



LAPORAN PROJEK AKHIR

JUDUL PROGRAM

PENDATAAN PENDUDUK DI WILAYAH KOTA SAMARINDA

MATAKULIAH:

ALGORITMA PEMROGRAMAN LANJUT

Disusun Oleh:

Kelompok / A 4 / 2022

Aziizah Oki Shofrina / 2109106004 (Ketua)

Chintia Liu Wintin / 2109106008

Muhammad Fikri / 2109106010

Bramantyo Ardi Harimurti Wibisono / 2109106013

Asisten Lab (Pembimbing):

Fayza Virdana Addiza

NIM : 2009106004

**PROGRAM STUDI INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI DAN KOMUNIKASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA
2022**

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Tuhan Yang Maha Esa yang telah menganugerahkan banyak nikmat sehingga kami dapat menyusun laporan projek akhir ini dengan baik. Pada program ini, kami mengambil tema tentang Pendataan Penduduk di Wilayah Kota Samarinda yang dapat digunakan dalam bidang pemerintahan untuk mengelola data penduduk.

Adapun tujuan kami membuat program ini adalah untuk memenuhi nilai kami dalam mengikuti projek akhir praktikum algoritma dan pemrograman lanjut. Selain itu, kami juga berharap dengan dibuatnya program sederhana ini, dapat mempermudah kegiatan perancangan administrasi dalam mengelola data penduduk khususnya di Wilayah Samarinda.

Laporan ini kami susun secara cepat dengan bantuan dan dukungan berbagai pihak diantaranya; Ibu Ir Novianti Puspitasari, S.Kom., M. Eng. dan Bapak Awang Harsa Kridalaksana, M.Kom. selaku dosen mata kuliah Algoritma Pemrograman Dasar, dan Mba Fayza Virdana Addiza selaku asisten lab / pembimbing kami. Oleh karena itu kami sampaikan terima kasih atas waktu, tenaga dan pikirannya yang telah diberikan.

Dalam penyusunan laporan ini, kami menyadari bahwa hasil laporan praktikum ini masih jauh dari kata sempurna. Sehingga kami selaku penyusun sangat mengharapkan kritik dan saran yang membangun dari pembaca sekalian.

Akhir kata Semoga laporan praktikum ini dapat memberikan manfaat untuk kelompok kami khususnya, dan masyarakat Indonesia umumnya.

Samarinda, 2 Juni 2022

Penyusun

TAKARIR

Berikut adalah daftar kata Bahasa asing yang telah kami terjemahkan ke dalam Bahasa Indonesia yang kami gunakan dalam program ini :

<i>Program</i>	<i>Program Komputer</i>
<i>Flowchart</i>	<i>Diagram Alir</i>
<i>Sort</i>	<i>Urut</i>
<i>Sorting</i>	<i>Pengurutan</i>
<i>Search</i>	<i>Cari</i>
<i>Searching</i>	<i>Pencarian</i>
<i>Loop</i>	<i>Perulangan</i>
<i>For</i>	<i>Untuk</i>
<i>While</i>	<i>Ketika</i>
<i>Input</i>	<i>Masukkan</i>
<i>Output</i>	<i>Keluaran</i>
<i>Method</i>	<i>Cara</i>
<i>Log In</i>	<i>Masuk</i>
<i>Log Out</i>	<i>Keluar</i>
<i>Sing Up</i>	<i>Daftar</i>
<i>Source Code</i>	<i>Sumber Code</i>
<i>File</i>	<i>Berkas</i>
<i>Eror</i>	<i>Kesalahan</i>
<i>Handling</i>	<i>Penanganan</i>
<i>User</i>	<i>Pengguna</i>
<i>Leveling User</i>	<i>Tingkatan Pengguna</i>
<i>Ascending</i>	<i>Pengurutan Terkecil</i>
<i>Descending</i>	<i>Pengurutan Terbesar</i>

DAFTAR ISI

KATA PENGANTAR	i
TAKARIR.....	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	iv
BAB 1	1
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional	1
1.3 Rumusan Masalah	1
1.4 Batasan Masalah	2
1.5 Tujuan	2
BAB 2	3
PERANCANGAN	3
2.1 Analisis Program.....	3
2.2 Flowchart	3
2.3 Konsep / Materi Praktikum yang Dipakai.....	11
BAB 3	15
HASIL DAN PEMBAHASAN.....	15
3.1 Tampilan Program	15
3.2 Source Code	32
BAB 4	118
KESIMPULAN DAN SARAN.....	118
4.1 Kesimpulan	118
4.2 Saran	118
DAFTAR PUSTAKA	120
LAMPIRAN.....	121

DAFTAR GAMBAR

Gambar 1. 1 Flowchart Utama.....	4
Gambar 1. 2 Flowchart Penduduk.....	5
Gambar 1. 3 Flowchart Pemerintah (Status Hidup, Diagram Data, Hapus)	6
Gambar 1. 4 Flowchart Sorting & Search.....	7
Gambar 1. 5 Flowchart Penduduk Lanjutan	8
Gambar 1. 6 Flowchart Pengisian Data Penduduk.....	9
Gambar 1. 7 Flowchart Ubah Data Penduduk	10
Gambar 2. 4 Tampilan Menu Penduduk	17
Gambar 2. 6 Tampilan Isi Data Jalan, Kecamatan, Kelurahan	18
Gambar 2. 4 Tampilan Hasil Data Diri Penduduk.....	19
Gambar 2. 5 Tampilan Menu Data Diri Penduduk	19
Gambar 2. 6 Tampilan Menu Ubah Data Diri Penduduk.....	20
Gambar 2. 7 Tampilan Menu Ubah Password	20
Gambar 2. 8 Tampilan Konfirmasi Penduduk	21
Gambar 2. 9 Tampilan Hasil Ubah Password Penduduk	21
Gambar 3. 1 Tampilan Menu Login Akun Pemerintah.....	21
Gambar 3. 2 Tampilan Menu Pemerintah	22
Gambar 3. 3 Tampilan Menu Data Penduduk.....	22
Gambar 3. 4 Tampilan Sorting Ascending.....	23
Gambar 3. 5 Tampilan Sorting Descending	23
Gambar 3. 6 Tampilan Sorting NIK Ascending	24
Gambar 3. 7 Tampilan Sorting NIK Descending.....	24
Gambar 3. 8 Tampilan Sorting Tanggal Diperbarui Ascending	25
Gambar 3. 9 Tampilan Sorting Tanggal Diperbarui Descending	25
Gambar 3. 10 Tampilan Searching NIK	26
Gambar 3. 11 Tampilan Hasil Searching NIK.....	26
Gambar 4. 1 Diagram Hasil Pendataan Penduduk	27
Gambar 4. 2 Diagram Kecamatan.....	27
Gambar 4. 3 Diagram Status Hidup	28
Gambar 4. 4 Diagram Jenis Kelamin	28
Gambar 4. 5 Diagram Usia	28
Gambar 4. 6 Diagram Agama	29
Gambar 4. 7 Diagram Golongan Darah	29
Gambar 4. 8 Diagram Status Perkawinan	29
Gambar 5. 1 Tampilan Menu Ubah Status Hidup.....	30
Gambar 5. 2 Tampilan Opsi Ubah Status Hidup	30
Gambar 5. 3 Tampilan Hasil Ubah Status Hidup.....	31
Gambar 6. 1 Tampilan Hapus Data Penduduk.....	31
Gambar 7. 1 Tampilan Program Dihentikan	32

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Program ini awalnya dibuat semata-mata karena tuntunan tugas projek akhir dari praktikum itu sendiri, akan tetapi kami membuat program ini dapat di manfaatkan kedalam dunia pemerintahan. Program ini sangat membantu di dalam bidang pengelolaan data kependudukan. Dengan kemajuan teknologi sekarang yang segala sesuatunya dapat di lakukan melalui media online. Sehingga program ini di buat untuk pendataan penduduk yang dapat digunakan dalam mengelola jumlah penduduk serta mengetahui identitas seperti status hidup penduduk dan sebagainya.

1.2 Kebutuhan Fungsional

Untuk kegunaan dari program ini sendiri ada beberapa fungsi yang dapat kita manfaatkan, diantaranya :

- a) Program menggunakan berbagai atribut data dokumen penduduk.
- b) Program menggunakan metode CRUD untuk mengelola data penduduk.
- c) Program menggunakan sorting dan searching untuk mengakses data penduduk.
- d) Program dapat menampilkan diagram hasil pendataan penduduk.
- e) Program terintegrasi dengan file.txt sebagai database untuk menyimpan data penduduk.
- f) Program memiliki *2 leveling user*, yaitu pemerintah dan penduduk.
- g) Program memiliki *error -handling*.

1.3 Rumusan Masalah

Tema yang kami ambil ini bagaimana agar dunia pemerintahan bisa meningkatkan kualitasnya di bidang teknologi. Dan juga bagaimana agar mempermudah karyawan dalam menginput data penduduk di suatu willyah tersebut

1.4 Batasan Masalah

Program penyusun tentunya tidak sempurna dan masih banyak membutuhkan kritik dan saran oleh Aslab, oleh karena itu kami menjelaskan batasan-batasan dari program yang telah kami buat, yaitu :

- Jangkauan dalam program ini hanya kami atau hanya bisa digunakan di wilayah samarinda.
- Program ini tidak bisa mendeteksi nomor NIK tersebut dari daerah mana.
- Dalam program ini penduduk tidak dapat mengganti statusnya didalam program.
- Sorting dalam program kami ini hanya dapat digunakan berdasarkan Nama, NIK, dan Tanggal Update.
- Searching dalam program ini hanya dapat mensearching NIK saja.

1.5 Tujuan

Tujuan kami membuat program ini adalah untuk memenuhi nilai kami dalam mengikuti projek akhir praktikum algoritma dan pemrograman dasar. Selain itu, kami juga berharap dengan dibuatnya program sederhana ini, pihak kantoran dan karyawan menjadi lebih mudah dalam menentukan gaji yang didapat oleh masing-masing karyawan berdasarkan ketentuan yang berlaku. Sehingga nantinya total gaji yang didapat tidak akan bisa di manipulasi dan akan bersifat transparan tanpa ada kecurangan dari pihak manapun.

BAB 2

PERANCANGAN

2.1 Analisis Program

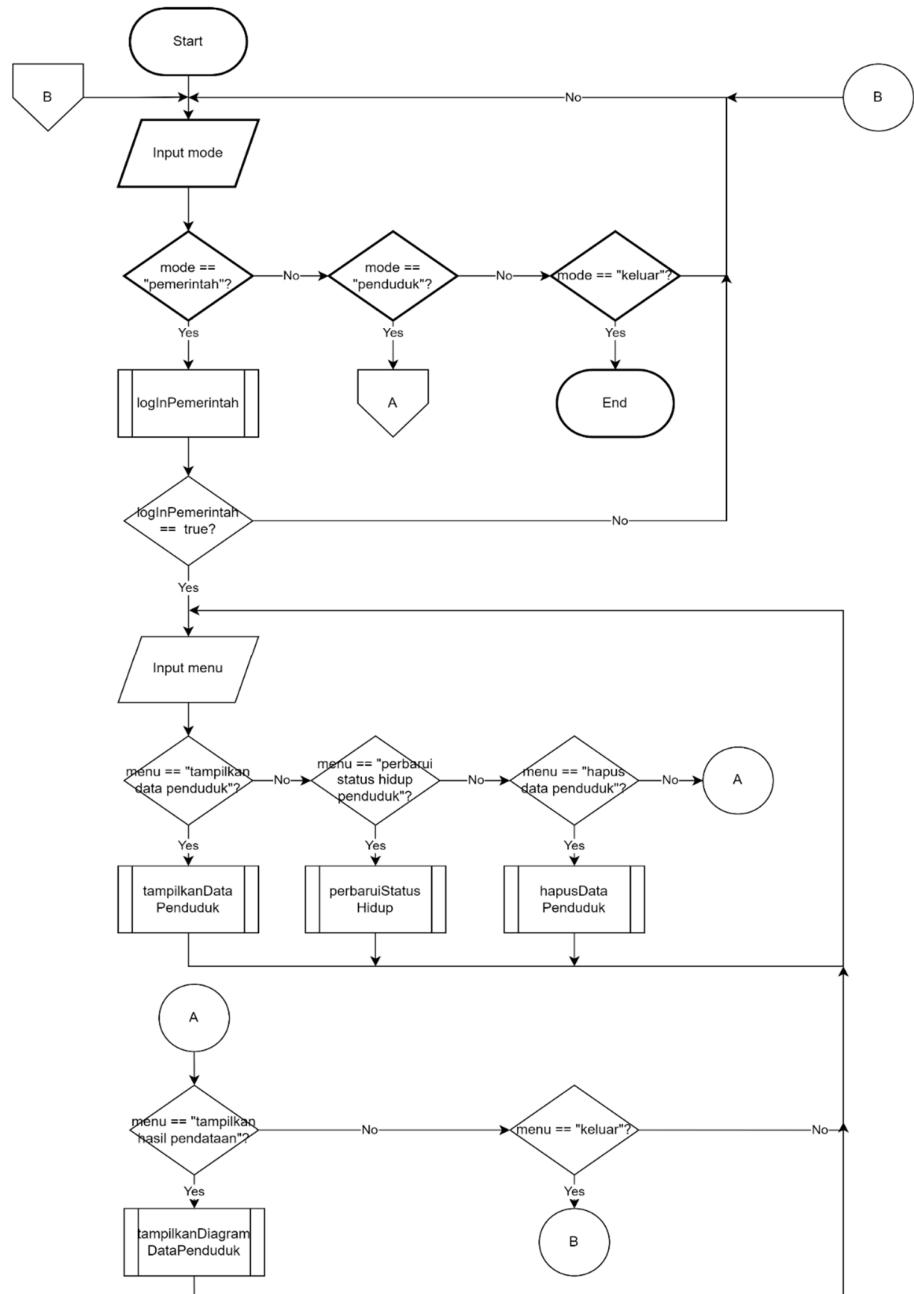
Program ini memiliki alur di dalamnya yang akan memudahkan para audiens dalam memahami arah kerja dari program kami tersebut. Adapun alur dari program yang kami buat akan dijabarkan seperti berikut :

