aplikasi yang kami buat, dimana materi-materi ini telah didapatkan selama modul 1 sampai modul 5 di mata kuliah Praktikum Algoritma dan Pemrograman. Maka dari itu, setiap konsep yang ada di dalam program yang kami buat ini berasalah dari materi yang telah diberikan, meskipun ada beberapa materi yang dipelajari secara prinbad. Berikut penjabaran dari kelima materi yang kami aplikasikan ke dalam program:

### 3.2.1 Struktur Dasar bahasa pemrograman C

Secara umum, dalam menggunakkan Bahasa Pemrograman C, setidaknya suatu program memiliki fungsi utama yang biasa disebut fungsi “main”, dan *file header* yang merupakan *file* berekstensi .h yang berisi fungsi-fungsi lain yang nantinya dibutuhkan dalam membangun sebuah program. Didalam program yang kami buat, kami menggunakan *header* “stdio.h” yang biasa digunakan untuk melakukan operasi *input-output*, selain itu kami juga menggunakan stdlib.h sebagai operasi pembanding, dan tidak lupa string.h yang berfungsi untuk melakukan manipulasi string.



Gambar 3. 2 Pengunaan *file header*



Gambar 3. 3 Fungsi utama atau *main*

### 3.2.2 Penyeleksian Kondisi dan Perulangan

Penyeleksian Kondisi atau bisa juga disebut sebagai percabangan digunakan untuk membuat program seolah-olah berpikir dan menentukan tindakan sesuai dengan logika/kondisi yang kita berikan. Dalam program yang kami buat, terdapat beberapa bagian yang menggunakan konsep percabangan yakni:

1. If-Else

Percabangan ini banyak ditemukan dalam program yang kami buat. Contoh saja pada *fungsi* main dimana pada bagian tersebut terdapat penyeleksian kondisi dimana program menanyakan kepada *user* apakah ingin mengulang ke menu, jika kondisi terpenuhi maka akan menjalankan proses yang terdapat pada blok percabangan *if*, sedangkan jika kondisi tidak terpenuhi maka akan menjalankan proses yang terdapat pada blok *else*. Pada intinya, percabangan ini kita gunakan untuk menyeleksi dari 2 kondisi, contoh lainnya selain pada fungsi *main* antara lain: memeriksa nama pengguna dan kata sandi pada menu admin, memeriksa stok tiket yang tersedia pada menu pembeli dan masih banyak lagi.

