# LAPORAN

**PROJECT AKHIR**

**ALGORITMA PEMROGRAMAN II**



DOSEN PENGAMPU:

Faisal Akbar, S. Kom, M.T.

Disusun oleh:

1. Anam Sadat (14522708)
2. Eysa Aulia Ahmad (14522689)
3. Munji Iyabi (14522693)

**TEKNIK INFORMATIKA**

**STIKOM POLTEK CIREBON**

Jl. Brigjend Dharsono No. 33, Kedawung, Cirebon, Jawa Barat 45153, Indonesia

# KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT, yang atas rahmat-Nya maka kami dapat menyelesaikan penyusunan Laporan Project Akhir ini.

Dalam Penulisan laporan ini kami merasa masih banyak kekurangan-kekurangan baik pada teknis penulisan maupun materi, mengingat akan kemampuan yang dimiliki penyusun. Untuk itu kritik dan saran dari semua pihak sangat penulis harapkan demi penyempurnaan pembuatan makalah ini.

Akhirnya kami berharap semoga Allah memberikan imbalan yang setimpal pada mereka yang telah memberikan bantuan, dan dapat menjadikan semua bantuan ini sebagai ibadah serta laporan ini dapat bermanfaat bagi pembaca. Amiin Yaa Robbal ‘Alamiin.

Sebagai penyusun kami mengucapkan terima kasih kepada Bapak Faisal Akbar, S. Kom, M.T. yang telah membagikan ilmu pengetahuan hingga kami dapat melakukan penyusunan laporan ini.

|  |  |
| --- | --- |
|  | Cirebon, 10 Juli 2023  Penyusun, |

# DAFTAR ISI

[LAPORAN 1](#_Toc140750644)

[KATA PENGANTAR i](#_Toc140750645)

[DAFTAR ISI ii](#_Toc140750646)

[BAB I PENDAHULUAN 1](#_Toc140750647)

[1.1 Deskripsi Masalah 1](#_Toc140750648)

[1.2 Rumusan Masalah 1](#_Toc140750649)

[1.3 Batasan Masalah 1](#_Toc140750650)

[1.4 Tujuan 2](#_Toc140750651)

[BAB II DASAR TEORI 3](#_Toc140750652)

[2.1 Perkondisian 3](#_Toc140750653)

[2.1.1 If else 3](#_Toc140750654)

[2.1.2 Switch case 3](#_Toc140750655)

[2.2 Perulangan 4](#_Toc140750656)

[2.2.1 Pernyataan Goto 4](#_Toc140750657)

[2.2.2 While 4](#_Toc140750658)

[2.3 Prosedur 5](#_Toc140750659)

[BAB III PERANCANGAN 6](#_Toc140750660)

[3.1 Flowchart 6](#_Toc140750661)

[3.2 Analisis Program 8](#_Toc140750662)

[BAB IV HASIL DAN PEMBAHASAN 9](#_Toc140750663)

[4.1 Implementasi Program/Tutorial 9](#_Toc140750664)

[4.2 Source Code 15](#_Toc140750665)

[BAB V PENUTUP 37](#_Toc140750666)

[5.1 Kesimpulan 37](#_Toc140750667)

[5.2 Saran 37](#_Toc140750668)

# PENDAHULUAN

## Deskripsi Masalah

Di negara Wakanda, pemerintah menghadapi tantangan dalam proses pengurusan Surat Izin Mengemudi (SIM) bagi penduduknya. Proses manual yang rumit, waktu yang dibutuhkan yang lama, dan adanya kemungkinan kesalahan administrasi menyebabkan ketidaknyamanan dan kurang efisiennya proses pengurusan SIM. Oleh karena itu, diperlukan pembuatan sistem yang dapat mempermudah dan meningkatkan efisiensi dalam pengurusan Surat Izin Mengemudi di negara Wakanda.

## Rumusan Masalah

* Bagaimana cara membuat sistem yang efektif untuk proses pengurusan Surat Izin Mengemudi di negara Wakanda?
* Bagaimana cara mengoptimalkan proses pendaftaran, pengujian, dan penerbitan SIM agar lebih efisien dan akurat?
* Bagaiamana sistem dapat membantu meningkatkan keamanan dan validitas Surat Izin Mengemudi di negara Wakanda?

## Batasan Masalah

* Sistem yang dibuat difokuskan untuk digunakan oleh otoritas yang berwenang di Wakanda, seperti Departemen Trasnportasi atau instansi serupa.
* Sistem akan mencakup fitur-fitur seperti pendaftar calon pengemudi, lihat antrian, pembayaran, hapus antrian, mencari data dari antrian, dan tampilkan statistic.
* Pembuatan sistem akan mempertimbangkan kebijakan dan regulasi pemerintah Wakanda terkait privasi, keamanan data, dan persyaratan pengujian kemampuan mengemudi.

## Tujuan

* Mempermudah penduduk Wakanda dalam proses pengurusan Surat Izin Mengemudi.
* Meningkatkan efisiensi dan akurasi dalam proses pendaftaran, pengujian, dan penerbitan SIM di negara Wakanda.
* Menjamin keamanan dan validitas Surat Izin Mengemudi yang dikeluarkan oleh pemerintah Wakanda.
* Meningkatkan kesadaran dan kepatuhan masyarakat terhadap aturan lalu lintas dan regulasi berkendara di negara Wakanda.

# DASAR TEORI

## Perkondisian

Pernyataan pengkondisian adalah intuksi pada pemrograman untuk menjalankan suatu baris program sesuai dengan kondisi tertentu.

### If else

Kondisi **IF ELSE** adalah kondisi dimana jika pernyataan benar (True) maka kode dalam **IF** akan dieksekusi, tetapi jika bernilai salah (False) maka akan mengeksekusi kode di dalam **ELSE**.