Program yang penyusun rancang ini, berjalan dari beberapa tahap, akan tetapi sebelum masuk ke dalam tahap-tahap tersebut, setelah memulai program anda akan langsung disuguhkan oleh beberapa pilihan, program akan menampilkan pilihan penduduk atau pemerintah. Ketika memilih menu penduduk program akan menampilkan menu lagi yang berisi login atau signup. Selanjutnya user memilih menu sign up untuk membuat akun agar bisa mengisi data nantinya, setelah berhasil membuat akun, user masuk ke menu login. Setelah itu akan berlanjut ke dalam pengisian data . setelah selesai pengisian data user dapat melihat dan mengubah datanya ketika ada yang salah. Selain itu ada menu pemerintah, ketika user masuk ke menu pemerintah , menu pemerintah hasil bisa melihat data penduduk, dan mengubah status penduduk tersebut dari hidup ke wafat. Dan data yang di input user masuk ke dalam file.txt.

2.2 Flowchart

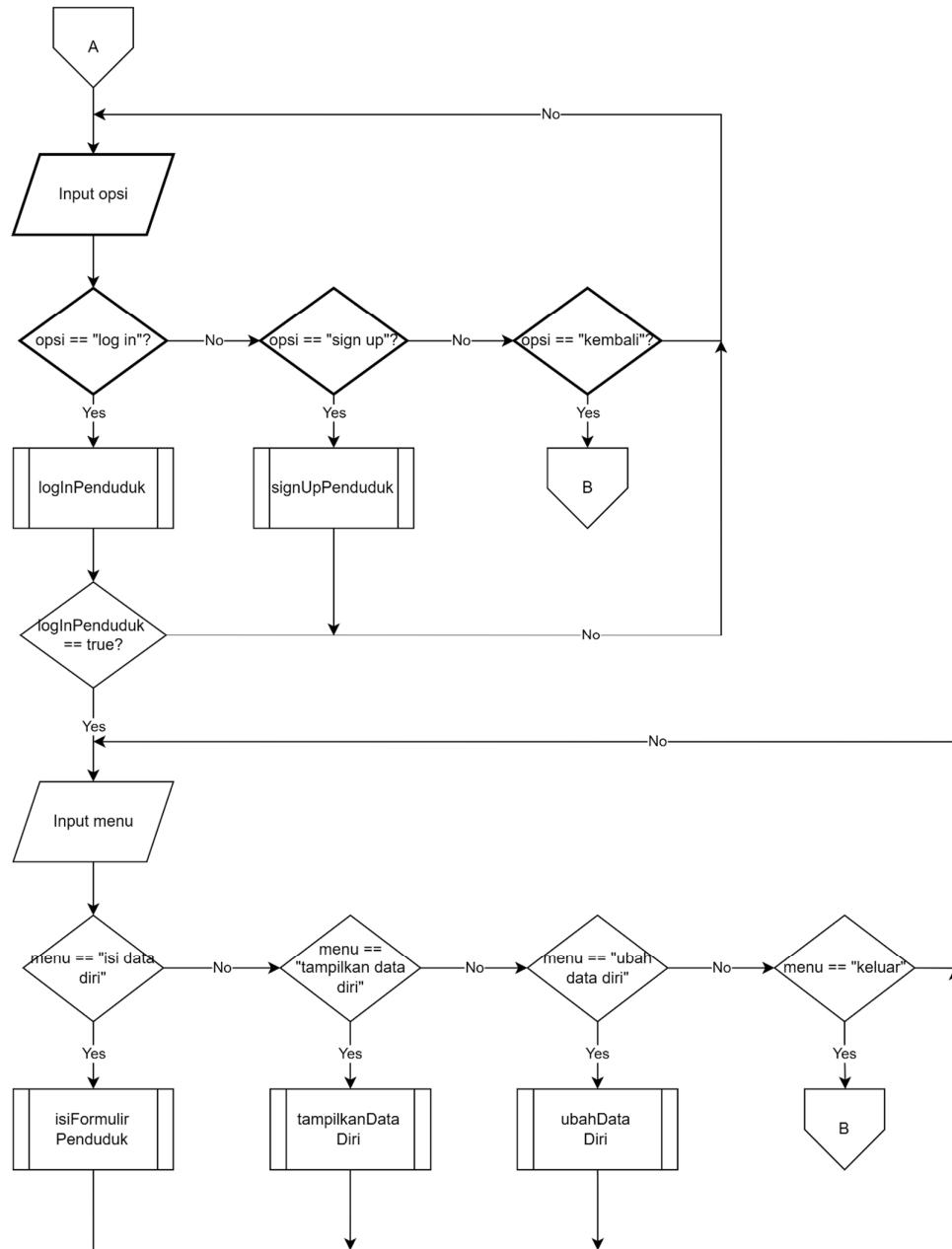
Flowchart yang kami gunakan dalam alur program ini terbagi menjadi 7 gambar flowchart. Hal ini dikarenakan panjangnya alur flowchart yang tidak memungkinkan kami untuk membuat 1 flowchart saja. Untuk gambar tiap flowchart akan kami jabarkan sebagai berikut :

1.) Flowchart Utama



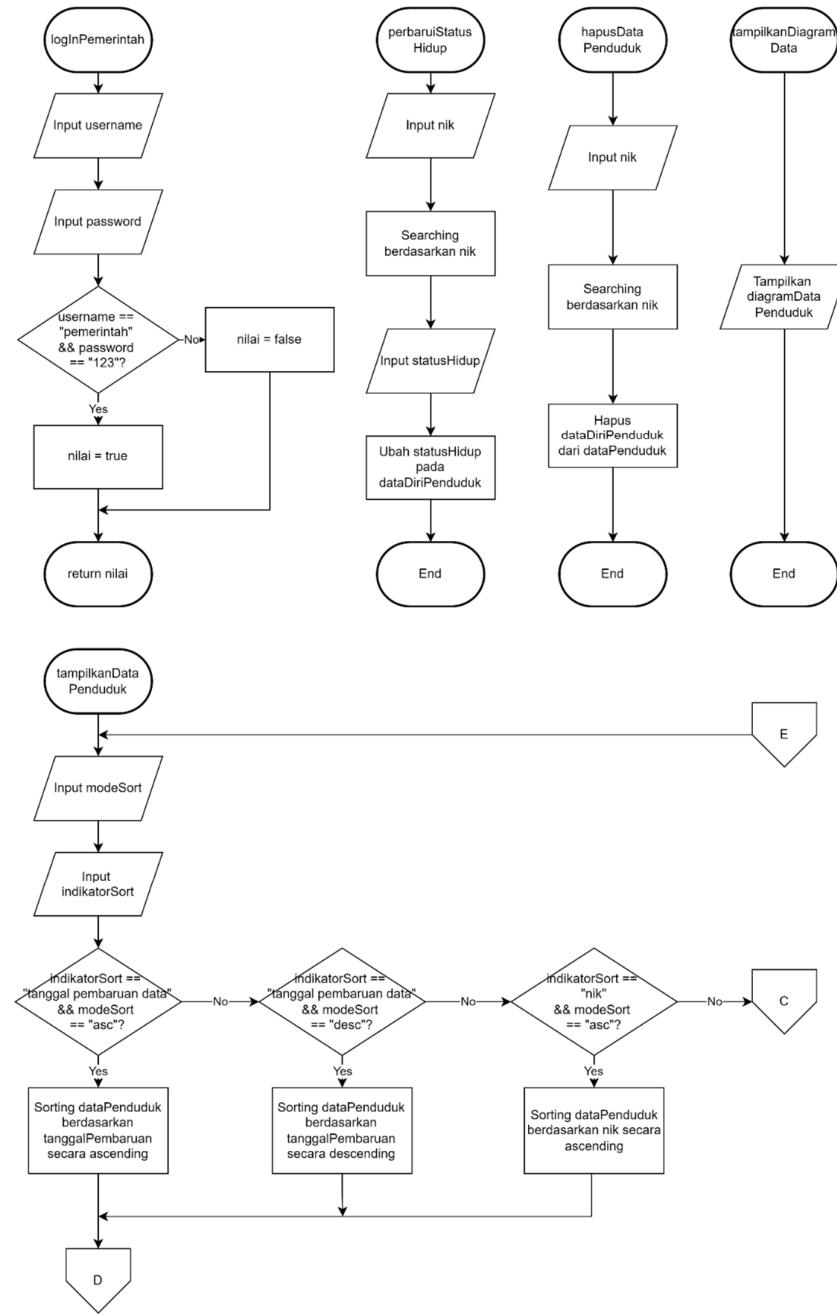
Gambar 1. 1 Flowchart Utama

2.) Flowchart Penduduk



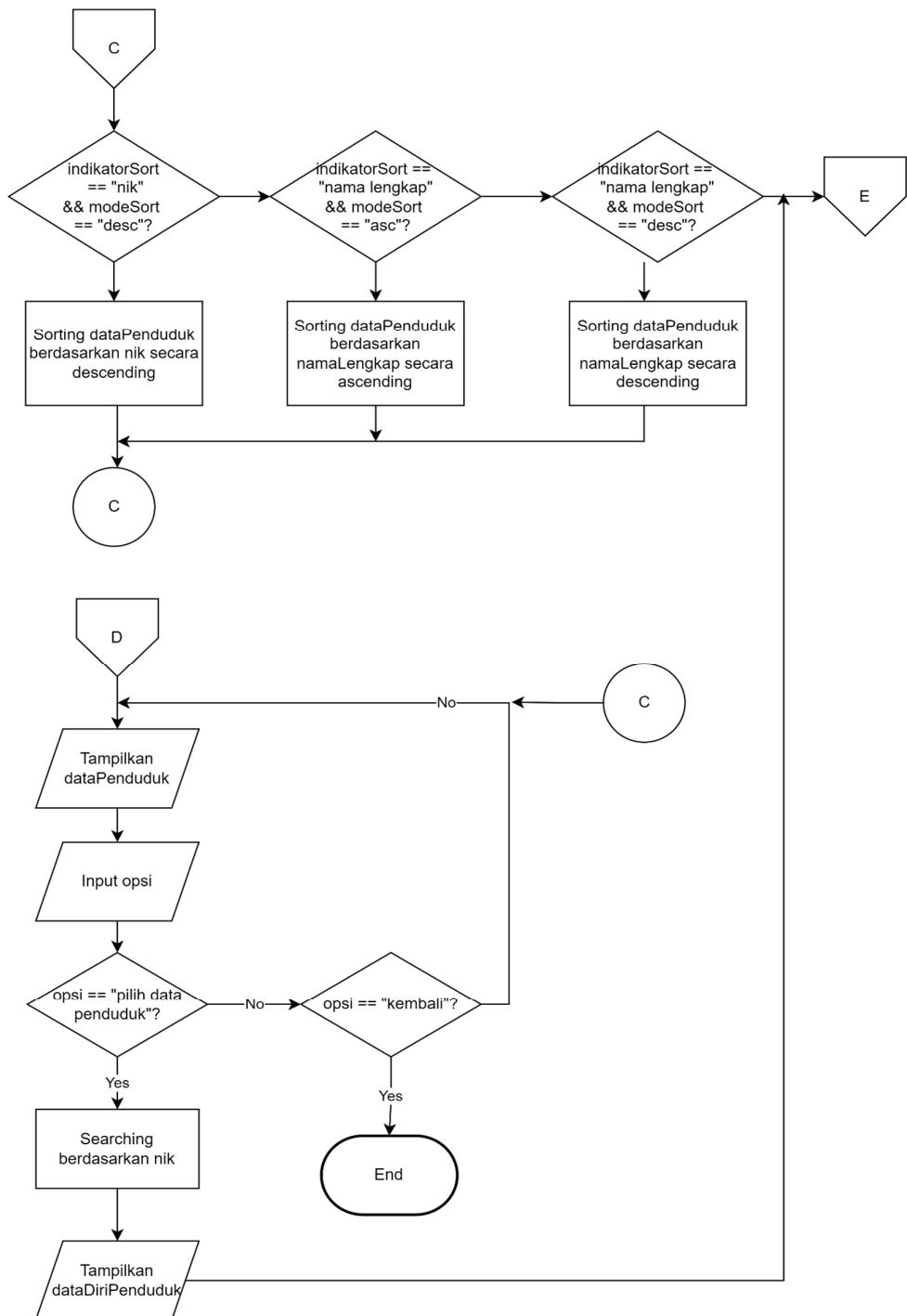
Gambar 1. 2 Flowchart Penduduk

3.) Flowchart Pemerintah (Status Hidup, Diagram Data, Hapus)