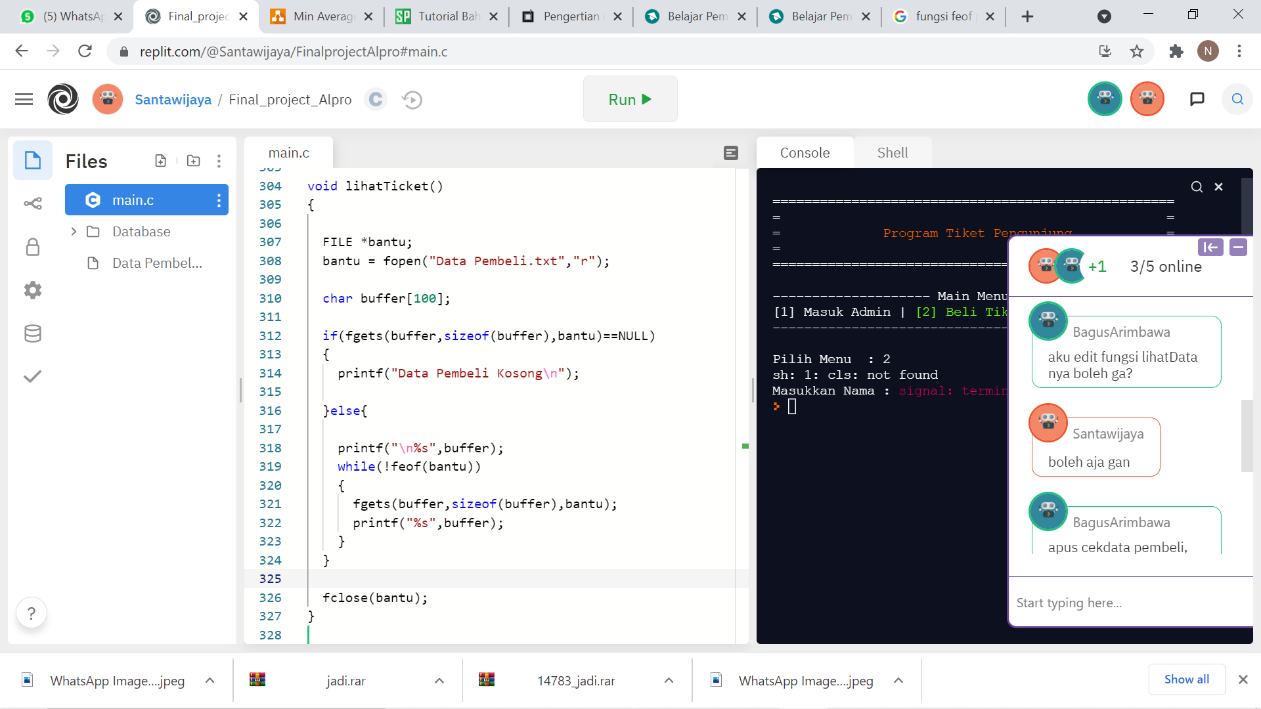
1. If

Percabangan ini hampir sama seperti dengan percabangan *if-else*, hanya saja yang membedakannya adalah percabangan ini hanya memiliki 1 blok kondisi yang akan menjalankan proses didalamnya, namun dengan ketika kondisi terpenuhi. Contoh yang terdapat dalam program salah satunya pada menu pembeli, pada bagian menyeleksi *input* kelas tiket oleh pengguna, yang jika kondisi dengan sintaks “input tiket=1” terpenuhi, maka komputer akan menjalankan proses didalam blok *if* tersebut.

1. Switch/case

Pada program yang kami buat, *switch case* kebanyakan kami aplikasikan pada bagian yang kaitannya dengan menu atau pilihan oleh pengguna. *Switch* ini memiliki beberapa kondisi yang berupa *case­-case* dimana jika terpenuhi maka akan menjalankan proses didalam *case* tersebut dan diikuti dengan *break* agar saat *case* yang terpilih dijalankan, perintah *break* akan menghentikan proses di dalam *switch* agar *case* setelahnya tidak ikut dijalankan. Salah satu contohnya terdapat pada bagian menu utama program dimana terdapat 3 *case*, tiap-tiap *case* akan menjalankan masing masing proses didalamnya apabila kondisinya sesuai dengan inputan variabel “menu” oleh pengguna.

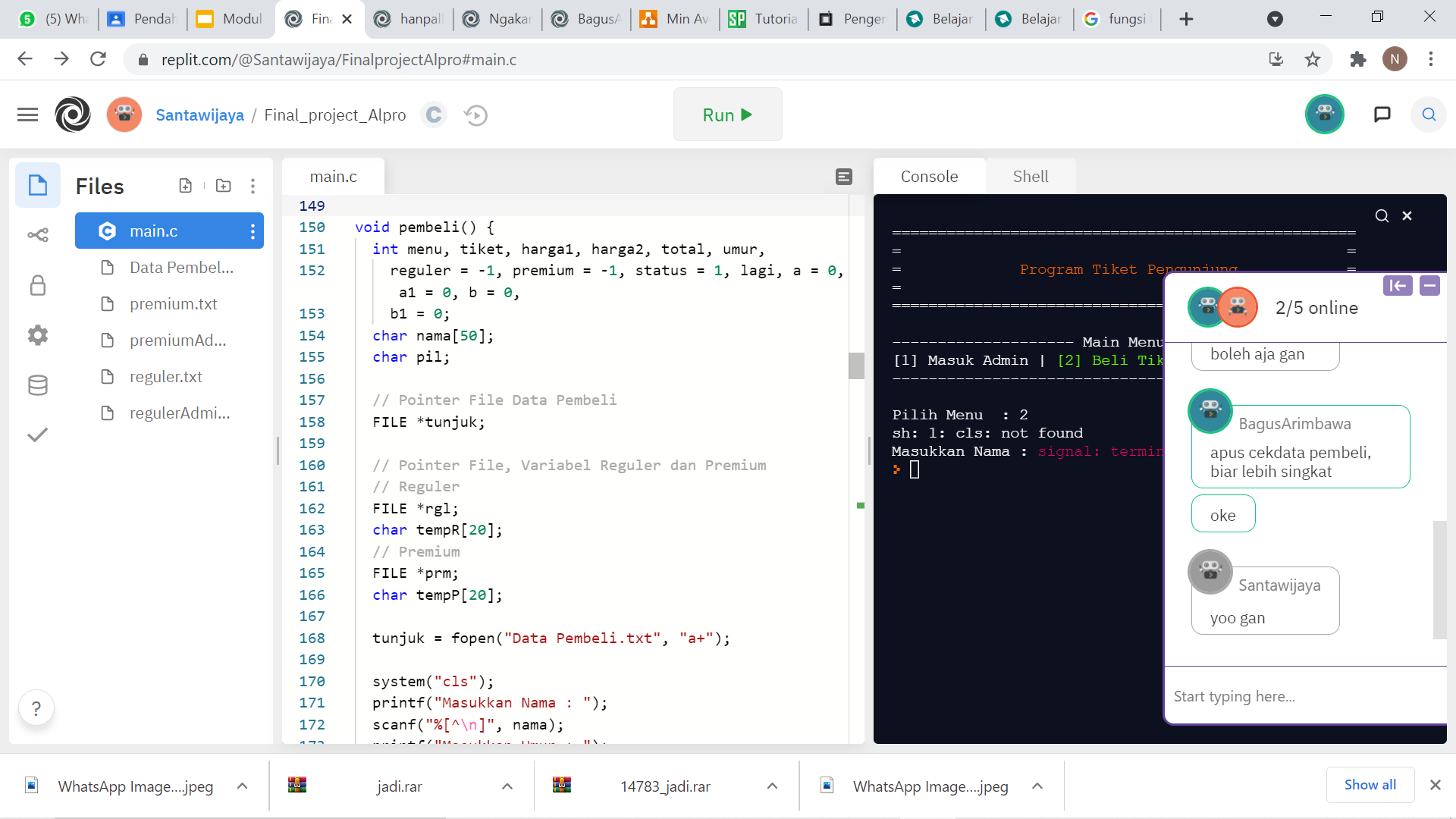
Untuk perulangan sendiri, cukup umum dalam mengolah proses yang berulang. Secara garis besar perulangan sendiri terdiri dari perulangan *for*, *while* dan *do-while*. Dalam program yang kami buat, kebanyakan kami menggunakan perulangan *while*. Untuk perulangan *while*, salah satu contoh yaitu pada fungsi bernama “lihatTicket” untuk menampilkan seluruh data pembelian tiket, Ketika variabel *pointer* bernama bantu yang ada di dalam sebuah *file* bernama “data pembeli.txt” berada di posisi bukan akhir dari *file* tersebut*,* dimana sintaksnya adalah “(while(!feof(bantu))”, maka komputer akan selalu mencetak karakter di baris dia berada yang ada di dalam *file* tersebut. Pada dasarnya perulangan *while* ini tidak akan berhenti selama kondisi di dalamnya terus terpenuhi.