Berikut fungsi if else didalam program Pendaftaran SIM:

* Untuk pengecekkan antrian masih kosong.
* Untuk menentukan jenis sim dan harganya.
* Digunakan untuk pembayaran sesuai dengan harga.
* Untuk mengecek head berada detail(node hanya ada satu).
* Untuk mengecek data yang dicari apakah ada atau tidak.
* Untuk mengecek data yang dicari adalah nik atau nomor hp.
* Untuk mengecek di node tersebut jenis kelaminnya perempuan atau laki-laki dan mengecek jenis SIMnya adalah a, b, atau c.

### Switch case

Pernyataan switch adalah **pernyataan pilihan ganda**. Setelah memberikan pilihan dan ekspresi yang relevan untuk setiap pilihan, Itu melihat melalui pilihan sampai menemukan pilihan yang cocok dengan ekspresi dan mengeksekusinya.

Berikut fungsi switch case didalam program Pendaftaran SIM:

* Untuk memilih pilihan menu di sistem Pendaftaran SIM

## Perulangan

Perulangan adalah suatu proses eksekusi terus statemen-statemen dalam sebuah program secara terus menerus sampai terdapat kondisi untuk menghentikannya.

### Pernyataan Goto

Pernyataan goto merupakan intruksi untuk mengarahkan eksekusi program ke pernyataan yang diawali dengan suatu label. Label merupakan pengenal yang diawali denga(:).

Berikut fungsi goto didalam program Pendaftaran SIM:

* Untuk mengulangi pembayaran yang gagal dan input pilihan menu yang tidak valid.

### While

Perulangan while merupakan perulangan untuk Untuk mengulang suatu proses yang belum diketahui jumlahnya. Pengecekan kondisi akan dilakukan terlebih dahulu. Jika kondisi masih bernilai true, maka looping akan terus berlanjut.

Berikut fungsi while didalam program Pendaftar SIM:

* Untuk mencari node dalam linked list.
* Untuk mengecek jenis SIM a, b, dan c
* Digunakan untuk mengulang program selama tidak memilih menu exit.

## Prosedur

Prosedur merupakan suatu program terpisah dalam blok sendiri yang berfungsi sebagai subprogram (progran bagian).

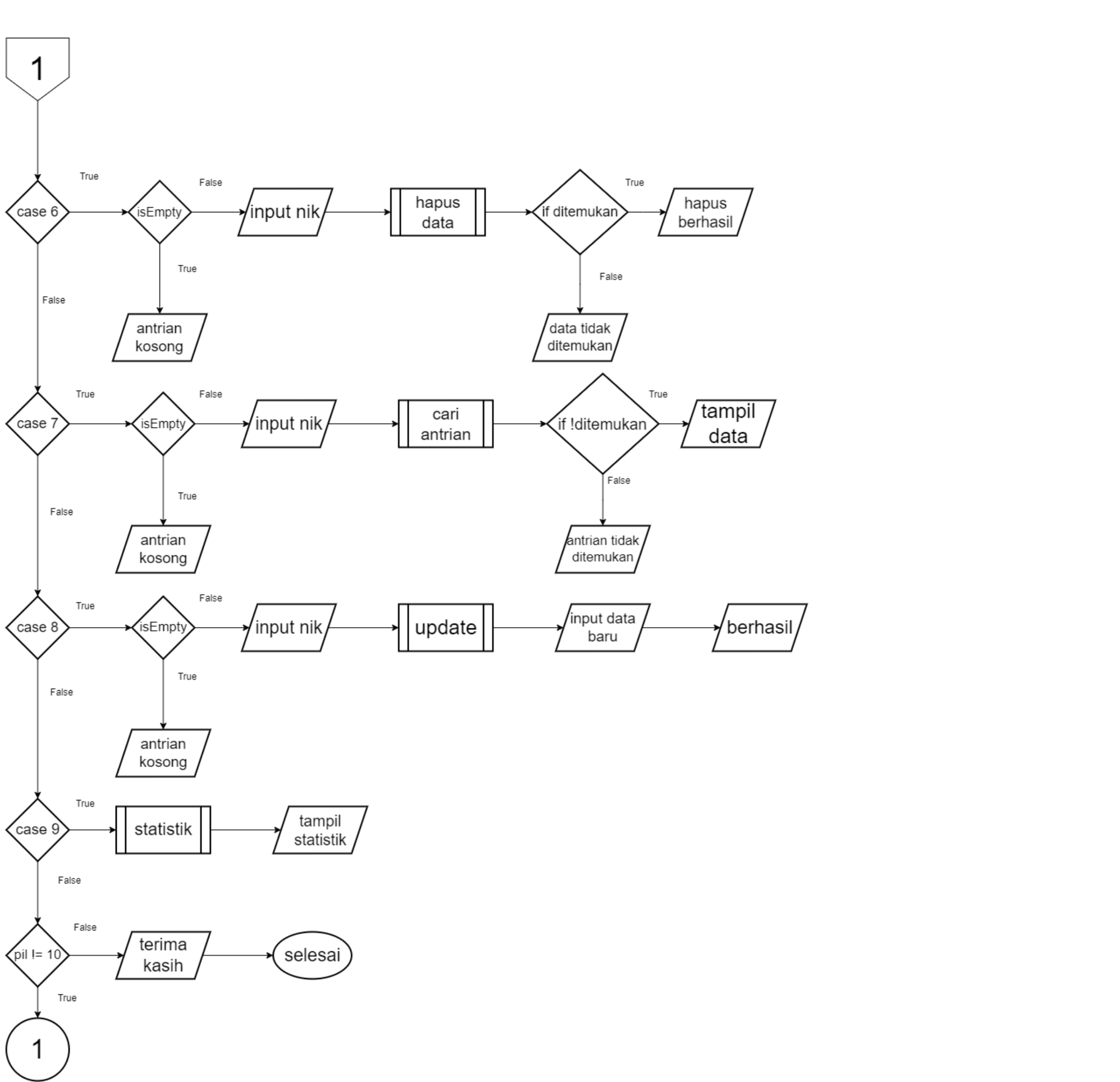
Berikut fungsi prosedur didalam program Pendaftaran SIM:

* Untuk membuat function setiap fitur agar di fungsi main tertata dengan rapih dan mudah dibaca.