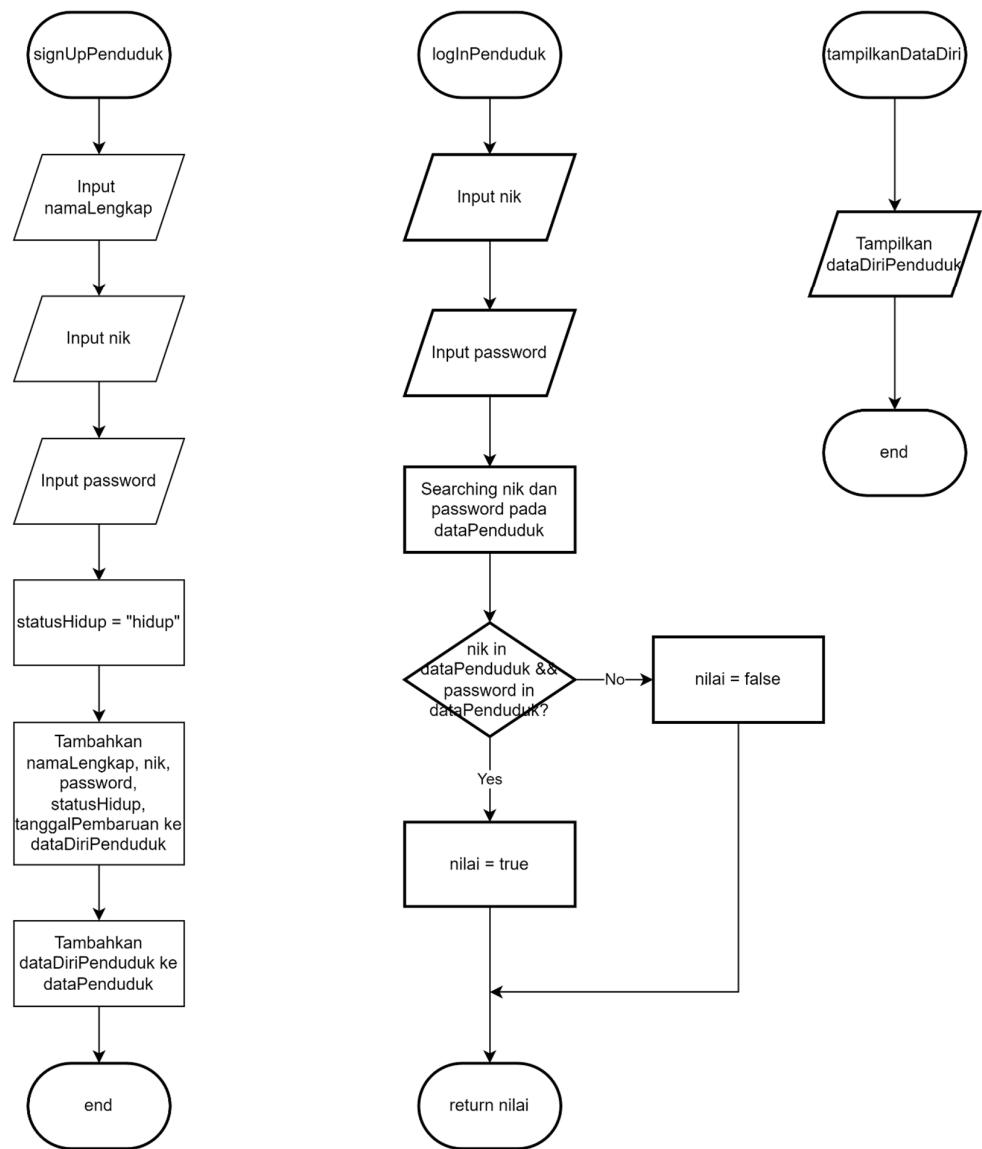
Gambar 1. 3 Flowchart Pemerintah (Status Hidup, Diagram Data, Hapus)

4.) Flowchart Sorting dan Search



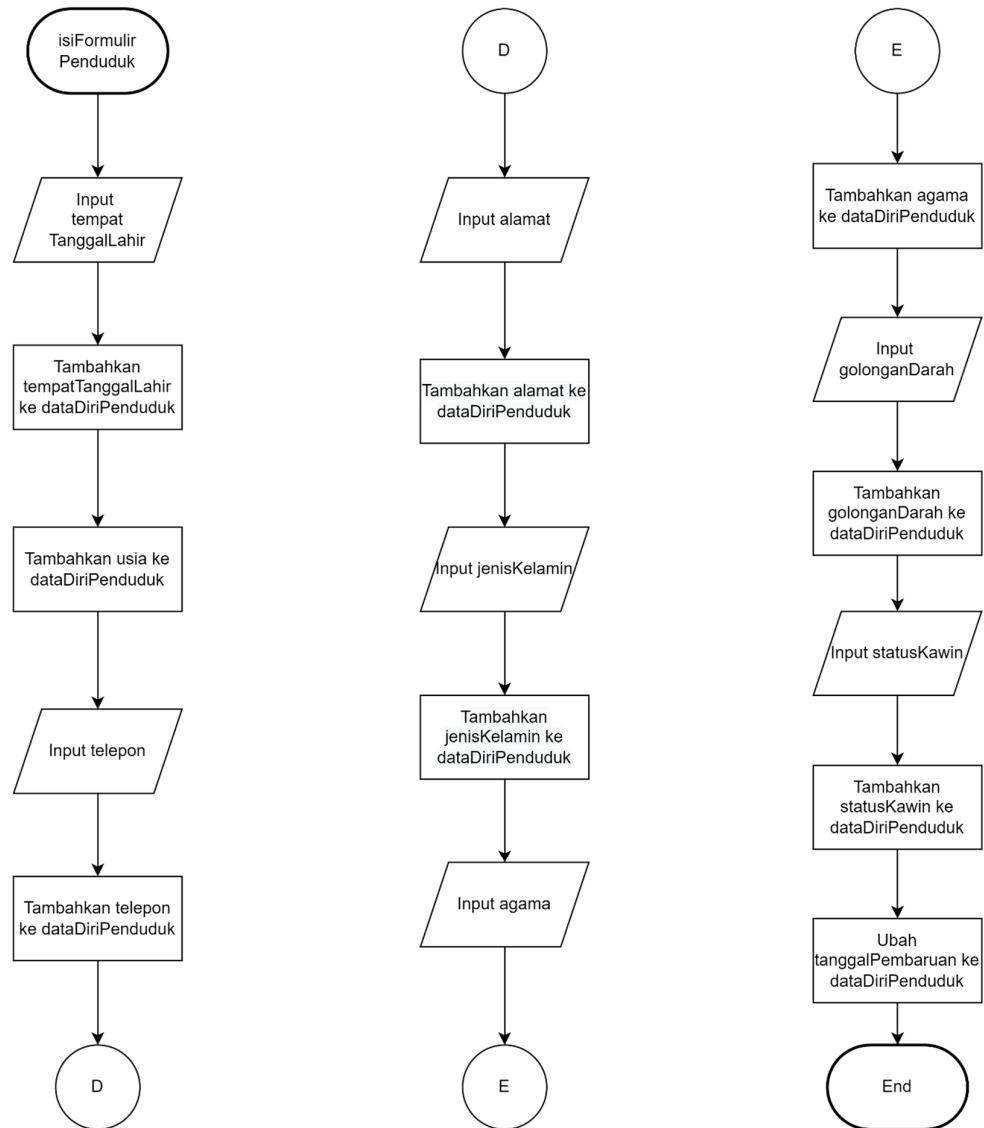
Gambar 1. 4 Flowchart Sorting & Search

5.) Flowchart Penduduk Lanjutan



Gambar 1. 5 Flowchart Penduduk Lanjutan

6.) Flowchart Pengisian Data Penduduk



Gambar 1. 6 Flowchart Pengisian Data Penduduk

7.) Flowchart Ubah Data Penduduk dan Waktu Ubah



Gambar 1. 7 Flowchart Ubah Data Penduduk

2.3 Konsep / Materi Praktikum yang Dipakai

Untuk materi yang kami gunakan dalam penyusunan program ini berpatokan kepada modul yang telah diberikan. Dengan mengacu kepada modul 1-8 sebagai modul wajib yang harus ditampilkan pada program. Berikut materi - materi yang kami gunakan diantaranya adalah :

1.) Array dan Struct

Array sebagai struktur data yang digunakan untuk menyimpan sekumpulan data dalam satu tempat. Setiap data dalam array memiliki indeks, sehingga kita akan mudah memprosesnya. Struct di program ini di gunakan untuk pengelompokan data ke dalam sebuah “tabel” yang memiliki elemen-elemen (atau field-field seperti dalam database) yang saling terkait.

2.) Fungsi dan Prosedur

Program kami juga menggunakan fungsi dan prosedur didalamnya. Dengan menggunakan fungsi akan sangat memudahkan kami dalam proses penyusunan dan pembuatan program. Karena fungsi sangat berperan penting dalam mengatur kerapian dan kemudahan sebuah kode program yang ada.

3.) Percabangan

Kami menggunakan percabangan *if, else if, else* serta *switch case* dalam penyusunan program ini. Percabangan sendiri memiliki fungsi untuk melanjutkan program apabila kondisi bernilai benar dan akan menghentikan program apabila kondisi bernilai salah.

4.) Perulangan

Perulangan *while, for* juga kami gunakan dalam program ini. Perulangan berfungsi untuk melakukan perulangan terhadap kondisi apabila kondisi tersebut dinilai belum memenuhi persyaratan yang ada.

Adapun berbagai fungsi tambahan yang kami pakai antar lain sebagai berikut:

1.) Fungsi system("cls")

untuk membersihkan layar pada program yang akan dijalankan sehingga user di dalam program dapat menentukan saat untuk menghapus data yang telah dijalankan tanpa harus menutup program tersebut dan membukanya kembali.

2.) Fungsi setiosflags(ios::fixed)

Merupakan suatu fungsi manipulator yang digunakan untuk mengatur sejumlah format keluaran data.

3.) Fungsi setprecision()

suatu fungsi manipulator yang digunakan untuk mengatur jumlah digit desimal yang ingin ditampilkan.

4.) Fungsi ifstream

meng-output-kan file yang sudah ada kedalam cmd. File yang dioutput ini boleh dalam bentuk .csv atau .txt asalkan filenya memang sudah ada. Dengan menggunakan ifstream, data yang ada didalam file csv atau txt akan ditampilkan di cmd.

5.) Fungsi ofstream

fungsi yang digunakan untuk menginput data pada file, biasanya file yang digunakan berformat csv (ms.excel) atau berformat txt (notepad).

6.) Fungsi std::string.compare()

anggota publik dari kelas string. Ini membandingkan nilai objek string (atau substring) dengan urutan karakter yang ditentukan oleh argumennya.

7.) Fungsi time_t

untuk kembali dari zaman Epoch (1970/01/01 00:00:00 UTC) waktu berlalu dalam hitungan detik. Jika detik tidak kosong, nilai kembali juga disimpan dalam variabel dalam hitungan detik.

8.) Fungsi time()

digunakan untuk mengambil waktu yang sekarang dari komputer kita. Hasil pemanggilan fungsi ini berupa angka-angka yang belum diterjemahkan,

kemudian akan disimpan dalam sebuah variabel yang telah didefinisikan sebelumnya, di program ini hasilnya akan disimpan di variabel "waktu".

9.) Fungsi cin.clear() :

digunakan yang digunakan untuk mengabaikan atau menghapus satu atau lebih karakter dari buffer input.

10.) Fungsi cin.ignore() :

digunakan yang digunakan untuk mengabaikan atau menghapus satu atau lebih karakter dari buffer input.

11.) Fungsi fflush(stdin):

menghapus buffer di dalam alat I/O. Adalah menampilkan isi data didalam suatu variabel atau nilai tanpa penampung variabel.

12.) Fungsi HANDLE :

digunakan untuk menangani error atau keadaan tertentu yang terjadi ketika program sedang berjalan. Ketika terjadi error saat program berjalan (runtime error) program akan mengarahkan alur eksekusi program ke fungsi khusus yang dinamakan handler.

13.) Fungsi GetStdHandle :

digunakan untuk mendapatkan handle ke layar. Fungsi ini mempunyai satu parameter, yaitu nStdHandle. nStdHandle adalah parameter untuk standard device. Dalam kasus ini, karena yang yang akan dilakukan adalah output ke layar, maka standard device yang digunakan adalah standard output device, yaitu STD_OUTPUT_HANDLE.

14.) Fungsi SetConsoleTextAttribute :

Fungsi yang disediakan oleh sistem operasi windows terdapat di header windows.h, yang digunakan untuk mengubah warna latar dan teks secara spesifik.