Gambar 3. 4 Pengunaan perulangan *while*

### 3.2.3 Array

Dalam konsep array atau larik, terdapat beberapa jenis *array* seperti *array* 1 dimensi, 2 dimensi, atau multidimensi. Pada program yang kami buat kami hanya menggunakan *array* 1 dimensi saja. Penggunaan *array* dalam program digunakan untuk menginisialisasi *array of char* seperti nama, variabel bantu untuk mencetak/ memasukkan data kedalam *file*. Contoh penggunaan *array* dapat kita lihat pada menu pembeli untuk menginisialisasi nama pembeli diikuti dengan batas ukuran dari array tersebut.



Gambar 3. 5 Penggunaan *array*

### 3.2.4 Pointer, Fungsi dan Operasi File

Program ini juga mempergunakan beberapa fungsi modular dan *pointer* untuk melakukan operasi *file*. Fungsi adalah sekelompok baris kode yang melakukan suatu tugas tertentu. Tiap program yang dibuat dengan bahasa pemrograman C akan memiliki setidaknya 1 fungsi, yaitu fungsi “main”. Dalam program ini, fungsi digunakan untuk memisahkan berbagai tugas yang dapat dipilih dan dijalankan dalam program. Sementara itu, *pointer* tipe FILE digunakan untuk menyimpan alamat *file* yang dibuka dengan menggunakkan fungsi fopen, sehingga dapat diakses untuk melakukan berbagai operasi *file* yang dibutuhkan sebelum *file* ditutup dan disimpan menggunakan fungsi fclose.

Program ini memiliki 2 fungsi utama, yaitu fungsi admin, yang memuat fungsi – fungsi operator admin, dan fungsi “pembeli” yang memuat fungsi – fungsi pengguna secara umum. Fungsi “admin” memuat beberapa fungsi, yaitu fungsi bernama “dashboardAdmin” yang berguna untuk menampilkan menu dari seorang admin, fungsi “scanTicket” yang berguna untuk mengecek valid atau tidaknya sebuah kode tiket yang dimasukkan, dan fungsi “lihatTicket” yang berguna untuk menampilkan data seluruh pembeli beserta ticketnya.