# PERANCANGAN

## Flowchart

## 



## Analisis Program

Program pertama kali dijalankan akan muncul tampilan “Selamat dating” dan menu dengan beberapa fitur, lalu program meminta kepada customer untuk memilih beberapa menu.

Menu yang pertama adalah tambah antrian peserta kemudian diarahkan untuk mengisi beberapa data diri untuk dimasukan kedalam SIM seperti nama, nik, alamat, jenis kelamin, jenis SIM, dan nomor hp.Menu yang kedua sama seperti menu yang pertama namun khusus peserta yang mendapatkan VIP.

Menu yang ketiga adalah menu pembayaran yang dimana peserta akan disuruh membayar harga sim sesuai yang ia pilih sebelumnya, jika pembayaran gagal atau uang kurang maka disuruh untuk melakukan pembayaran ulang.

Menu yang keempat adalah menu lihat seluruh antrian. Menu yang kelima sama seperti menu yang pertama namun hanya antrian terdepan saja.

Menu yang enam adalah menu menghapus antrian jika peserta berhalangan hadir atau mengundurkan diri. Pertama program meninta menginput nik yang ingin di hapus, jika sudah maka data tersebut secara otomatis akan terhapus.

Menu yang ke tujuh adalah menu mencari data. Ketika customer ingin mencari data cukup menginput nik atau nomor hp, setelah itu data akan ditampilkan jika ketemu, jika tidak maka data tidak ditemukan.

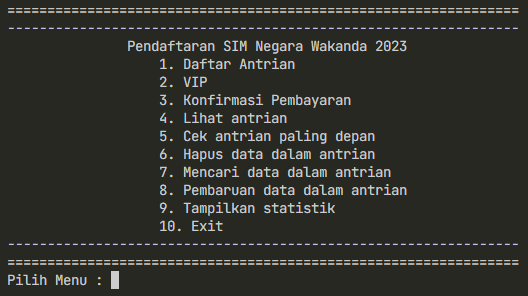
Menu yang ke delapan adalah menu update data, jika ada peserta yang ingin memperbarui data diri nya, mungkin karena kesalahan dari nama, nik dll, maka cukup masukan nik dari input sebelumnya lalu masukan ulang data diri peserta Kembali.

Menu yang ke Sembilan adalah menu untuk menampilkan statistic dari jenis kelamin dan jenis SIM.

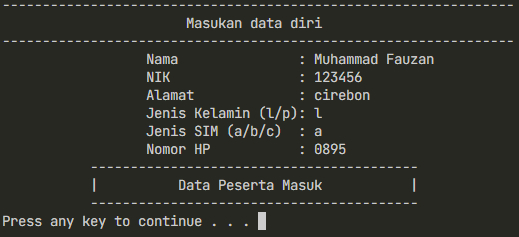
Dan yang terakhir menu ke sepuluh adalah keluar dari program dan sebelum keluar program akan menampilkan kalimat “Terima kasih”.

# HASIL DAN PEMBAHASAN

## Implementasi Program/Tutorial

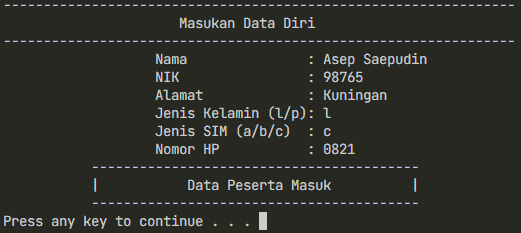


Pertama program akan menampilkan menu seperti gambar diatas. Lalu pilih menu, misal kita pilih menu pertama, ketik 1 kemudian aka nada perintah untuk mengisi data diri.

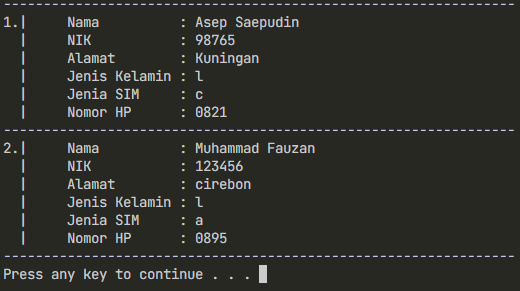


Setelah mengisi data diri maka akan diberitahu bahwa data telah masuk. Pada gambar diatas ada perintah “Press any key to continue…” itu kita disuruh tekan bebas tombol keyboard (akan sering muncul dibeberapa menu), Jika sudah maka program Kembali ke menu awal.

Oke selanjutnya kita coba menu yang kedua, kita klik 2.

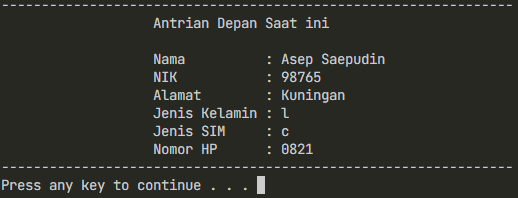


Sama seperti pada menu ke satu namun pada menu kedua peserta dimasukan kedalam antrian pertama karena VIP seperti gambar diatas. Untuk membuktikannya kita pilih menu ketiga untuk melihat antrian saat ini.