15.) Fungsi COORD :

sebuah struct yang berisikan dua variabel bertipe SHORT, yaitu X dan Y. Sedangkan dwCursorPosition adalah sebuah parameter bertipe struct COORD yang digunakan untuk menentukan posisi cursor di layar.

16.) Fungsi endl :

fungsi manipulator yang digunakan untuk menyisipkan karakter NewLine atau mengatur pindah baris. Fungsi ini sangat berguna untuk piranti keluaran berupa file di disk. File header yang harus disertakan adalah file header iostream.h .

17.) Fungsi Sleep :

sebagai timer untuk memberikan tundaan atau delay selama lima detik. Seperti yang telah dijelaskan sebelumnya, fungsi ini termasuk kategori *blocking* sehingga aplikasi utama tidak dapat mengerjakan *task* atau tugas lain.

FILE HEADER

1.) iostream :

Sebagai library untuk menerima input dan memberi output (cin untuk input, cout utk output)

2.) string :

Dengan fungsi string, Anda bisa membuat ekspresi di Access yang memanipulasi teks dalam berbagai cara. Misalnya, Anda mungkin hanya ingin menampilkan bagian dari nomor seri pada formulir. Atau, Anda mungkin perlu bergabung (menggabungkan) beberapa string bersama-sama, seperti nama belakang dan nama depan.

3.) windows.h :

untuk memanggil perintah-perintah yang ada pada command prompt seperti echo, color, attrib, Sleep, dll.

4.) ctime :

fungsi ini digunakan untuk mengonversi nilai yang dihasilkan dari pemanggilan fungsi time() ke dalam bentuk string.

5.) conio.h :

untuk memanggil console DOS I/O (input output).

6.) iomanip:

Digunakan untuk mengatur tampilan layar untuk menggunakan manipulato

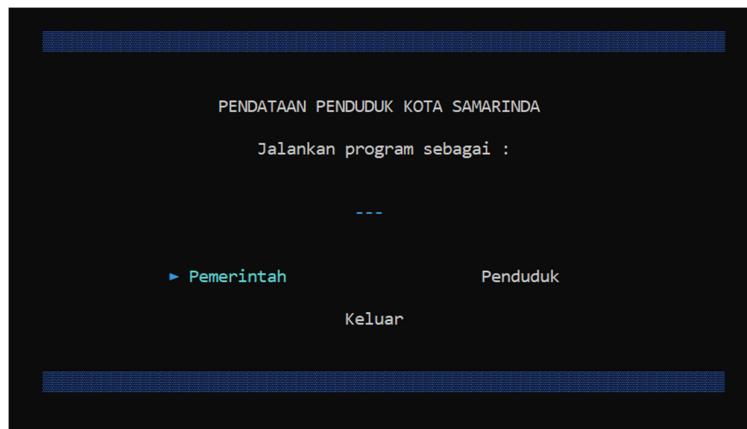
BAB 3

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

Terdapat beberapa bagian tampilan program yang ditampilkan saat program dijalankan. Tampilan dimulai dari menampilkan menu awal hingga diakhiri dengan pemberitahuan penghentian jalannya program. Berikut ini penyusun jabarkan beberapa bagian tampilan pada program yang penyusun selesaikan.

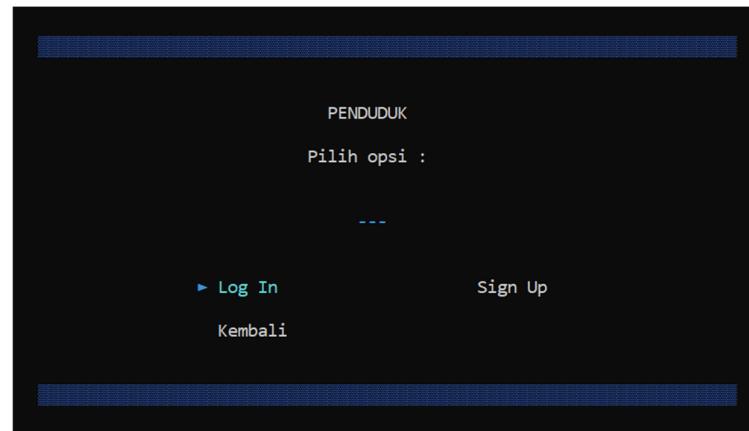
1. Tampilan Menu Awal



Gambar 2. 1 Tampilan Awal Program

Berikut adalah menu awal yang menampilkan dua mode yang bisa digunakan, yaitu sebagai pemerintah dan sebagai penduduk. Selain itu, juga ada pilihan untuk keluar yang akan menghentikan jalannya program.

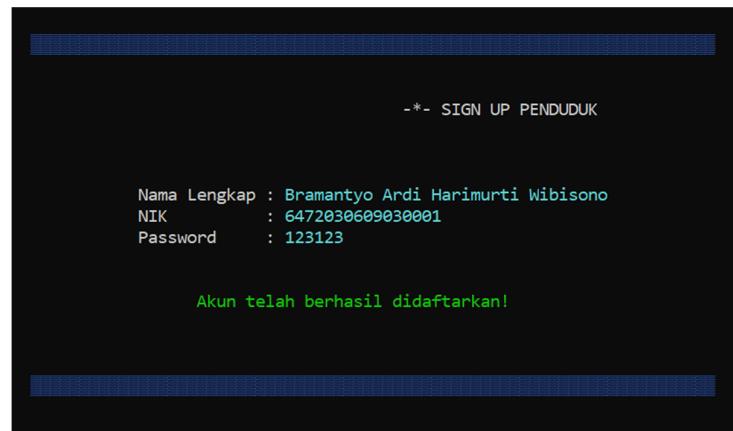
2. Tampilan Menu Masuk Akun Penduduk



Gambar 2. 1 Tampilan Menu Masuk Akun Penduduk

Berikut adalah tampilan untuk menu masuk sebagai penduduk di mana terdapat 3 menu, yaitu Log In untuk masuk menggunakan akun penduduk, Sign Up untuk mendaftarkan akun penduduk, dan Kembali untuk kembali ke menu awal.

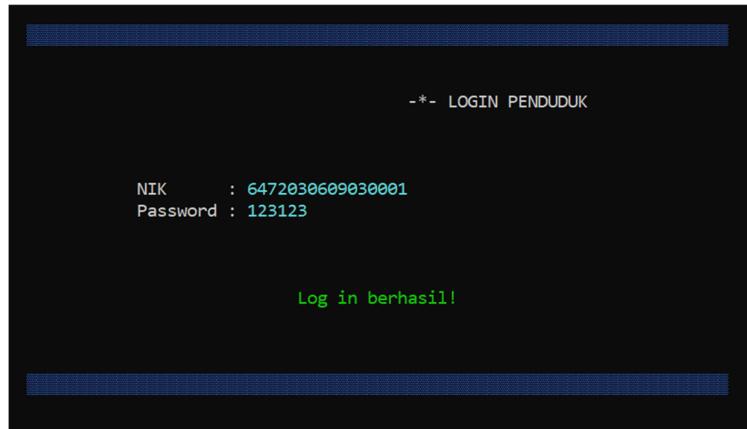
3. Tampilan Sign Up



Gambar 2. 2 Tampilan Sign Up Penduduk

Di atas adalah tampilan untuk menu Sign Up yang mengharuskan user menginput nama lengkap, NIK, dan password untuk menambahkan akun baru.

4. Tampilan Log In

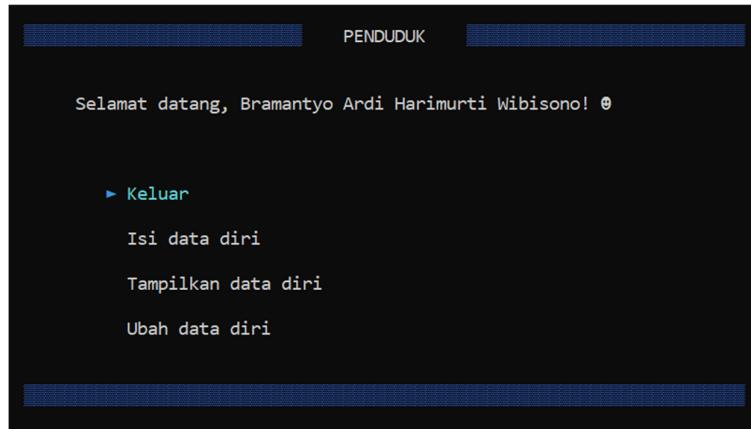


Gambar 2. 3 Tampilan Log In Penduduk

Gambar 2. 3 Tampilan Log In

Ini adalah tampilan menu Log In yang mengharuskan user memasukkan NIK yang telah terdaftar dan password yang sesuai dengan akun yang didaftarkan sebelumnya.

5. Tampilan Menu Akun Penduduk



Gambar 2. 1 Tampilan Menu Penduduk

Berikut adalah tampilan menu yang ditampilkan setelah melakukan login.

Terdapat 4 pilihan menu, yakni keluar, isi data diri, tampilkan data diri, dan ubah data diri.

6. Tampilan Menu Isi Data Diri

Formulir Biodata Penduduk

Nama Lengkap : Bramantyo Ardi Harimurti Wibisono
NIK : 6472030609030001

Tempat/Tanggal Lahir
Tempat : Samarinda

-- isi di bawah ini dengan angka

Tanggal : 6
Bulan : 9
Tahun : 2003

No. Telepon : 0816244735

<= Keluar Lanjutkan =>

Gambar 2. 5 Tampilan Menu Isi Data Diri Penduduk

Berikut adalah halaman untuk mengisi formulir biodata penduduk. Pada menu ini terbagi menjadi 3 halaman yang masing-masingnya untuk mengisi data diri untuk data yang berbeda. Pada halaman pertama ini, user akan

Formulir Biodata Penduduk

Alamat
Jalan : PERUM RATINDO GRIYA PERMAI NO 35

-- isi dengan memilih angka pada opsi

Kecamatan
[1] Palaran [5] Samarinda Utara [9] Samarinda Kota
[2] Samarinda Seberang [6] Sungai Kunjang [10] Loa Janan Ilir
[3] Samarinda Ulu [7] Sambutan
[4] Samarinda Ilir [8] Sungai Pinang
: 3

Kelurahan
[1] Teluk Lerong Ilir [4] Sidodadi [7] Gunung Kelua
[2] Jawa [5] Air Hitam [8] Bukit Pinang
[3] Air Putih [6] Dadi Mulya
: 5

Gambar 2. 2 Tampilan Isi Data Jalan, Kecamatan, Kelurahan

diminta untuk memasukkan tempat, tanggal, bulan, dan tahun lahir, serta nomor telepon.

Pada halaman kedua, user diminta untuk menginput alamat yang meliputi jalan, kecamatan, dan kelurahan sesuai dengan opsi yang disediakan.

```
Formulir Biodata Penduduk

Jenis Kelamin
[1] Laki-laki          [2] Perempuan
: 1

Agama
[1] Islam            [3] Katolik           [5] Buddha
[2] Kristen          [4] Hindu             [6] Konghuchu
: 1

Golongan Darah
[1] A                [2] B                 [3] AB               [4] O
: 4

Status Perkawinan
[1] Belum kawin      [2] Kawin            [3] Cerai hidup    [4] Cerai mati
: 1

Formulir telah berhasil diisi!

<= Keluar
```

Gambar 2. 3 Tampilan Hasil Data Diri Penduduk

Pada halaman terakhir, user akan diminta memasukkan jenis kelamin, agama, golongan darah, dan status perkawinan yang sesuai dengan opsi yang telah disediakan.