Kemudian fungsi “scanTicket” menggunakan *pointer* untuk melakukan operasi *file*, dalam hal ini membuka sebuah *file* mlik admin sesuai dengan tipe dari tiket yang dipilih, dan *file* admin ini dibuka dalam mode baca. Jika yang terpilih adalah verifikasi tiket regular, maka akan membuka *file* berjudul “regulerAdmin.txt”. Jika yang terpilih verifikasi tiket premium, maka akan membuka file berjudul “premiumAdmin.txt”. Selanjutnya, tiap baris dalam *file* admin terpilih akan dibaca dan dibandingkan dengan kode yang dimasukkan oleh pengguna. Jika kode ditemukan sebelum *pointer-in-file* (kursor) mencapai akhir dari *file*, maka kode tersebut akan dianggap valid, namn jika kode tidak ditemukan, maka kode tidak akan dianggap valid.

Fungsi “lihatTicket” menggunakan *pointer* untuk melakukan operasi *file* juga, dalam hal ini membuka file berjudul “Data Pembeli.txt” dalam mode baca. Fungsi ini membaca dan mengecek isi *file* yang dibuka. Jika baris pertama kosong, *file* akan mengasumsikan *file* adalah kosong, dan akan menampilkan pesan “error”. Jika baris pertama ditemukan satu karakter saja, maka fungsi akan menampilkan seluruh isi *file* hingga mencapai akhir *file*.

Fungsi utama lainnya, yaitu fungsi “pembeli”, fungsi ini berguna untuk melakukan proses pembelian tiket. Pengguna memasukkan data diri, kemudian memilih jenis tiket yang ingin dipilih. Fungsi “userdisplay” menampilkan menu tiket beserta harganya dan banyak tiket tiap jenis yang tersisa. Tiket bisa dibeli berulang - ulang dalam kombinasi apapun selama stok tiket masih ada, dan harga tiket yang harus dibayar akan sesuai dengan kombinasi jumlah tiket yang dibeli. Stok tiket dicek menggunakan fungsi bernama “hitungTiketReguler” dan “hitungTiketPremium. Meski keduanya berbeda nama, namun kedua fungsi tersebut bekerja dengan cara yang sama, yaitu menghitung jumlah baris dalam *file* masing-masing.

Kemudian data diri pembeli atau pengguna serta data transaksi dimasukkan menggunakkan mode *append* ke dalam sebuah *file* dengan nama “Data Pembeli.txt”. Stok tiket yang tersedia dalam *file* basis data tiket masing – masing dikurangkan tiap adanya pembelian tiket tipe tersebut menggunakan fungsi “hapusReguler” dan “hapusPremium”. Keduanya bekerja dengan cara menyalin isi seluruh dari file basis data tiket kecuali baris yang berisi kode yang dikeluarkan ke dalam sebuah *file* .txt sementara, kemudian *file* asli akan dihapus, lalu merubah nama *file* sementara menjadi nama *file* yang asli.

Selain 2 fungsi utama tadi, program ini juga menggunakan fungsi pembantu, yaitu fungsi bernama “display” yang berfungsi untuk menampilkan tampilan menu, dan fungsi “clear” yang berfungsi untuk menghapus *output* di terminal/ program dari layar yang tentu saja bersifat *cross-platform* (tetap berfungsi di Sistem operasi Windows ataupun UNIX).