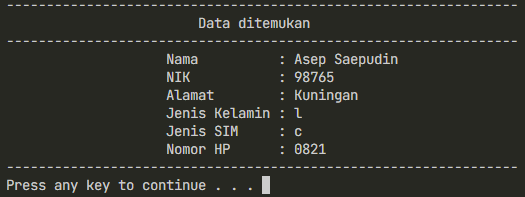
Terbukti ya yang awalnya kita input Muhammad Fauzan tetapi yang berada diposisi pertama yaitu Asep Saepuding karena menggunakan menu VIP.

Kita masuk ke menu berikutnya, menu ke 5. Menu ke-5 sama seperti menu ke 4 namun pada menu ke 5 hanya menampilkan antrian pertama saja.

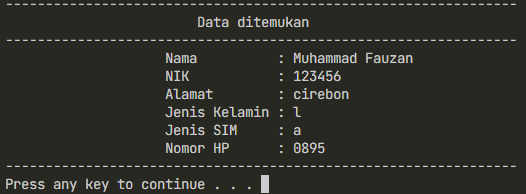


Setelah kita lihat antrian, misalkan kita data seseorang namun karena antrian yang terlalu Panjang sehingga kesulitan untuk mencarinya. Tenang solusinya kita pilih menu ke 7 yaitu menacari data dalam antrian, dengan menu ini kita bisa dengan cepat mencari data yang sedang kita cari, cukup kita masukan nik atau nomor hp.

Oke kita coba cari data dengan nik terlebih dahulu, kita cari 98765.

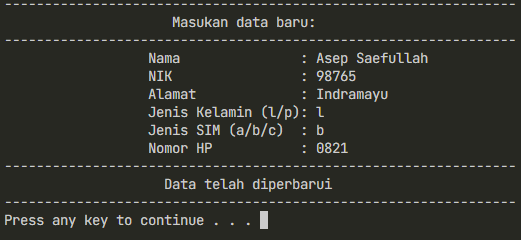
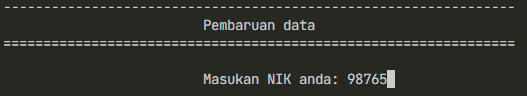


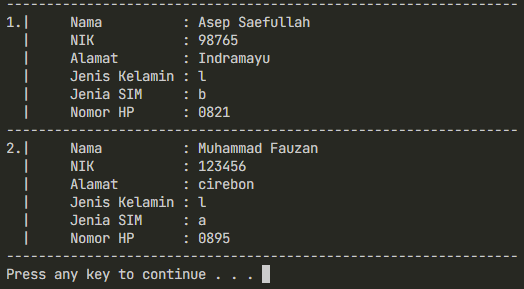
Exclent! Kita berhasil data ditemukan dan ditampilkan dilayar. Bagaimana mencari dengan nomor hp? Oke kita coba juga, kita input 0895.



Sama ya berhasil, jadi tidak perlu khawatir jika lupa NIK, masih bisa dengan nomor hp.

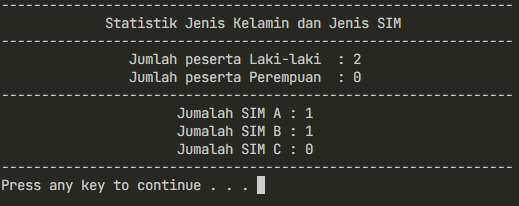
Bagaimana jika kita salah masukan data diri, jangan khawatir kita ada menu update data/ pembaruan data di menu ke 8. Missal kita ingin mengganti data dengan nik 98765.



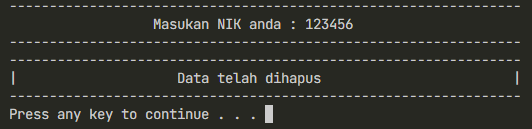
Program akan menampilkan seperti Digambar, setelah data dinput maka data peserta antrian telah diperbarui, mari kita lihat hasilnya.

Data telah diperbarui, berfungsi dengan baik ya.

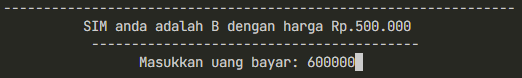
Pada menu berikutnya ada menu statistic dinomor 9. Ini tampilannya.

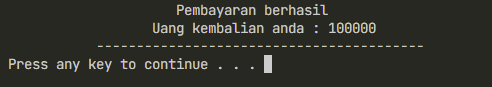


Bagaimana jika ada beberapa peserta yang ingin mengundurkan diri? Kita ada fitur hapus data dimenu ke 6. Cukup masukan nik yang ingin dihapus missal kita masukan nik 123456.

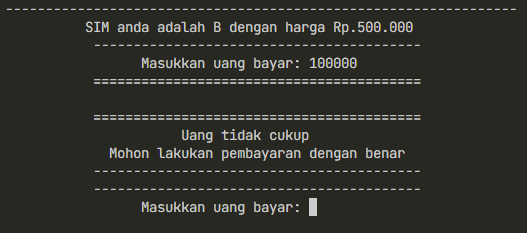
Maka data secara otomatis langsung terhapus.

Baik setelah kita mengenal beberapa fitur menu kita masuk ke menu ke 3 menu pembayaran. Program akan menampilkan harga sim sesuai diinputkan sebelumnya.



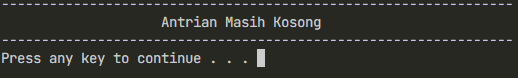


Setelah input bayar maka berhasil namun jika uang nya kurang maka otomatis pembayaran gagal dan disuruh bayar ulang.

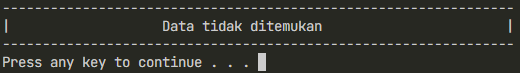


Sebelum ke menu terakkhir ada beberapa hal yang perlu saya sampaikan.