7. Tampilan Menu Tampilkan Data Diri

```
Biodata Penduduk

Nama : Bramantyo Ardi Harimurti Wibisono
NIK  : 6472030609030001
Tempat/Tanggal Lahir : Samarinda, 6-9-2003
Usia : 19

Alamat
Jalan : PERUM RATINDO GRIYA PERMAI NO 35
Kelurahan : Air Hitam
Kecamatan : Samarinda Ulu

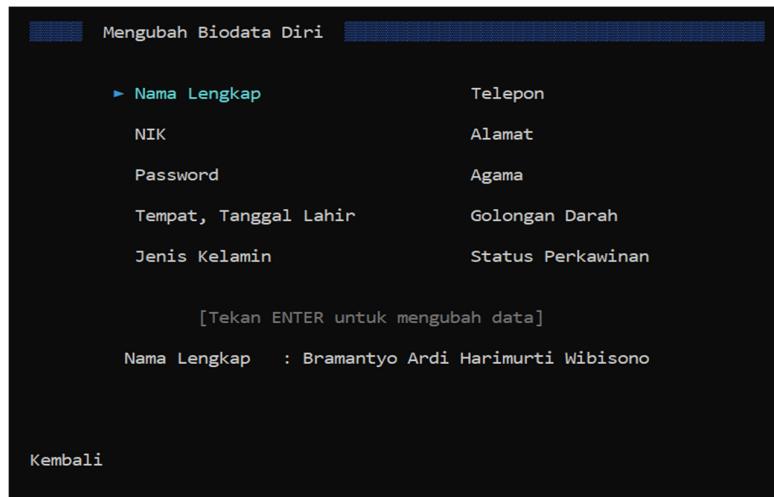
No. HP : 0816244735
Jenis Kelamin : Laki-Laki
Agama : Islam
Golongan Darah : O
Status Perkawinan : Belum kawin
Status Hidup : Hidup

<= Kembali
```

Gambar 2. 4 Tampilan Menu Data Diri Penduduk

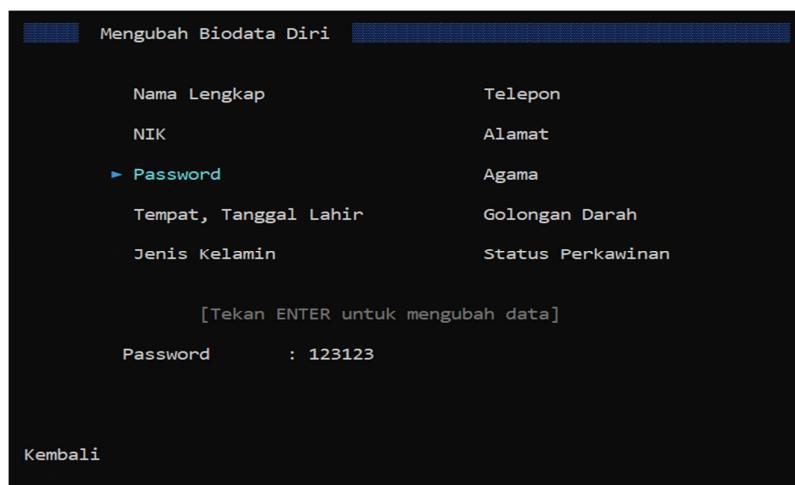
Ini adalah tampilan untuk menu tampilkan data diri yang akan menampilkan data diri penduduk yang telah mengisi formulir.

8. Tampilan Menu Ubah Data Diri



Gambar 2. 5 Tampilan Menu Ubah Data Diri Penduduk

Berikut adalah tampilan untuk menu untuk mengubah data diri penduduk. Penduduk bisa memilih ingin mengubah data yang ingin diubah dan tekan ENTER apabila ingin mengubahnya seperti gambar di bawah ini.



Gambar 2. 6 Tampilan Menu Ubah Password

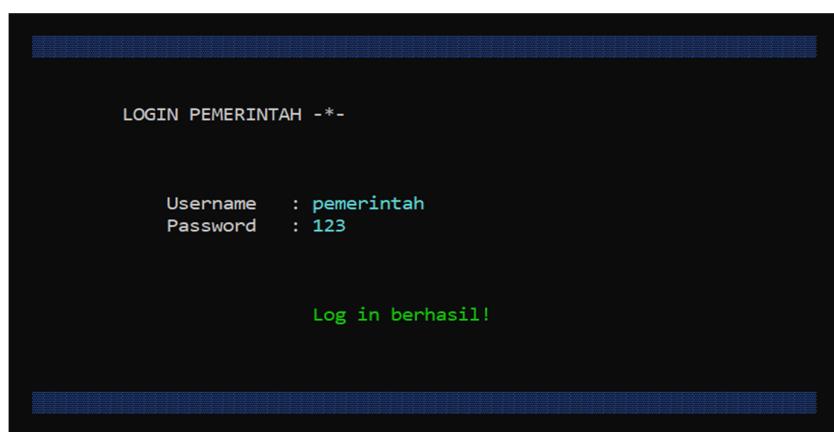


Gambar 2. 7 Tampilan Konfirmasi Penduduk



Gambar 2. 8 Tampilan Hasil Ubah Password Penduduk

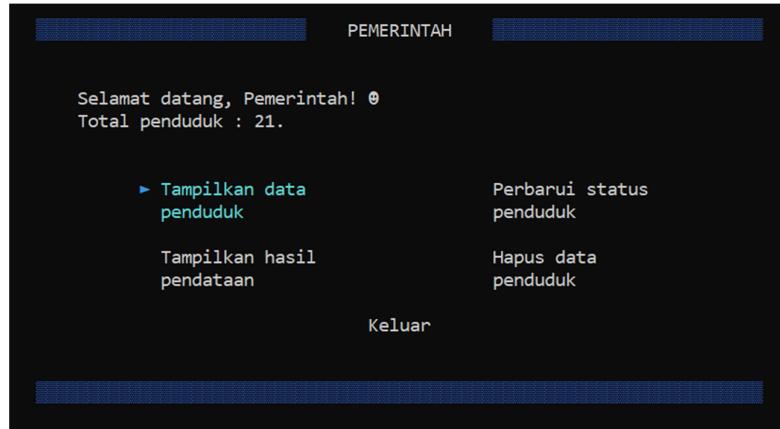
9. Tampilan Menu Log In Akun Pemerintah



Gambar 3. 1 Tampilan Menu Login Akun Pemerintah

Ini adalah tampilan apabila pada menu awal memilih untuk masuk sebagai pemerintah. Pada halaman ini, user akan diminta username dan password akun pemerintah untuk login.

10. Tampilan Menu Pemerintah



Gambar 3. 2 Tampilan Menu Pemerintah

Ini adalah tampilan menu pada akun pemerintah. Terdapat 4 pilihan menu, yaitu tampilkan data penduduk, tampilkan hasil pendataan, perbarui status penduduk, dan hapus data penduduk.

11. Tampilan Menu Tampilkan Data Penduduk

Data Penduduk			
ASCENDING [Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap ▼	NIK	Diperbarui Pada
1	Abimanyu	6472054281040006	2/6/2022
2	Agung Pratama	3171062101940005	3/6/2022
3	Azizah Oki Shofrina	6472026311020004	2/6/2022
4	Bambang Susanto	6472036901950002	2/6/2022
5	Bramantyo Ardi Harimu	6472030609030001	3/6/2022
6	Budiman Perwira	6472054910020005	2/6/2022
7	Chintia Liu Wintin	6472054911030006	2/6/2022
8	Enji	6472024102180001	3/6/2022
9	Galih Sandara	6571012002020002	3/6/2022
10	Ismail	6472094711060001	2/6/2022

Masukkan NIK :

<= Kembali Berikutnya =>

Gambar 3. 3 Tampilan Menu Data Penduduk

Berikut adalah tampilan untuk menu tampilkan data penduduk. Pada menu ini bisa dilakukan sorting berdasarkan nama lengkap, NIK, dan tanggal data diperbarui. Sorting juga bisa dilakukan secara ascending maupun descending dengan menekan SPACE pada keyboard. Selain itu, di sini user bisa melihat detail data satu penduduk dengan memasukkan NIK penduduk yang ingin dilihat. Contoh tampilannya seperti gambar di bawah ini.

- Sorting berdasarkan nama lengkap.

ASCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap ▼	NIK	Diperbarui Pada
1	Abimanyu	6472054281040006	2/6/2022
2	Agung Pratama	3171062101940005	3/6/2022
3	Aziizah Oki Shofrina	6472026311020004	2/6/2022
4	Bambang Susanto	6472036901950002	2/6/2022
5	Bramantyo Ardi Harimu	6472030609030001	3/6/2022
6	Budiman Perwira	6472054910020005	2/6/2022
7	Chintia Liu Wintin	6472054911030006	2/6/2022
8	Enji	6472024102180001	3/6/2022
9	Galih Sandara	6571012002020002	3/6/2022
10	Ismail	6472094711060001	2/6/2022

Gambar 3. 4 Tampilan Sorting Ascending

DESCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap ▼	NIK	Diperbarui Pada
1	steven	6474563489761234	2/5/2022
2	Yulianti Haji	8204040711170002	3/6/2022
3	Yu Hao	6472026010200002	3/6/2022
4	Yanti Susanta	65010115303600001	3/6/2022
5	Wulandari	7604014910700002	3/6/2022
6	Sri Handayani	9207024904800001	3/6/2022
7	Shendy Sanjaya	6472051812920005	2/6/2022
8	Muhammad fikri	6474022404030003	2/5/2022
9	Muhammad Ahsan	6205041111870001	3/6/2022
10	Miko	6472345643215436	3/6/2022

Gambar 3. 5 Tampilan Sorting Descending

b. Sorting berdasarkan NIK.

ASCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap	NIK ▼	Diperbarui Pada
1	Agung Pratama	3171062101940005	3/6/2022
2	Muhammad Ahsan	6205041111870001	3/6/2022
3	Enji	6472024102180001	3/6/2022
4	Yu Hao	6472026010200002	3/6/2022
5	Aziizah Oki Shofrina	6472026311020004	2/6/2022
6	Bramantyo Ardi Harimu	6472030609030001	3/6/2022
7	Bambang Susanto	6472036901950002	2/6/2022
8	Shendy Sanjaya	6472051812920005	2/6/2022
9	Abimanyu	6472054281040006	2/6/2022
10	Budiman Perwira	6472054910020005	2/6/2022

Gambar 3. 6 Tampilan Sorting NIK Ascending

DESCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap	NIK ▼	Diperbarui Pada
1	Sri Handayani	9207024904800001	3/6/2022
2	Yulianti Haji	8204040711170002	3/6/2022
3	Wulandari	7604014910700002	3/6/2022
4	Galih Sandara	6571012002020002	3/6/2022
5	Yanti Susanta	6501015303600001	3/6/2022
6	steven	6474563489761234	2/5/2022
7	Muhammad fikri	6474022404030003	2/5/2022
8	Miko	6472345643215436	3/6/2022
9	Ismail	6472094711060001	2/6/2022
10	Jiang Chen	6472060208010003	2/6/2022

Gambar 3. 7 Tampilan Sorting NIK Descending

- c. Sorting berdasarkan tanggal diperbarui.

ASCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap	NIK	Diperbarui Pada ▼
1	steven	6474563489761234	2/5/2022
2	Muhammad Fikri	6474022404030003	2/5/2022
3	Budiman Perwira	6472054910020005	2/6/2022
4	Shendy Sanjaya	6472051812920005	2/6/2022
5	Aziizah Oki Shofrina	6472026311020004	2/6/2022
6	Chintia Liu Wintin	6472054911030006	2/6/2022
7	Ismail	6472094711060001	2/6/2022
8	Abimanyu	6472054281040006	2/6/2022
9	Jiang Chen	6472060208010003	2/6/2022
10	Bambang Susanto	6472036901950002	2/6/2022

Gambar 3. 8 Tampilan Sorting Tanggal Diperbarui Ascending

DESCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap	NIK	Diperbarui Pada ▼
1	Sri Handayani	9207024904800001	3/6/2022
2	Yulianti Haji	8204040711170002	3/6/2022
3	Wulandari	7604014910700002	3/6/2022
4	Galih Sandara	6571012002020002	3/6/2022
5	Yanti Susanta	6501015303600001	3/6/2022
6	Bramantyo Ardi Harimu	6472030609030001	3/6/2022
7	Enji	6472024102180001	3/6/2022
8	Miko	6472345643215436	3/6/2022
9	Agung Pratama	3171062101940005	3/6/2022
10	Muhammad Ahsan	6205041111870001	3/6/2022

Gambar 3. 9 Tampilan Sorting Tanggal Diperbarui Descending

d. Searching NIK.