# BAB IV

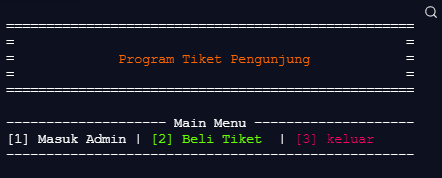
# HASIL DAN IMPLEMENTASI

## 4.1 Sintaks program

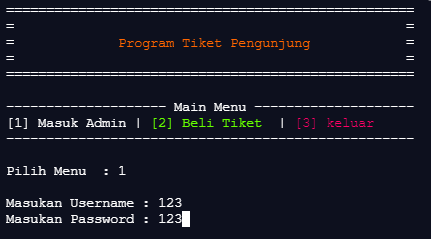
|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  // Prototipe UI  void display();  void dashboardAdmin();  void userdisplay();  // Prototipe fungsi menu utama  void admin();  void pembeli();  // Prototipe fungsi admin Ticket  void scanTicket(int);  void lihatTicket();  int cekData\_Pembeli();  // Prototipe fungsi pembeli ticket  int hitungTiketReguler();  int hitungTiketPremium();  void hapusReguler();  void hapusPremium();  // Prototipe Clear Screen  void clear();  //---------------  // Fungsi Utama  //---------------  int main()  {  int menu, status = 1, ulang;  while (status) {  clear();  display();  printf("-------------------- Main Menu --------------------\n");  printf("[1] Masuk Admin | \033[0;32m[2] Beli Tiket \033[0m | "  "\033[0;31m[3] keluar \033[0m\n");  printf("---------------------------------------------------\n");  puts("");  printf("Pilih Menu\t: ");  scanf("%d", &menu);  getchar();  switch (menu) {  case 1:  admin();  break;  case 2:  pembeli();  break;  case 3:  printf("\nTERIMA KASIH.\n");  return 0;  break;  }  printf("\n\nApakah ingin kembali ke menu utama ? [1/0]: ");  scanf("%d", &ulang);  getchar();  if (ulang == 1)  status = 1;  else  status = 0;  }  printf("\nTERIMA KASIH.\n");  }  void admin() {  int menu, status = 1, ulang, total;  int hasil1, hasil2;  char namaAdmin[20];  char passwordAdmin[20];  char admin[] = "123";  char pass[] = "123";  // Variabel Untuk Scan  int cek;  printf("\nMasukan Username : ");  scanf("%[^\n]", namaAdmin);  getchar();  printf("Masukan Password : ");  scanf("%[^\n]", passwordAdmin);  getchar();  hasil1 = strcmp(namaAdmin, admin);  hasil2 = strcmp(passwordAdmin, pass);  total = hasil1 + hasil2;  if (total == 0) {  while (status) {  clear();  printf("(1). Scan Tiket\n");  printf("(2). Lihat Data Ticket\n");  printf("(3). Kembali ke menu utama\n");  printf("\nPilih Menu\t: ");  scanf("%d", &menu);  getchar();  switch (menu) {  case 1:  printf("\nCek Ke-Validan Tiket\n");  printf("[1] Cek Premium\n");  printf("[2] Cek Reguler\n");  printf("\nPilihan : ");  scanf("%d", &cek);  getchar();  scanTicket(cek);  break;  case 2:  lihatTicket();  break;  case 3:  return;  break;  default:  printf("\nTidak ada pilihan...\n");  break;  }  printf("\nApakah ingin mengulang ?");  printf("\nTekan 1 jika iya, tekan 0 jika tidak");  printf("\npilihan : ");  scanf("%d", &ulang);  getchar();  if (ulang == 1)  status = 1;  else  status = 0;  }  } else {  printf("Usename dan Password tidak valid...");  }  }  void pembeli() {  int menu, tiket, harga1, harga2, total, umur,  reguler = -1, premium = -1, status = 1, lagi, a = 0, a1 = 0, b = 0,  b1 = 0;  char nama[50];  char pil;  // Pointer File Data Pembeli  FILE \*tunjuk;  // Pointer File, Variabel Reguler dan Premium  // Reguler  FILE \*rgl;  char tempR[20];  // Premium  FILE \*prm;  char tempP[20];  tunjuk = fopen("Data Pembeli.txt", "a+");  system("cls");  printf("Masukkan Nama : ");  scanf("%[^\n]", nama);  printf("Masukkan Umur : ");  scanf("%d", &umur);  fprintf(tunjuk, "#-----#\n");  fprintf(tunjuk, "Nama Pembeli : %s\n", nama);  fprintf(tunjuk, "Umur Pembeli : %d\n", umur);  fprintf(tunjuk, "\nDAFTAR TIKET\n");  fclose(tunjuk);  while (status) {  clear();  printf("\nDaftar Harga Tiket\n");  printf(  "[1] \033[0;34m Reguler - 50k Tersisa %d lembar \033[0m\n",  hitungTiketReguler());  printf(  "[2] \033[0;36m Premium - 100k Tersisa %d lembar \033[0m\n\n", hitungTiketPremium());  printf("Masukkan Kelas tiket yang ingin dibeli : ");  scanf("%d", &tiket);  getchar();  if (tiket == 1) {  if (hitungTiketReguler() == 0)  printf("\nmohon maaf tiket sudah habis...