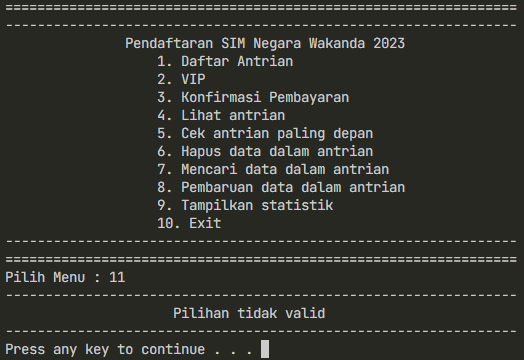
Jika kita memilih menu selain nomor 1 dan 2 tapi kita belum menambahkan antrian maka semua menu selain 1 dan 2 akan menampilkan antrian masih kosong.



Pada menu 6, 7, dan 8 apabila mencari data tidak sesuai dalam antrian maka program akan menampilkan “Data tidak ditemukan.”

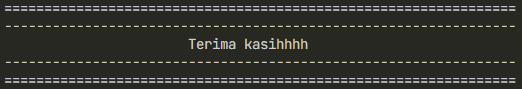


Dan terakhir jika memilih menu namun tidak sesuai pilihan diatas misal kita ketik 11.



Maka ada peringatan bahwa pilihan tidak valid dan program akan menampilkan Kembali menu tersebut.

Menu terakhir yaitu menu ke 10 menu keluar, setelah kita selesai dengan program kita, pilih menu 10 untuk keluar program. Sebelum program ditutup, program menampilkan kalimat “Terima Kasih.”



## Source Code

#include <iostream> // untuk mengaktifkan cout, cin dan endl

#include <string.h> // untuk mengaktifkan tipe data karakter

using namespace std;

// Node dalam linked list

struct Node{

string nama, nik, alamat, jenisKelamin, jenisSim, noHp; //deklarasi tipe data string/ karakter

Node \*next; // deklarasi pointer next

};

struct queue{

Node \*front;

Node \*rear;

};

Node \*head, \*tail, \*front, \*rear; // deklarasi variabel pointer head dan tail

// inisialisasi awal bahwa head dan tail bernilai NULL

void init(){

head = front = tail = rear = NULL; //deklarasi variabel pointer bernilai NULL

}

//pengecekan apakah antrian kosong atau tidak?

bool isEmpty(){

return tail == NULL ; //deklarasi tail sebagai NULL

}

// tambah antrian depan(VIP)

void insertDepan(string nama, string nik, string alamat, string jenisKelamin, string jenisSim, string noHp){

// pembuatan node baru

Node \*baru;

baru = new Node;

// input data node dari hasil input manual menggunakan parameter

baru->nama = nama;

baru->nik = nik;

baru->alamat = alamat;

baru->jenisKelamin = jenisKelamin;

baru->jenisSim = jenisSim;

baru->noHp = noHp;

// pengkondisian antrian kosong atau tidak

if(isEmpty()){

head = tail = baru; //menjadikan head dan tail berada di node baru

tail->next = NULL; //tail yang menunjuk ke node selanjutnya adalah null

}else{

baru->next = head; // memasukan node baru ke antrian depan

head = baru; // deklarasi head berada di node baru

}

// cout << "----------------------------------------------------------------\n";

cout << " -----------------------------------------\n";

cout << " | Data Peserta Masuk |\n";

cout << " -----------------------------------------\n";

}

// tambah belakang(Regular)

void insertBelakang(string nama, string nik, string alamat, string jenisKelamin, string jenisSim, string noHp){

//Membuat Node baru

Node \*baru;

baru = new Node;

// input data node dari hasil input manual menggunakan parameter

baru->nama = nama;

baru->nik = nik;

baru->alamat = alamat;

baru->jenisKelamin = jenisKelamin;

baru->jenisSim = jenisSim;

baru->noHp = noHp;

// pengkondisian antrian kosong atau tidak

if(isEmpty()){

head = tail = baru;

tail->next = NULL; //tail menunjuk node selanjutnya bernilai NULL

}else{

tail->next = baru; //tail menunjuk node selanjutnya bernilai node baru

tail = baru; // posisi tail dipindahkan ke node baru

}

cout << " -----------------------------------------\n";

cout << " | Data Peserta Masuk |\n";

cout << " -----------------------------------------\n";

}

// proses penambahan antrian dalam queue

void enqueue(string nama, string nik, string alamat, string jenisKelamin, string jenisSim, string noHp){

insertBelakang(nama, nik, alamat, jenisKelamin, jenisSim, noHp);

front = head; //front diposisikan berada dihead

rear = tail; //rear diposisikan berada ditail

}

// proses pengambilan antrian dalam queue

void enqueueDepan(string nama, string nik, string alamat, string jenisKelamin, string jenisSim, string noHp){

insertDepan(nama, nik, alamat, jenisKelamin, jenisSim, noHp);

front = head;

rear = tail;

}

// pembayaran sekaligus antrian depan dihapus

void pembayaran() {

// deklarasi variabel pointer hapus berada dihead

Node \*hapus = head;

int bayar; //deklarasi variabel integer

cout << "----------------------------------------------------------------\n";

// pengecekkan jenis sim

if (hapus->jenisSim == "a") {

cout << " SIM anda adalah A dengan harga Rp.750.000" << endl;

} else if (hapus->jenisSim == "b") {

cout << " SIM anda adalah B dengan harga Rp.500.000" << endl;

} else if (hapus->jenisSim == "c") {

cout << " SIM anda adalah C dengan harga Rp.300.000" << endl;

} else {

cout << " Jenis SIM tidak valid" << endl;

return;

}

bayar: // untuk mengarahkan eksekusi program dari goto bayar menuju label bayar;

cout << " -----------------------------------------\n";

cout << " Masukkan uang bayar: ";

cin >> bayar;

cout << " =========================================\n\n";

cout << " =========================================\n";