Data Penduduk			
ASCENDING			
[Tekan SPACE untuk mengubah mode sorting]			
No	Nama Lengkap ▼	NIK	Diperbarui Pada
1	Abimanyu	6472054281040006	2/6/2022
2	Agung Pratama	3171062101940005	3/6/2022
3	Aziizah Oki Shofrina	6472026311020004	2/6/2022
4	Bambang Susanto	6472036901950002	2/6/2022
5	Bramantyo Ardi Harimu	6472030609030001	3/6/2022
6	Budiman Perwira	6472054910020005	2/6/2022
7	Chintia Liu Wintin	6472054911030006	2/6/2022
8	Enji	6472024102180001	3/6/2022
9	Galih Sandara	6571012002020002	3/6/2022
10	Ismail	6472094711060001	2/6/2022

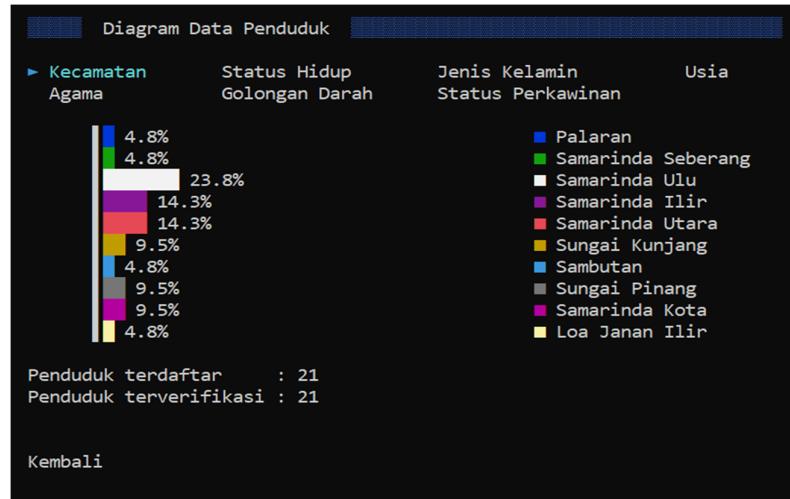
► Masukkan NIK : 6472054911030006

Gambar 3. 10 Tampilan Searching NIK

Biodata Penduduk	
Nama	: Chintia Liu Wintin
NIK	: 6472054911030006
Tempat/Tanggal Lahir	: Samarinda, 9-11-2003
Usia	: 19
Alamat	
Jalan	: D.I.Panjaitan
Kelurahan	: Bandara
Kecamatan	: Sungai Pinang
No. HP	: 081545962990
Jenis Kelamin	: Perempuan
Agama	: Islam
Golongan Darah	: A
Status Perkawinan	: Belum kawin
Status Hidup	: Hidup
<= Kembali	

Gambar 3. 11 Tampilan Hasil Searching NIK

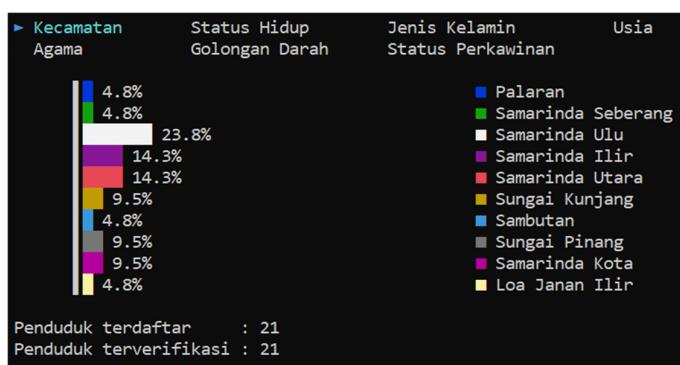
12. Tampilan Menu Hasil Pendataan Penduduk



Gambar 4. 1 Diagram Hasil Pendataan Penduduk

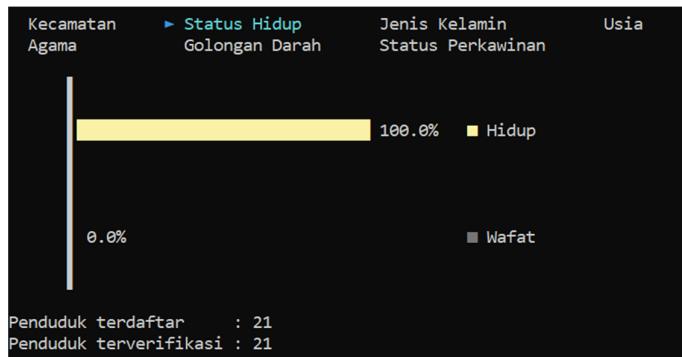
Berikut ini adalah tampilan menu tampilkan hasil pendataan berupa diagram batang dari seluruh data penduduk yang telah mengisi formulir. Terdapat 7 jenis diagram, yaitu berdasarkan kecamatan, status hidup, jenis kelamin, usia, agama, golongan darah, dan status perkawinan.

a. Berdasarkan kecamatan.



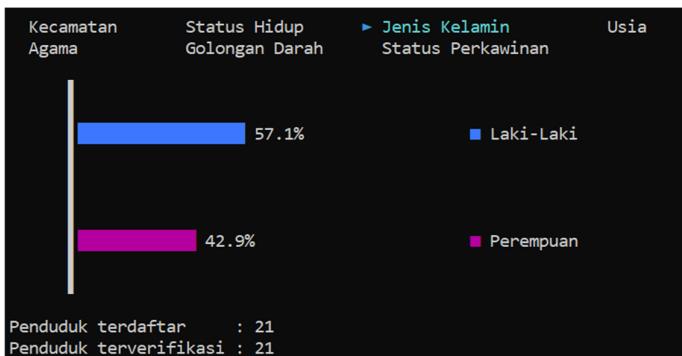
Gambar 4. 2 Diagram Kecamatan

b. Berdasarkan status hidup.



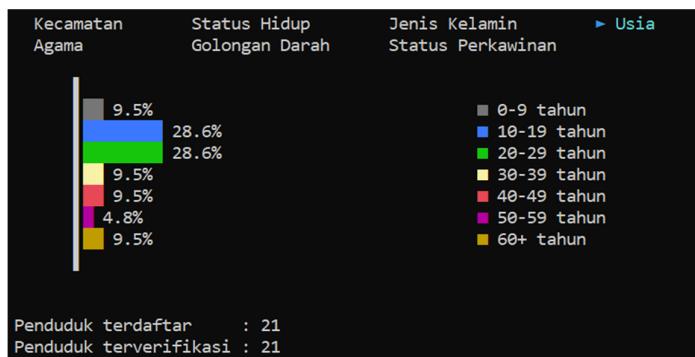
Gambar 4. 3 Diagram Status Hidup

c. Berdasarkan jenis kelamin.



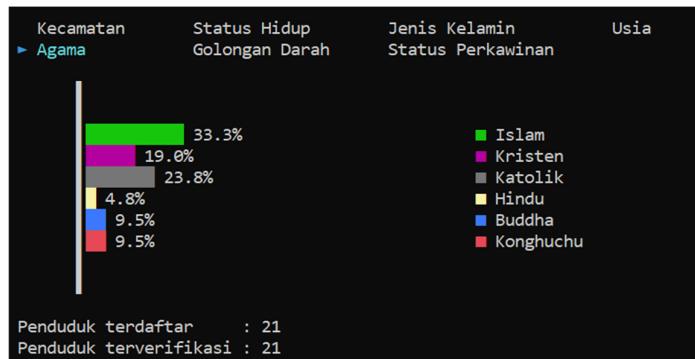
Gambar 4. 4 Diagram Jenis Kelamin

d. Berdasarkan usia.



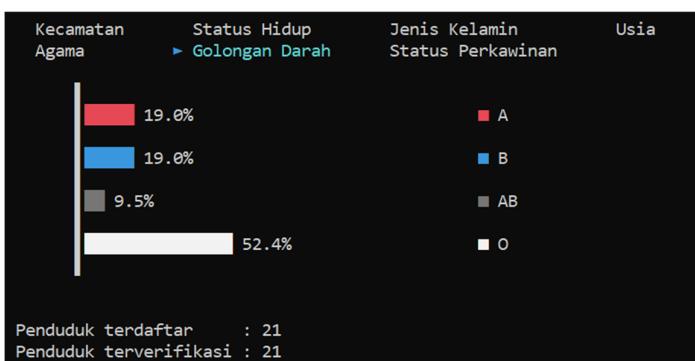
Gambar 4. 5 Diagram Usia

e. Berdasarkan agama.



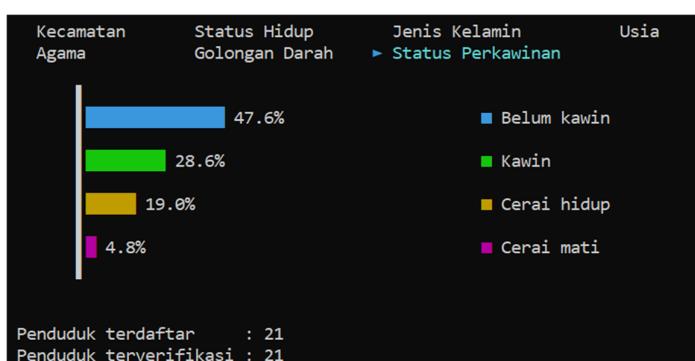
Gambar 4. 6 Diagram Agama

f. Berdasarkan golongan darah.



Gambar 4. 7 Diagram Golongan Darah

g. Berdasarkan status perkawinan.



Gambar 4. 8 Diagram Status Perkawinan

13. Tampilan Menu Perbarui Status Hidup

```
Ubah Status Hidup Penduduk

Masukkan NIK yang ingin diubah : 7604014910700002
=====
Hasil pencarian 7604014910700002 ditemukan.

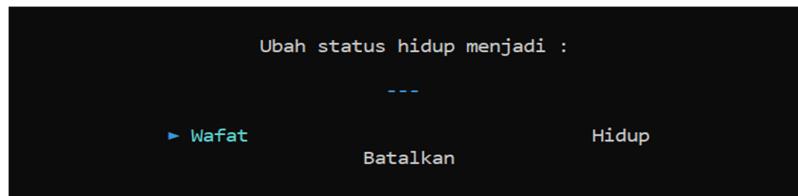
Apakah Anda yakin ingin mengubah status hidup
data penduduk berikut ini?

Nama : Wulandari
NIK  : 7604014910700002
Status Hidup : Hidup

=====
Tekan [Enter] untuk Melanjutkan . . .
```

Gambar 5. 1 Tampilan Menu Ubah Status Hidup

Berikut adalah tampilan untuk menu perbarui status hidup penduduk. Pada menu ini, user akan diminta memasukkan NIK yang ingin diubah kemudian akan ditampilkan data penduduk sesuai NIK tersebut. Apabila ingin melanjutkan perubahan, tekan ENTER.



Gambar 5. 2 Tampilan Opsi Ubah Status Hidup

Akan muncul opsi seperti gambar di atas di mana user bisa memilih status hidup yang dikehendaki, yaitu wafat atau hidup. Jika ingin membatalkan, pilih batalkan.

```

Ubah Status Hidup Penduduk

Masukkan NIK yang ingin diubah : 7604014910700002
=====
Hasil pencarian 7604014910700002 ditemukan.

Apakah Anda yakin ingin mengubah status hidup
data penduduk berikut ini?

Nama : Wulandari
NIK : 7604014910700002
Status Hidup : Hidup

=====
Ubah status hidup menjadi :

---
<= Kembali           Status hidup pada data penduduk ini
                           telah berhasil diubah.

```

Gambar 5. 3 Tampilan Hasil Ubah Status Hidup

Jika berhasil mengubah data, maka tampilan akan seperti gambar di atas.

14. Tampilan Menu Hapus Data Penduduk

```

Hapus Data Penduduk

Masukkan NIK yang ingin dihapus : 6472054911030006
=====
Hasil pencarian 6472054911030006.

Apakah Anda yakin ingin menghapus data
penduduk berikut ini?

Nama : Chintia Liu Wintin
NIK : 6472054911030006
Status Hidup : Hidup

Ketik 6472054911030006 untuk menghapus.
: 6472054911030006

=====
Sukses... Data Berhasil Terhapus.
=====
```

Gambar 6. 1 Tampilan Hapus Data Penduduk

Berikut adalah tampilan menu hapus data penduduk. Pertama, user akan diminta menginput NIK penduduk yang datanya ingin dihapus, kemudian ditampilkan pada program. Jika ingin benar-benar menghapus, maka user harus memasukkan ulang NIK penduduk tersebut kembali dengan benar dan data akan terhapus.

15. Tampilan Program Dihentikan\



Gambar 7. 1 Tampilan Program Dihentikan

Berikut adalah tampilan program apabila program berhasil dihentikan melalui pilihan keluar di menu awal.

3.2 Source Code

Di bawah ini adalah source code mengenai program yang penyusun telah kerjakan.

*catatan : *code* yang akan kami tampilkan mungkin menjadi sedikit kurang rapi dikarenakan *code* ini merupakan hasil *copy paste* dari Visual Studio Code

```
// include library
#include <iostream>
#include <string>    // string, to_string(), stoi
#include <windows.h>
#include <ctime>
#include <conio.h>
#include <iomanip>
#include <fstream>

using namespace std;
```

```

// key pada keyboard
#define UP 72
#define DOWN 80
#define LEFT 75
#define RIGHT 77

// menu yang dipilih
#define SELECT 11
#define UNSELECT 7
#define ENTER 13
#define RESET 7
#define SIGNED 3
#define UNSIGNED 0
#define FAIL 12
#define SUCCESS 10
#define MARKED 8

// penyimpanan data
#define STORAGE 1000

// pilihan utk isi formulir
const string GENDER[] = {"Laki-Laki", "Perempuan"};
const string AGAMA[] = {"Islam", "Kristen", "Katolik", "Hindu", "Buddha",
"Konghuchu"};
const string GOLDAR[] = {"A", "B", "AB", "O"};
const string STATUSKAWIN[] = {"Belum kawin", "Kawin", "Cerai hidup", "Cerai
mati"};
const string STATUSHIDUP[] = {"Wafat", "Hidup"};
const string KECAMATAN[] = {
    "Palaran",     "Samarinda Seberang", "Samarinda Ulu",
    "Samarinda Ilir", "Samarinda Utara",   "Sungai Kunjang",  "Sambutan",
    "Sungai Pinang", "Samarinda Kota",    "Loa Janan Ilir"
};
const string KELURAHAN[10][8] = {
    {"Rawa Makmur", "Handil Bakti", "Bukuan", "Simpang Pasir", "Bantuas"},
    {"Sungai Keledang", "Baq'a", "Mesjid", "Mangkupalas", "Tenun", "Gunung Panjang"}
};