\n");  else {  harga1 = 50000;  a++;  a1++;  rgl = fopen("reguler.txt", "r");  tunjuk = fopen("Data Pembeli.txt", "a");  fgets(tempR, sizeof(tempR), rgl);  printf("\nTIKET REGULER");  printf("\nNama : %s", nama);  printf("\nNomer Tiket : %s",tempR);  printf("\nMasa Berlaku : 26/05/2021 [21:00]");  fprintf(tunjuk,"Nomer Tiket : %s",tempR);  fprintf(tunjuk,"Masa Berlaku : 26/05/2021 [21:00]");  fprintf(tunjuk,"\n\n");  fclose(rgl);  fclose(tunjuk);  if (a1++)  hapusReguler();  }  }  if (tiket == 2) {  if (hitungTiketPremium() == 0)  printf("\nmohon maaf tiket sudah habis\n");  else {  harga2 = 100000;  b++;  b1++;  prm = fopen("premium.txt", "r");  tunjuk = fopen("Data Pembeli.txt", "a");  fgets(tempP, sizeof(tempP), prm);  printf("\nTIKET PREMIUM\n");  printf("\nNama: %s", nama);  printf("\nNomer Tiket: %s", tempP);  printf("\nMasa Berlaku : 26/05/2021 [21:00]");  fprintf(tunjuk,"Nomer Tiket : %s",tempP);  fprintf(tunjuk,"Masa Berlaku : 26/05/2021 [21:00]");  fprintf(tunjuk,"\n\n");  fclose(prm);  fclose(tunjuk);  if (b1++)  hapusPremium();  }  }  printf("\n\nApakah ada Transaksi lain [1/0] ? : ");  scanf("%d", &lagi);  if (lagi == 1)  status = 1;  if (lagi == 0)  status = 0;  }  total = (a \* harga1) + (b \* harga2);  printf("\n\nTotal harga harus dibayar sebanyak : Rp %d ", total);  }  // ---------------------  // Fungsi sebagai admin  // ---------------------  void scanTicket(int x)  {  char kode[100];  char buffer[100];  printf("\nInput Kode : ");  scanf("%s",kode);  getchar();  FILE \*scn;  if (x == 1) {  scn = fopen("premiumAdmin.txt","r");  } else {  scn = fopen("regulerAdmin.txt","r");  }  int comp = 1;  while (!feof(scn))  {  fscanf(scn,"%s\n",&buffer);  comp = comp \* strcmp(kode,buffer);  }  if (comp == 0){  printf("\n\nTicket [ %s ] Valid sampai 26/05/2021 [21:00]\n",kode);  } else {  printf("\n\nTicket Tidak Valid\n");  }  fclose(scn);  }  void lihatTicket()  {  FILE \*bantu;  bantu = fopen("Data Pembeli.txt","r");  char buffer[100];  cekData\_Pembeli();  if(cekData\_Pembeli()==1)  {  printf("Data Pembeli Kosong\n");  }else{  printf("\n");  while(!feof(bantu))  {  fgets(buffer,sizeof(buffer),bantu);  printf("%s",buffer);  }  }  fclose(bantu);  }  int cekData\_Pembeli()  {  FILE \*cek = fopen("Data Pembeli.txt","r");  char temp[100];  if((fgets(temp,sizeof(temp),cek))==NULL)  return 1;  else  return 0;  fclose(cek);  }  // ----------------------  // Fungsi sebagai pembeli  // ----------------------  int hitungTiketReguler() {  // Variabel  int baris = 1;  char buff[50];  FILE \*tunjuk1;  tunjuk1 = fopen("reguler.txt", "r");  while (fgets(buff, sizeof(buff), tunjuk1) != NULL) {  baris++;  }  fclose(tunjuk1);  return baris - 1;  }  int hitungTiketPremium() {  // Variabel  int baris = 1;  char buff[50];  FILE \*tunjuk1;  tunjuk1 = fopen("premium.txt", "r");  while (fgets(buff, sizeof(buff), tunjuk1) != NULL) {  baris++;  }  fclose(tunjuk1);  return baris - 1;  }  void hapusReguler() {  char filename[] = "reguler.txt";  char temp[20];  FILE \*rgl1;  FILE \*rgl2;  int del = 1;  int line = 0;  rgl1 = fopen(filename, "r");  rgl2 = fopen("dum.txt", "w");  while (fgets(temp, sizeof(temp), rgl1) != NULL) {  line++;  if (line != del)  fputs(temp, rgl2);  }  fclose(rgl1);  fclose(rgl2);  remove(filename);  rename("dum.txt", filename);  }  void hapusPremium() {  char filename[] = "premium.txt";  char temp[20];  FILE \*prm1;  FILE \*prm2;  int del = 1;  int line = 0;  prm1 = fopen(filename, "r");  prm2 = fopen("dum.txt", "w");  while (fgets(temp, sizeof(temp), prm1) != NULL) {  line++;  if (line != del)  fputs(temp, prm2);  }  fclose(prm1);  fclose(prm2);  remove(filename);  rename("dum.txt", filename);  }  //-------------  // Fungsi Lain  //-------------  void clear()  {  #ifdef \_WIN32  std : system("cls");  #else  std : system("clear");  #endif  }  void display() {  puts("");  printf("===================================================\n");  printf("= =\n");  printf("= \033[0;33m Program Tiket Pengunjung \033[0m "  "=\n");  printf("= =\n");  printf("===================================================\n");  puts("");  }  void dashboardAdmin() {  puts("");  printf("===================================================\n");  printf("= =\n");  printf("= \033[0;33m Dashboard Admin \033[0m"" =\n");  printf("= =\n");  printf("===================================================\n");  puts("");  }  void userdisplay(){  puts("");  printf("===================================================\n");  printf("= =\n");  printf("= \033[0;33m Dashboard User \033[0m"" =\n");  printf("= =\n");  printf("===================================================\n");  puts("");  } |