// jika sim a dan uang bayar lebih besar sama dengan 750.000

if (hapus->jenisSim == "a" && bayar >= 750000) {

int uangKembali = bayar - 750000; //deklarasi type data variabel uangKembali adalah integer dan diberi nilai input bayar dikurangi 750.000

system("cls"); //clear tampilan

cout << " -----------------------------------------\n";

cout << " Pembayaran berhasil" << endl;

cout << " Uang kembalian anda : " << uangKembali << endl;

cout << " -----------------------------------------\n";

if(head == tail){ // jika node hanya satu

head = tail = NULL;

return;

}

head = head->next; // memindahkan head ke node selanjutnya

delete hapus; //menghapus node

} else if (hapus->jenisSim == "b" && bayar >= 500000) { // jika sim b dan uang bayar lebih besar sama dengan 500.000

int uangKembali = bayar - 500000; //deklarasi type data variabel uangKembali adalah integer dan diberi nilai input bayar dikurangi 500.000

system("cls");

cout << " Pembayaran berhasil" << endl;

cout << " Uang kembalian anda : " << uangKembali << endl;

cout << " -----------------------------------------\n";

if(head == tail){

head = tail = NULL;

return;

}

head = head->next;

delete hapus;

} else if (hapus->jenisSim == "c" && bayar >= 300000) { // jika sim c dan uang bayar lebih besar sama dengan 300.000

int uangKembali = bayar - 300000; //deklarasi type data variabel uangKembali adalah integer dan diberi nilai input bayar dikurangi 300.000

system("cls");

cout << " Pembayaran berhasil" << endl;

cout << " Uang kembalian anda : " << uangKembali << endl;

cout << " -----------------------------------------\n";

if(head == tail){

head = tail = NULL;

return;

}

head = head->next;

delete hapus;

} else {

cout << " Uang tidak cukup\n";

cout << " Mohon lakukan pembayaran dengan benar" << endl;

cout << " -----------------------------------------\n";

goto bayar;

}

}

//operasi pengambilan antrian dari queue

void dequeue(){

if(isEmpty()){

cout << " -----------------------------------------\n";

cout << " | Antrian masih kosong |\n";

cout << " -----------------------------------------\n";

}else{

pembayaran();

front = head; //front diposisikan dihead

}

}

//lihat antrian

void lihatAntrian(){

Node \*bantu = head; //deklarasi pointer bantu dengan diposisikan dihead

int count = 1; // deklarasi variabel count diberi nilai 1

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian Masih Kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

// melakukan iterasi ke setiap node menggunakan pointer bantu, selama bantu tidak sama dengan NULL maka iterasi tetap dilakukan

while(bantu != NULL){

cout << "----------------------------------------------------------------\n";

cout << count++ << ".|" << "\t"<< "Nama : " << bantu->nama << endl; //increment count untuk nomor antrian

cout << " |" << "\t" << "NIK : " << bantu->nik << endl;

cout << " |" << "\t" << "Alamat : " << bantu->alamat << endl;

cout << " |" << "\t" << "Jenis Kelamin : " << bantu->jenisKelamin << endl;

cout << " |" << "\t" << "Jenia SIM : " << bantu->jenisSim << endl;

cout << " |" << "\t" << "Nomor HP : " << bantu->noHp << endl;

bantu = bantu->next;

}

cout << "----------------------------------------------------------------\n";

}

}

// antran paling depan

void lihatAntrianDepan(){

if (isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian Masih Kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

cout << "----------------------------------------------------------------\n";

cout << " Antrian Depan Saat ini\n\n"; // melihat antrian paling depan dengan head menunjuk ke data peserta

cout << " Nama : " << front->nama << endl;

cout << " NIK : " << front->nik << endl;

cout << " Alamat : " << front->alamat << endl;

cout << " Jenis Kelamin : " << front->jenisKelamin << endl;

cout << " Jenis SIM : " << front->jenisSim << endl;

cout << " Nomor HP : " << front->noHp << endl;

cout << "----------------------------------------------------------------\n";

}

}

// hapus data dengan nik

void deleteNik(string inputNik){

Node \*hapus = head;

Node \*bantu = head;

Node \*sebelum = NULL; // Menyimpan node sebelumnya dengan deklarasi awal adalah null

bool ditemukan = false; //deklarasi variabel ditemukan dengan tipe data bool yang bernilai false

if(head == tail){ //kasus ketika antrian hanya 1

if(hapus->nik == inputNik){

head = tail = NULL;

cout << "----------------------------------------------------------------\n";

cout << "| Data telah dihapus |\n";

cout << "----------------------------------------------------------------\n";

return;

}else{

cout << "----------------------------------------------------------------\n";

cout << "| Data tidak ditmukan |\n";

cout << "----------------------------------------------------------------\n";

return;

}

}

while(bantu != NULL){

if(bantu->nik == inputNik){

hapus = bantu;

ditemukan = true;

break;

}

sebelum = bantu;

bantu = bantu->next;

}

if(ditemukan){

if(hapus == head){ // kasus ketika hapus adalah head / hapus bagian depan

head = head->next;

delete hapus;

}else if(hapus == tail){ // Kasus ketika hapus adalah tail / hapus bagian belakang

tail = sebelum;

tail->next = NULL;

delete hapus;

}else{ // Kasus ketika hapus berada di tengah / hapus bagian tengah

bantu = hapus->next;

sebelum->next = bantu;

delete hapus;

}

cout << "----------------------------------------------------------------\n";

cout << "| Data telah dihapus |\n";

cout << "----------------------------------------------------------------\n";

}else{

cout << "----------------------------------------------------------------\n";

cout << "| Data tidak ditemukan |\n";

cout << "----------------------------------------------------------------\n";

}

}

// mencari data dengan nik atau nomor hp menggunakan input manual

void search(string input){

Node \*bantu = head;

bool ditemukan = false;