```

```

    {"Teluk Lerong Ilir", "Jawa", "Air Putih", "Sidodadi", "Air Hitam", "Dadi Mulya",
"Gunung Kelua", "Bukit Pinang"},

    {"Selili", "Sungai Dama", "Sidomulyo", "Sidodamai", "Pelita"},

    {"Sempaja Selatan", "Lempake", "Sungai Siring", "Sempaja Utara", "Tanah Merah",
"Sempaja Barat", "Sempaja Timur", "Budaya Pampang"},

    {"Loa Bakung", "Loa Buah", "Karang Asam Ulu", "Lok Bahu", "Teluk Lerong Ulu",
"Karang Asam Ilir", "Karang Anyar"},

    {"Sungai Kapih", "Sambutan", "Makroman", "Sindang Sari", "Pulau Atas"},

    {"Temindung Permai", "Sungai Pinang Dalam", "Gunung Lingai", "Mugirejo",
"Bandara"},

    {"Karang Mumus", "Pelabuhan", "Pasar Pagi", "Bugis", "Sungai Pinang Luar"},

    {"Simpang Tiga", "Tani Aman", "Sengkotek", "Harapan Baru", "Rapak Dalam"}}

};

const string HARI[]      = {"Minggu", "Senin", "Selasa", "Rabu", "Kamis", "Jumat",
"Sabtu"};

```

// ## Modul 4 - Struct

```

// struct alamat
struct Alamat {
    string jalan;
    short int kecamatan = -1;
    short int kelurahan = -1;
};

// struct penduduk
struct Penduduk{
    string tanggalPembaruan;
    string namaLengkap;
    char nik[17];
    string password;
    string ttl;
    Alamat alamat;
    string telepon;
    int usia = -1;
    short int gender = -1; // index GENDER
    short int agama = -1; // index AGAMA
    short int golDar = -1; // index GOLDAR
};

```

```

    short int statusKawin = -1;      // index STATUSKAWIN
    short int statusHidup = -1;      // index STATUSHIDUP
};

/*----- PROTOTYPE -----*/
// ## Modul 3 - Fungsi dan Prosedur

// tampilan mode penduduk
void menuMasukPenduduk();           // menu login atau signup
void logInPenduduk();               // log in penduduk
void signUpPenduduk();              // sign up penduduk
void menuPenduduk(Penduduk penduduk); // menu di akun penduduk
void isiFormulirPenduduk(Penduduk &penduduk); // isi formulir penduduk
bool cancelForm();                  // notif batal mengisi formulir
void tampilkanData(Penduduk penduduk); // tampilkan data diri penduduk
void ubahDataDiri(Penduduk &penduduk); // ubah data diri penduduk

// tampilan mode pemerintah
void logInPemerintah();            // log in pemerintah
void menuPemerintah();              // menu di akun pemerintah
void tampilkanDiagram();            // tampilkan diagram data penduduk
void tampilkanDataPenduduk();        // tampilkan seluruh data penduduk
void hapusDataPenduduk();           // hapus data penduduk
void ubahStatusHidup();             

// diagram
void diagramKecamatan();           // diagram berdasarkan kecamatan
void diagramGender();                // diagram berdasarkan gender
void diagramStatusHidup();          // diagram berdasarkan status hidup
void diagramUsia();                 // diagram berdasarkan usia
void diagramAgama();                // diagram berdasarkan agama
void diagramGolDar();                // diagram berdasarkan golongan darah
void diagramStatusKawin();           // diagram berdasarkan status

// tampilan opsi
void tampilkanOpsi(string opsi);     // opsi saat mengisi formulir

```

```

// tabel
void tabel(short int nama, short int nik, short int tanggal); // tabel data penduduk

// write & read txt
void importFromTxt(); // ambil semua data dari txt
void appendToTxt(Penduduk penduduk); // tambahkan data baru ke txt
void deleteFromTxt(Penduduk penduduk); // hapus data tertentu dari txt
void updateToTxt(Penduduk penduduk); // ubah data tertentu di txt

// sorting & searching
int searchNIK(string nik); // cari index NIK - binary search
void sortNIK(string mode); // sorting berdasarkan nik - insertion sort
void swap(Penduduk *elemen1, Penduduk *elemen2); // tukar tempat elemen
int partition(int low, int high, string mode); // partisi untuk quick sort
void sortNamaLengkap(int low, int high, string mode); // sorting berdasarkan nama lengkap - quick sort
void sortTanggal(string mode); // sorting berdasarkan tanggal diperbarui - shell sort
int indexElemen(short int array[], int cari, int length); // cari index elemen - sequential search

// tambahan
void color(unsigned short warna); // ganti warna
void gotoxy(int x, int y); // ganti posisi
tm *timeNow(); // waktu sekarang
string hariIni(); // tanggal hari ini
int random(); // angka random
void clearCin(); // bersihkan buffer
bool loginBerhasil(char nik[], string password); // memastikan nik dan pw benar
int banyakData(); // banyak data tersimpan
void karakter(int ascii, int jumlah, int warna=7); // dekorasi tampilan
void diagram(int jumlah, unsigned int warna=7); // batang diagram
bool formulirTerisi(Penduduk penduduk); // cek apakah formulir sudah diisi
void hapusTulisan(int panjang=50); // hapus tulisan pada terminal
int banyakDataTerisi(); // banyak data yg sudah diisi
int ambilTanggal(string str); // ambil tanggal dari format tanggal (string)
int ambilBulan(string str); // ambil bulan dari format tanggal (string)
int ambilTahun(string str); // ambil tahun dari format tanggal (string)

```

```

// cek pengisian formulir
bool isAngka(string str); // memastikan string diisi angka saja
bool isSpace(string str); // memastikan string diisi spasi saja
bool isTanggal(int tanggal); // memastikan sesuai ketentuan tanggal
bool isBulan(int bulan); // memastikan sesuai ketentuan bulan
bool isTahun(int tahun); // memastikan sesuai ketentuan tahun
bool isNIK(char nik[]); // memastikan sesuai ketentuan nik
bool isTelp(string telp); // memastikan sesuai ketentuan telepon

// ## Modul 2 - Array dan Pointer

// data tersimpan
Penduduk dataPenduduk[STORAGE];

/*----- MAIN PROGRAM -----*/
int main(){
    // default
    bool running = true; // program berjalan
    int notif = SIGNED,
        opsi1 = SELECT, sign1, // warna opsi & tanda 1 - pemerintah
        opsi2 = UNSELECT, sign2, // warna opsi & tanda 2 - penduduk
        opsi3 = UNSELECT, sign3; // warna opsi & tanda 3 - keluar

    while (running) {

        importFromTxt(); // ambil data dari txt

        // tampilan -----
        // warna tanda saat memilih opsi
        sign1 = (opsi1 == SELECT) ? SIGNED : UNSIGNED;
        sign2 = (opsi2 == SELECT) ? SIGNED : UNSIGNED;
        sign3 = (opsi3 == SELECT) ? SIGNED : UNSIGNED;

        system("cls"); // bersihkan terminal
    }
}

```

```

// batas atas
cout << endl << endl
<< "\t"; karakter(177, 70, 9);
cout << endl << endl << endl;

// judul
cout << "\t\t\t PENDATAAN PENDUDUK KOTA SAMARINDA \n\n";
cout << "\t\t\t Jalankan program sebagai : \n\n\n";

color(notif); cout << "\t\t\t\t --- \n\n\n";

// menu
color(sign1); cout << "\t\t\t " << char(16); // opsi 1
color(opsi1); cout << " Pemerintah" ;

color(sign2); cout << "\t\t\t " << char(16); // opsi 2
color(opsi2); cout << " Penduduk\n\n";

color(sign3); cout << "\t\t\t\t " << char(16); // opsi 3
color(opsi3); cout << " Keluar \n\n";

color(RESET); cout << endl;

// batas bawah
cout << "\t"; karakter(177, 70, 9);

// navigasi -----
// pilih mode
unsigned char mode, key;
mode = getch();

// opsi yg tersedia
switch (mode) {

    // tekan ENTER
    case ENTER:

```

```

// mode pemerintah
if (opsi1 == SELECT) {
    logInPemerintah();

// mode penduduk
} else if (opsi2 == SELECT) {
    menuMasukPenduduk();

// keluar
} else if (opsi3 == SELECT) {
    running = false;

}

break;

// tekan UP || RIGHT || DOWN || LEFT
case 224:
    key = getch();

// pilih opsi 1 - pemerintah
if (key == LEFT || key == UP && opsi3 == SELECT) {
    opsi1 = SELECT;
    opsi2 = UNSELECT;
    opsi3 = UNSELECT;

// pilih opsi 2 - penduduk
} else if (key == RIGHT) {
    opsi2 = SELECT;
    opsi1 = UNSELECT; opsi3 = UNSELECT;

// pilih opsi 3 - keluar
} else if (key == DOWN) {
    opsi3 = SELECT;
    opsi1 = UNSELECT; opsi2 = UNSELECT;
}

break;

```

```

// opsi tidak ada
default:
    notif = FAIL;
    continue; // lanjutkan openPage

    break;
}

notif = SIGNED;
}

// program dihentikan
system("cls");

color(14);
cout << "\n\n\n\n\n\n\n\n\n\t\t"
<< "Program dihentikan."
<< endl << endl << endl << endl;

color(RESET);

getch();

return 0;
}

/*
----- TAMPILAN -----
----- */

// modul 3 - Fungsi dan Prosedur
// penduduk -----
=====

void menuMasukPenduduk() {

    // default
    bool openPage = true; // selama halaman dibuka
    short int notif = SIGNED,

```

```

opsi1 = SELECT, sign1, // login
opsi2 = UNSELECT, sign2, // sign up
opsi3 = UNSELECT, sign3, // kembali

while (openPage) {

    // tampilan -----
    // warna tanda saat memilih opsi
    sign1 = (opsi1 == SELECT) ? SIGNED : UNSIGNED;
    sign2 = (opsi2 == SELECT) ? SIGNED : UNSIGNED;
    sign3 = (opsi3 == SELECT) ? SIGNED : UNSIGNED;

    system("cls"); cout << endl << endl;
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 18); // batas atas
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 5); // batas bawah

    // judul
    cout << "\t\t\t PENDUDUK \n\n";
    cout << "\t\t\t Pilih opsi : \n\n\n";

    color(notif); cout << "\t\t\t\t --- \n\n\n";

    // menu
    color(sign1); cout << "\t\t\t\t << char(16); // opsi 1
    color(opsi1); cout << " Log In " ;

    color(sign2); cout << "\t\t\t\t << char(16); // opsi 2
    color(opsi2); cout << " Sign Up\n\n ";

    color(sign3); cout << "\t\t\t\t << char(16); // opsi 3
    color(opsi3); cout << " Kembali\n\n ";

    // navigasi -----
    // pilih menu
    unsigned char menu, key;
}

```

```

menu = getch();

// opsi yg tersedia
switch (menu) {

    // tekan ENTER
    case ENTER:

        // kembali
        if (opsi1 == SELECT) {
            logInPenduduk();

        // log in
        } else if (opsi2 == SELECT) {
            signUpPenduduk();

        // sign up
        } else if (opsi3 == SELECT) {
            openPage = false;
        }

        break;

    // tekan UP || RIGHT || DOWN || LEFT
    case 224:
        key = getch();

        // pilih opsi 1 - log in
        if (
            key == UP && opsi3 == SELECT ||
            key == LEFT && opsi2 == SELECT
        ) {
            opsi1 = SELECT;
            opsi2 = UNSELECT; opsi3 = UNSELECT;
        }

        // pilih opsi 2 - sign up
        else if (

```

```

key == RIGHT && opsi1 == SELECT ||
key == RIGHT && opsi3 == SELECT
) {
    opsi2 = SELECT;
    opsi1 = UNSELECT; opsi3 = UNSELECT;

    // pilih opsi 3 - kembali
} else if (
    key == DOWN && opsi1 == SELECT ||
    key == DOWN && opsi2 == SELECT
) {
    opsi3 = SELECT;
    opsi1 = UNSELECT; opsi2 = UNSELECT;
}

break;

default:
    notif = FAIL;
    continue; // lanjutkan openPage

break;
}

notif = SIGNED;

}

}

void logInPenduduk() {
    system("cls");
    cout << endl << endl;

    // dekorasi
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 18); // batas atas
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 5); // batas bawah

    // deklarasi var
}

```

```

char nik[17];
string password;

//judul
cout << "\t\t\t -*- LOGIN PENDUDUK\n\n\n" << endl;

// masukkan nik
cout << "\t\t NIK : "; color(SELECT);
cin.get(nik, 17); clearCin(); color(RESET);

// masukkan password
cout << "\t\t Password : "; color(SELECT);
getline(cin, password); fflush(stdin); color(RESET);

cout << endl << endl << endl;

// login berhasil
if (logInBerhasil(nik, password)) {

    color(SUCCESS);
    cout << "\t\t\t Log in berhasil!"
        << endl << endl;
    color(RESET); getch();

    int index = searchNIK(nik);      // index nik
    menuPenduduk(dataPenduduk[index]); // masuk ke akun sesuai nik

    // login gagal
} else {

    color(FAIL);
    cout << "\t\t\t Log in gagal! "
        << endl << endl;
    color(RESET); getch();
}

}