Tabel 4. 1 Sintaks program

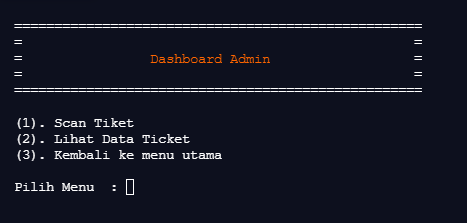
## 4.2 Contoh output program



Gambar 4. 1 Tampilan menu utama

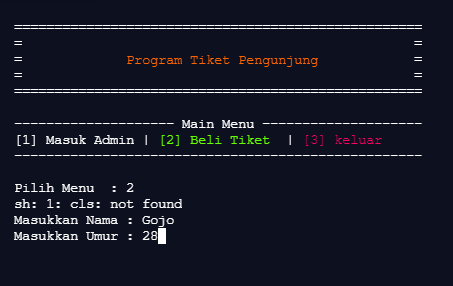


Gambar 4. 2 Tampilan sesi input

Gambar 4. 3 Tampilan menu admin



Gambar 4. 4 Menu admin pilihan 1



Gambar 4. 5 Tampilan pengguna membeli tiket



Gambar 4. 6 Tampilan proses pengguna membeli tiket

## 4.3 Penjelasan singkat program

Disini kami membuat sebuah Program Manajemen Tiket yang cukup *advanced* yang mencangkup seluruh modul yang telah diajarkan oleh para asisten dosen mata kuliah Algoritma dan Pemrograman, tentu saja program ini dibuat menggunakan bahasa C. Disini kita menginisiasikan beberapa fungsi seperti fungsi “main”, fungsi “admin” dan fungsi “user” lalu kemudian didalam fungsi “admin” ada beberapa fungsi lainnya yang berfungsi untuk memeriksa tiket dan melihat tiket, di bagian fungsi untuk memeriksa tiket, program melakukan *scan* atau pemeriksaan nomor tiket, apakah nomor tiket tersebut valid atau tidak, disini juga akan diperlihatkan masa berlaku yang telah ditentukan.

Lalu untuk bagian fungsi lihat data tiket itu digunakan untuk melihat tiket yang telah dipesan atau dibeli oleh para pengguna. Lanjut pada fungsi beli tiket disini tentu saja pengguna akan membeli tiket, namun sebelum membeli, pengguna harus terlebih dahulu mengisikan data diri pengguna, kemudian setelah pengguna selesai memasukkan data diri, pengguna akan diarahkan ke menu pilihan tiket.

Disini terdapat dua pilihan, diantaranya tiket premium dan regular, ketika pengguna memilih salah satu dari kedua tiket tersebut, maka tiket tersebut akan dimasukan ke dalam keranjang, setelah itu pengguna akan dihadapkan dengan sebuah kondisi atau pertanyaan “Apakah anda ingin menbeli lagi atau tidak?” Jika iya, maka akan program akan menampilkan kembali menu pembelian tiket. Namun jika tidak, maka pengguna akan diberitahukan sebuah nominal harga yang sesuai dengan tiket yang telah dipesan, entah itu reguler atau premium, serta hanya membeli satu atau membeli lebih dari satu, semuanya akan diakumulasikan.

Dan terakhir, pengguna akan mendapatkan sebuah barisan kode acak dari basis data yang telah dibuat oleh pengembang, dimana kode ini adalah karakter unik yang ada pada tiket tersebut. Jika tidak ada transaksi lagi, maka program akan diselesaikan dan dimatikan.

# BAB V

# PENUTUP

## 5.1 Kesimpulan

Berdasarkan hasil riset atau penelitian yang dilakukan oleh Kementrian Komunikasi dan Informatika, data penggunaan internet di Indonesia mengalamai peningkatan yang luar biasa, hal ini sendiri didukung dengan semakin banyaknya orang yang menggunakkan ponsel pintar, bahkan menurut angka dan data dari hasil riset tersebut, pengguna ponsel pintar di Indonesia sudah mencapai 167 juta orang atau kurang 89% penduduk di Indonesia. Dari hal ini kita bisa prediksi bahwa, semua hal bisa saja akan mengarah ke hal-hal yang berbau digital, teknologi ataupun komputer.