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian masih kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

while(bantu != NULL){

if(bantu->nik == input || bantu->noHp == input){

cout << "----------------------------------------------------------------\n";

cout << " Data ditemukan\n";

cout << "----------------------------------------------------------------\n";

cout << " Nama : " << bantu->nama << endl;

cout << " NIK : " << bantu->nik << endl;

cout << " Alamat : " << bantu->alamat << endl;

cout << " Jenis Kelamin : " << bantu->jenisKelamin << endl;

cout << " Jenis SIM : " << bantu->jenisSim << endl;

cout << " Nomor HP : " << bantu->noHp << endl;

cout << "----------------------------------------------------------------\n";

ditemukan = true;

return;

}

bantu = bantu->next;

}

if(!ditemukan){ // jika ditemukan bernilai tidak true

cout << "----------------------------------------------------------------\n";

cout << "| Data tidak ditemukan |\n";

cout << "----------------------------------------------------------------\n";

}

}

}

// pembaruan data dengan input nik

void pembaruanData(string inputNik){

Node \*bantu = head;

bool ditemukan = false;

//melakukan iterasi selama bantu tidak null, maka iterasi berlanjut

while(bantu != NULL){

//kasus bantu menunjuk nik sama dengan nik yang diinput

if(bantu->nik == inputNik){

cout << "----------------------------------------------------------------\n";

cout << " Masukan data baru: \n";

cout << "----------------------------------------------------------------\n";

cin.ignore();

cout << " Nama : "; getline(cin, bantu->nama);

cout << " NIK : "; getline(cin, bantu->nik);

cout << " Alamat : "; getline(cin, bantu->alamat);

cout << " Jenis Kelamin (l/p): "; getline(cin, bantu->jenisKelamin);

cout << " Jenis SIM (a/b/c) : "; getline(cin, bantu->jenisSim);

//ketika sim bukan a,b, dan c, maka input ulang

while(bantu->jenisSim != "a" && bantu->jenisSim != "b" && bantu->jenisSim != "c"){

cout << "----------------------------------------------------------------\n";

cout << " Maaf jenis tidak valid\n";

cout << "----------------------------------------------------------------\n";

cout << " Jenis SIM (a/b/c) : "; getline(cin, bantu->jenisSim);

}

cout << " Nomor HP : "; getline(cin, bantu->noHp);

cout << "----------------------------------------------------------------\n";

cout << " Data telah diperbarui\n";

cout << "----------------------------------------------------------------\n";

ditemukan = true; // mengubah nilai ditemukkan menjadi true

break;

}

bantu = bantu->next;

}

// jika ditemukan tidak bernilai true

if(!ditemukan){

cout << "----------------------------------------------------------------\n";

cout << " Data tidak ditemukan\n";

cout << "----------------------------------------------------------------\n";

}

}

// lihat data statistik dari jenis kelamin dan jenis sim

void statistik(){

Node \*bantu = head;

int jumlahLakilaki = 0;

int jumlahPerempuan = 0;

int jumlahSimA = 0;

int jumlahSimB = 0;

int jumlahSimC = 0;

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian masih kosong\n";

cout << "----------------------------------------------------------------\n";

return;

}else{

while(bantu != NULL){

//kasus jika jenis kelamin ada laki-laki atau perempuan, maka increment

if(bantu->jenisKelamin == "l"){

jumlahLakilaki++;

}else if(bantu->jenisKelamin == "p"){

jumlahPerempuan++;

}

//kasus jika jenis sim ada a, b, dan c, maka increment

if(bantu->jenisSim == "a"){

jumlahSimA++;

}else if(bantu->jenisSim == "b"){

jumlahSimB++;

}else if (bantu->jenisSim == "c"){

jumlahSimC++;

}

bantu = bantu->next;

}

}

cout << "----------------------------------------------------------------\n";

cout << " Statistik Jenis Kelamin dan Jenis SIM\n";

cout << "----------------------------------------------------------------\n";

cout << " Jumlah peserta Laki-laki : " << jumlahLakilaki << endl;

cout << " Jumlah peserta Perempuan : " << jumlahPerempuan << endl;

cout << "----------------------------------------------------------------\n";

cout << " Jumalah SIM A : " << jumlahSimA << endl;

cout << " Jumalah SIM B : " << jumlahSimB << endl;

cout << " Jumalah SIM C : " << jumlahSimC << endl;

cout << "----------------------------------------------------------------\n";

}

// program utama / awal program berjalan

int main(){

int pilihan = 0;

string nama, nik, alamat, jenisKelamin, jenisSim, noHp, inputNik, input;

// ketika pilihan adalah 10 maka berhenti

while(pilihan != 10){

cout << "================================================================\n";

cout << "----------------------------------------------------------------\n";

cout << "\t Pendaftaran SIM Negara Wakanda 2023\n";

cout << " 1. Daftar Antrian\n";

cout << " 2. VIP\n";

cout << " 3. Konfirmasi Pembayaran\n";

cout << " 4. Lihat antrian\n";

cout << " 5. Cek antrian paling depan\n";

cout << " 6. Hapus data dalam antrian\n";

cout << " 7. Mencari data dalam antrian\n";

cout << " 8. Pembaruan data dalam antrian\n";

cout << " 9. Tampilkan statistik\n";

cout << " 10. Exit\n";

cout << "----------------------------------------------------------------\n";

cout << "================================================================\n";

cout << "Pilih Menu : ";

cin >> pilihan;

// pencocokan input pilihan yang diminta

switch (pilihan){

case 1:

system("cls");

cout << "----------------------------------------------------------------\n";

cout << " Masukan data diri\n";

cout << "----------------------------------------------------------------\n";

// berfungsi untuk membersihkan karakter newline (\n) dari buffer input

cin.ignore();