Tentu hal ini menjadi sebuah peluang besar, khususnya bagi mereka yang menekuni bidang-bidang yang berkaitan dengan komputer. Hal ini bisa ditandai dengan semakin banyak aplikasi-aplikasi ataupun program yang bisa digunakan, baik itu menggunakkan komputer ataupun ponsel pintar. Dari sini bisa ditarik sebuah kesimpulan bahwa, membuat suatu produk yang berbau digital memiliki masa depan yang cerah, dan jika berbicara kaitannya dengan membuat produk digital, maka tidak akan lepas dari yang namanya pemrograman.

Pemrograman sendiri adalah sebuah metode komunikasi manusia dengan komputer agar komputer bisa melakukan pekerjaan yang sesuai dengan keinginan manusia, jika seseorang sudah menguasai pemrograman tingkat tinggi, maka dengan komputer dan tingkat kreativitas yang tinggi, seseorang sudah bisa mendapatkan pendapatan dari kemampuan dan skill yang dimiliki. Maka sebaiknya, khususnya untuk anak-anak muda, mulailah belajar tentang teknologi khususnya komputer, karena dunia akan terus berkembang, dan penulis yakin kehidupan manusia tidak akan lepas dari teknologi. Dan kegiatan tugas akhir ini sangatlah bermanfaat dalam melatih mahasiswa mengkombinasikan kemampuann atau skill mereka dengan kreativitas atau ide mereka masing-masing.

## 5.2 Saran

Saran untuk kedepannya adalah semoga semakin banyak orang atau pribadi yang tertarik dengan dunia komputer, khususnya yang berhubungan dengan pemrograman. Selain membantu pemerintah dalam mencukupi kebutuhan *programmer* di Indonesia, penulis yakin bahwa setiap orang yang belajar tentang komputer atau pemrograman akan memiliki pola pikir yang lebih sistematis dan terstuktur dibandingkan dengan cara bagaimana ia berpikir sebelumnya.

Kemudian, karena ada banyak sekali kegiatan serta teknologi yang berkaitan dengan komputer, penulis menyarankan untuk pilihlah pekerjaan yang memang cocok dengan diri sendiri, tetaplah penasaran dengan hal yang digemari. Selain itu, ada banyak sekali teknologi ataupun bahasa pemrograman yang bisa dipilih, karena itu disarankan untuk belajarlah dari hal yang dasar, jika masih awal-awal belajar, pilihlah teknologi yang mudah untuk dipelajari. Karena, semakin lama jam terbang dalam belajar, tanpa sadar proses itu akan membawa kemampuan kalian ke tingkat yang lebih tinggi.

# DAFTAR PUSTAKA

Sutiono. (2021). Apa itu Algoritma Percabangan? [Online] https://dosenit.com/kuliah-it/pemrograman/algoritma-percabangan. (Diakses 24 Mei 2021).

Khoirunisa, Rizqia. (2016). Algorimta pengulangan looping: (pengertian, contoh kasus dan konsep dasar) [Online]. http://referensisiswa.blogspot.com/2016/09/algoritma-pengulangan-looping.html#:~:text=Pengulangan%20atau%20disebut%20sebagai%20looping,dan%20untuk%20mempersingkat%20instruksi%20program. (Diakses 24 Mei 2021).

Muhardian, Ahmad. (2019). Belajar Pemrograman C #9: Mengenal Struktur Data Array pada C. [Online] https://www.petanikode.com/c-array/. (Diakses 24 Mei 2021)

Muhardian, Ahmad. (2019). Belajar Pemrograman C #11 Apa itu Pointer? [Online] https://www.petanikode.com/c-pointer/. (Diakses 24 Mei 2021).

Programiz. (2021). C Files I/O: Opening, Reading, Writing and Closing a file. [Online] https://www.programiz.com/c-programming/c-file-input-output. (Diakses 24 Mei 2021).

Mahir Koding. (2016). Operasi FILE dalam Bahasa C. [Online] https://www.mahirkoding.com/operasi-file-dalam-bahasa-c/. (Diakses 24 Mei 2021).

Mahir Koding. (2016). Operasi FILE dalam Bahasa C. [Online] https://www.mahirkoding.com/operasi-file-dalam-bahasa-c/. (Diakses 24 Mei 2021).

Bansal, Ritik. (2018). Strupr() function in c. [Online] https://www.geeksforgeeks.org/strupr-function-in-c/. (Diakses 24 Mei 2021).

Bansal, Ritik. (2018). Strcmpi() function in c. [Online] https://www.geeksforgeeks.org/strcmpi-function-in-c/. (Diakses 24 Mei 2021).

Bansal, Ritik. (2018). Strlwr() function in c. [Online] https://www.geeksforgeeks.org/strlwr-function-in-c/. (Diakses 24 Mei 2021).

Programiz. (2020). C . [Online] https://www.programiz.com/cprogramming/library-function/string.h. (Diakses 24 Mei 2021).