//getline berfungsi untuk katakter yang mempunyai spasi

cout << " Nama : "; getline(cin, nama);

cout << " NIK : "; getline(cin, nik);

cout << " Alamat : "; getline(cin, alamat);

cout << " Jenis Kelamin (l/p): "; getline(cin, jenisKelamin);

cout << " Jenis SIM (a/b/c) : "; getline(cin, jenisSim);

// melakukan iterasi selama sim bukan a, b, dan c

while(jenisSim != "a" && jenisSim != "b" && jenisSim != "c"){

cout << "----------------------------------------------------------------\n";

cout << " Maaf jenis tidak valid\n";

cout << "----------------------------------------------------------------\n";

cout << " Jenis SIM : "; getline(cin, jenisSim);

}

cout << " Nomor HP : "; getline(cin, noHp);

enqueue(nama, nik, alamat, jenisKelamin, jenisSim, noHp);

system("pause");//untuk menahan layar

system("cls");//untuk clean layar

break;

case 2:

system("cls");

cout << "----------------------------------------------------------------\n";

cout << " Masukan Data Diri\n";

cout << "----------------------------------------------------------------\n";

cin.ignore();

cout << " Nama : "; getline(cin, nama);

cout << " NIK : "; getline(cin, nik);

cout << " Alamat : "; getline(cin, alamat);

cout << " Jenis Kelamin (l/p): "; getline(cin, jenisKelamin);

cout << " Jenis SIM (a/b/c) : "; getline(cin, jenisSim);

while(jenisSim != "a" && jenisSim != "b" && jenisSim != "c"){

cout << "----------------------------------------------------------------\n";

cout << " Maaf jenis tidak valid\n";

cout << "----------------------------------------------------------------\n";

cout << " Jenis SIM : "; getline(cin, jenisSim);

}

cout << " Nomor HP : "; getline(cin, noHp);

enqueueDepan(nama, nik, alamat, jenisKelamin, jenisSim, noHp);

system("pause");

system("cls");

break;

case 3:

system("cls");

dequeue();

system("pause");

system("cls");

break;

case 4:

system("cls");

lihatAntrian();

system("pause");

system("cls");

break;

case 5:

system("cls");

lihatAntrianDepan();

system("pause");

system("cls");

break;

case 6:

system("cls");

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian masih Kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

cout << "----------------------------------------------------------------\n";

cout << " Masukan NIK anda : "; cin >> inputNik;

cout << "----------------------------------------------------------------\n";

system("cls");

deleteNik(inputNik);

}

system("pause");

system("cls");

break;

case 7:

system("cls");

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian masih Kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

cout << "Masukan NIK/Nomor HP : ";

cin >> input;

system("cls");

search(input);

}

system("pause");

system("cls");

break;

case 8:

system("cls");

if(isEmpty()){

cout << "----------------------------------------------------------------\n";

cout << " Antrian masih Kosong\n";

cout << "----------------------------------------------------------------\n";

}else{

cout << "----------------------------------------------------------------\n";

cout << " Pembaruan data\n";

cout << "================================================================\n\n";

cout << " Masukan NIK anda: ";

cin >> inputNik;

system("cls");

pembaruanData(inputNik);

}

system("pause");

system("cls");

break;

case 9:

system("cls");

statistik();

system("pause");

system("cls");

break;

case 10:

system("cls");

cout << "================================================================\n";

cout << "----------------------------------------------------------------\n";

cout << " Terima kasihhhh\n";

cout << "----------------------------------------------------------------\n";

cout << "================================================================\n";

break;

default:

cout << "----------------------------------------------------------------\n";

cout << " Pilihan tidak valid\n";

cout << "----------------------------------------------------------------\n";

system("pause");

system("cls");

}

}

return 0;

}

# PENUTUP

## Kesimpulan

Setelah menyelesaikan pembuatan aplikasi pembantu dalam proses pengurusan Surat Izin Mengemudi (SIM) di negara Wakanda, dapat disimpulkan bahwa aplikasi ini memberikan solusi yang efektif dan efisien bagi pemerintah dalam mengelola data pengemudi dan administrasi SIM.

Aplikasi ini diharapkan dapat mempermudah penduduk Wakanda dalam proses pengurusan SIM, meningkatkan efisiensi pemerintah dalam mengelola data penduduk terkait SIM, serta meningkatkan keamanan dan validitas data SIM.

## Saran

Dalam pembuatan aplikasi Surat Izin Mengemudi (SIM) di negara Wakanda, ada beberapa saran sederhana yang perlu diperhatikan. Pertama, lakukan riset yang mendalam untuk memahami kebutuhan dan preferensi calon pengemudi serta pihak berwenang terkait.

Dengan memahami kebutuhan pengguna, aplikasi dapat dirancang sesuai dengan kebutuhan mereka. Kedua, desainlah antarmuka aplikasi dengan user-friendly, sehingga mudah dipahami dan digunakan oleh pengguna. Penggunaan ikon dan tampilan yang intuitif dapat membantu meningkatkan kenyamanan pengguna dalam menggunakan aplikasi.

Ketiga, perhatikan keamanan data pribadi calon pengemudi dengan baik. Pastikan adanya langkah-langkah keamanan yang kuat untuk melindungi informasi sensitif mereka. Keempat, sebelum diluncurkan secara resmi, lakukan uji coba aplikasi dalam skala kecil terlebih dahulu. Uji coba ini membantu mengidentifikasi dan memperbaiki potensi masalah sebelum aplikasi digunakan secara luas.

Terakhir, lakukan evaluasi berkala terhadap kinerja aplikasi dan lakukan perbaikan dan pemeliharaan secara rutin. Dengan menerapkan saran-saran sederhana ini, diharapkan aplikasi Surat Izin Mengemudi di negara Wakanda dapat berjalan dengan baik, meningkatkan efisiensi pengurusan SIM, dan memberikan manfaat yang signifikan bagi para pengemudi dan pemerintah.