```

```

void signUpPenduduk() {
    system("cls");
    cout << endl << endl;

    // dekorasi
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 18); // batas atas
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 5); // batas bawah

    // deklarasi var
    Penduduk pendudukBaru;

    // judul
    cout << "\t\t\t -*- SIGN UP PENDUDUK \n\n\n" << endl;

    // masukkan nama
    cout << "\t\t Nama Lengkap : "; color(SELECT);
    getline(cin, pendudukBaru.namaLengkap); color(RESET); fflush(stdin);

    // masukkan nik
    cout << "\t\t NIK : "; color(SELECT);
    cin.get(pendudukBaru.nik, 17); color(RESET); clearCin();

    // masukkan password
    cout << "\t\t Password : "; color(SELECT);
    getline(cin, pendudukBaru.password); color(RESET); fflush(stdin);

    cout << endl << endl;

    // gagal sign up - nik sudah ada
    if (searchNIK(pendudukBaru.nik) != -1) {
        color(FAIL); cout << "\t\t\t NIK telah terdaftar! \n\n" << endl;

        // gagal sign up - nik salah
    } else if (!isNIK(pendudukBaru.nik)) {
        color(FAIL); cout << "\t\t\t NIK yang dimasukkan tidak valid! \n\n" << endl;

        // gagal sign up - nama dikosongkan
    } else if (isSpace(pendudukBaru.namaLengkap)) {

```

```

color(FAIL); cout << "\t\t\t Nama tidak boleh kosong! \n\n" << endl;

// gagal sign up - password dikosongkan
} else if (isSpace(pendudukBaru.password)) {
    color(FAIL); cout << "\t\t\t Password tidak boleh kosong! \n\n" << endl;

// berhasil sign up
} else {

    pendudukBaru.tanggalPembaruan = hariIni(); // ubah tanggal diperbarui
    pendudukBaru.statusHidup = 1;           // status hidup = hidup

    appendToTxt(pendudukBaru);           // tambahkan ke file eksternal

    color(SUCCESS); cout << "\t\t\t Akun telah berhasil didaftarkan! \n\n" << endl;
}

color(RESET); cout << endl;
getch();

}

void menuPenduduk(Penduduk penduduk) {

// var yg dipakai
bool openPage = true;          // selama buka halaman
short int opsi1 = SELECT, sign1, // keluar
        opsi2 = UNSELECT, sign2, // isi data diri
        opsi3 = UNSELECT, sign3, // tampilkan data diri
        opsi4 = UNSELECT, sign4; // ubah data diri

while(openPage) {
    // tampilan -----
    // warna tanda saat memilih opsi
    sign1 = (opsi1 == SELECT) ? SIGNED : UNSIGNED;
    sign2 = (opsi2 == SELECT) ? SIGNED : UNSIGNED;
    sign3 = (opsi3 == SELECT) ? SIGNED : UNSIGNED;
}

```

```

sign4 = (opsi4 == SELECT) ? SIGNED : UNSIGNED;

system("cls");
cout << endl << endl;

// judul
cout << "\t"; karakter(177, 27, 9); // dekorasi atas
cout << " PENDUDUK "; karakter(177, 27, 9);

gotoxy(0, 18); cout << "\t"; karakter(177, 70, 9); // dekorasi bawah
gotoxy(0, 4);

// judul
cout << "\n\t Selamat datang, " << penduduk.namaLengkap << "!" << char(2)
<< endl << endl << endl << endl;

// menu
color(sign1); cout << "\t\t" << char(16);
color(opsi1); cout << " Keluar \n\n"; // opsi 1

color(sign2); cout << "\t\t" << char(16);
color(opsi2); cout << " Isi data diri \n\n"; // opsi 2

color(sign3); cout << "\t\t" << char(16);
color(opsi3); cout << " Tampilkan data diri\n\n"; // opsi 3

color(sign4); cout << "\t\t" << char(16);
color(opsi4); cout << " Ubah data diri \n\n\n"; // opsi 4

color(RESET);

// navigasi -----

// pilih menu
unsigned char menu, key;
menu = getch();

// key yg ditekan

```

```

switch (menu) {

    // tekan ENTER
    case ENTER:

        // kembali
        if (opsi1 == SELECT) {
            openPage = false;

        // isi formulir
        } else if (opsi2 == SELECT) {
            isiFormulirPenduduk(penduduk);

        // tampilkan data diri
        } else if (opsi3 == SELECT) {
            tampilkanData(penduduk);

        // ubah data diri
        } else if (opsi4 == SELECT) {
            ubahDataDiri(penduduk);
        }

        break;

    // tekan UP || DOWN
    case 224:
        key = getch();

        // tunjuk opsi 1 - keluar
        if (key == UP && opsi2 == SELECT) {
            opsi1 = SELECT;
            opsi2 = UNSELECT;

        // tunjuk opsi 2 - isi formulir
        } else if (
            key == DOWN && opsi1 == SELECT ||
            key == UP  && opsi3 == SELECT
        ) {

```

```

opsi2 = SELECT;
opsi1 = UNSELECT; opsi3 = UNSELECT;

// tunjuk opsi 3 - tampilkan data diri
} else if (
    key == DOWN && opsi2 == SELECT ||
    key == UP   && opsi4 == SELECT
) {
    opsi3 = SELECT;
    opsi2 = UNSELECT; opsi4 = UNSELECT;

// tunjuk opsi 4 - ubah data diri
} else if (key == DOWN && opsi3 == SELECT) {
    opsi4 = SELECT;
    opsi3 = UNSELECT;
}

break;
} // end switch

} // end while

}

// ## Modul 2 - Pointer
void isiFormulirPenduduk(Penduduk &penduduk) {
    // var yg dipakai
    bool openPage = true;
    short int page = 1;

    while (openPage) {
        system("cls");
        cout << endl << endl;

        // judul halaman
        cout << "\t";                 karakter(177, 5, 9); // dekorasi atas
        cout << " Formulir Biodata Penduduk "; karakter(177, 36, 9); // dekorasi bawah

```

```

cout << endl;

// jika formulir sudah diisi
if (formulirTerisi(penduduk)){
    page = 0; // tidak perlu mengisi
    color(SUCCESS); cout << "\n\n\t\tAnda telah mengisi formulir ini." << endl;
}

// isi formulir - halaman 1
if (page == 1) {
    cout << endl;

// tampilkan nama dan NIK
cout << "\tNama Lengkap : " << penduduk.namaLengkap << endl; // nama lengkap
cout << "\tNIK \t: " << penduduk.nik << endl; // NIK

// tampilkan formulir tempat tanggal lahir
cout << "\n\tTempat/Tanggal Lahir" << endl;
cout << " \t Tempat : \n"; color(8);
cout << "\n\t -- isi di bawah ini dengan angka \n\n"; color(7);
cout << " \t Tanggal : \n\n";
cout << " \t Bulan : \n\n";
cout << " \t Tahun : \n";

// tampilkan formulir no telepon
cout << "\n\tNo. Telepon\t: \n";

// isi tempat
string tempat;
while (isSpace(tempat)) {
    gotoxy(23, 8); hapusTulisan();
    gotoxy(23, 8); color(SELECT); getline(cin, tempat); fflush(stdin);

}

// isi tanggal, bulan, tahun
int tahunIni = timeNow()>tm_year + 1900;

```

```

short int jawaban[3], jarak = 0;

for (int i=0, i<3; i++) {

    // selama pengisian sesuai syarat
    while (true) {
        gotoxy(23, 12+jarak); hapusTulisan();
        gotoxy(23, 12+jarak); color(SELECT); cin >> jawaban[i];

        // jika isi telah sesuai syarat pengisian
        if (
            !cin.fail() &&                               // isi bukan string
            (
                i == 0 && isTanggal(jawaban[i]) || // syarat tanggal
                i == 1 && isBulan(jawaban[i]) || // syarat bulan
                i == 2 && isTahun(jawaban[i])      // syarat tahun
            )
        ) { clearCin(); break; }                      // berhenti isi

        clearCin(); // bersihkan cin
    }

    jarak += 2;
}

// simpan ttl
penduduk.ttl = tempat + "," + to_string(jawaban[0]) + "-"
    + to_string(jawaban[1]) + "-" + to_string(jawaban[2]);

// simpan usia
penduduk.usia = tahunIni - jawaban[2];

// isi nomor telepon
string noTelp;
while (!isTelp(noTelp)) {
    gotoxy(26, 18); hapusTulisan();
    gotoxy(26, 18); color(SELECT); getline(cin, noTelp); fflush(stdin);
}

```

```

// simpan nomor telepon
penduduk.telepon = noTelp;

color(RESET);

} else if ( page == 2 ) { // halaman 2
    cout << endl;

    // tampilkan formulir alamat
    cout << "\tAlamat      " << endl;
    cout << "\t Jalan : " << endl; color(8);
    cout << "\n\t-- isi dengan memilih angka pada opsi\n"; color(7);

    // tampilkan formulir kecamatan
    tampilkanOpsi("Kecamatan");

    // deklarasi var
    string jalan;
    short int noKec, noKel;

    // isi jalan
    while (isSpace(jalan)) {
        gotoxy(22, 5); hapusTulisan();
        gotoxy(22, 5); color(SELECT); getline(cin, jalan); fflush(stdin);
    }

    // isi kecamatan
    while (true) {
        gotoxy(13, 14); hapusTulisan();
        gotoxy(13, 14); color(SELECT); cin >> noKec;
        if (!cin.fail() && noKec > 0 && noKec <= 10) { clearCin(); break; }

        hapusTulisan; clearCin();
    }

    color(RESET);

```

```

// tampilkan formulir kelurahan
tampilkanOpsi(KECAMATAN[noKec-1]);

// isi kelurahan
while (true) {
    gotoxy(13, 20); hapusTulisan();
    gotoxy(13, 20); color(SELECT); cin >> noKel;

    if (
        !cin.fail() &&
        noKel >= 1 &&
        (
            noKec == 1 && noKel <= 5 || // Palaran          = 5 kelurahan
            noKec == 2 && noKel <= 6 || // Samarinda Seberang = 6 kelurahan
            noKec == 3 && noKel <= 8 || // Samarinda Ulu      = 8 kelurahan
            noKec == 4 && noKel <= 5 || // Samarinda Ilir     = 5 kelurahan
            noKec == 5 && noKel <= 8 || // Sungai Utara       = 8 kelurahan
            noKec == 6 && noKel <= 7 || // Sungai Kunjang      = 7 kelurahan
            noKec == 7 && noKel <= 5 || // Sambutan           = 5 kelurahan
            noKec == 8 && noKel <= 5 || // Sungai Pinang       = 5 kelurahan
            noKec == 9 && noKel <= 5 || // Samarinda Kota      = 5 kelurahan
            noKec == 10 && noKel <= 5 // Loa Janan Ilir      = 5 kelurahan
        )
    ) { clearCin(); break; }

    clearCin();
}

penduduk.alamat.jalan = jalan; // simpan jalan
penduduk.alamat.kecamatan = noKec - 1; // simpan kecamatan
penduduk.alamat.kelurahan = noKel - 1; // simpan kelurahan

// halaman 3
} else if ( page == 3 ) { // halaman 3

    // tampilkan formulir gender, agama, goldar, status kawin
    tampilkanOpsi("Jenis Kelamin");
    tampilkanOpsi("Agama");
}

```

```

tampilkanOpsi("Golongan Darah");
tampilkanOpsi("Status Perkawinan");

// deklarasi var
short int jawaban[4];
int jarak = 0;

// isi jenis kelamin, agama, golongan darah, status perkawinan
for (int i=0; i<4; i++) {

    while (true) {
        gotoxy(15, 6+jarak); hapusTulisan();
        gotoxy(15, 6+jarak); color(SELECT); cin >> jawaban[i];

        if (
            !cin.fail() &&
            jawaban[i] > 0 &&
            (
                i == 0 && jawaban[i] <= 2 || // ketentuan jenis kelamin
                i == 1 && jawaban[i] <= 6 || // ketentuan agama
                i == 2 && jawaban[i] <= 4 || // ketentuan golongan darah
                i == 3 && jawaban[i] <= 4 // ketentuan status perkawinan
            )
        ) { clearCin(); break; }

        clearCin();
    }

    if (i == 0) { jarak += 5; }
    else      { jarak += 4; }
}

penduduk.gender = jawaban[0]-1; // simpan jenis kelamin
penduduk.agama = jawaban[1]-1; // simpan agama
penduduk.golDar = jawaban[2]-1; // simpan golongan darah
penduduk.statusKawin = jawaban[3]-1; // simpan status perkawinan

penduduk.tanggalPembaruan = hariIni(); // tgl data pembaruan diubah

```

```

updateToTxt(penduduk);           // perbarui file eksternal

color(SUCCESS); cout << "\n\t"    Formulir telah berhasil diisi! \n" << endl;

}

gotoxy(0, 23);
color(SELECT); cout << "\t<= Keluar \t\t\t\t\t ";
if (page == 1 || page == 2) { cout << "Lanjutkan =>"; }

color(RESET); cout << endl << endl;

// navigasi -----
unsigned char opsi, key;
opsi = getch();

if (opsi == 224) {
    key = getch();

    // tekan LEFT
    if (key == LEFT) {

        // jika ingin keluar saat mengisi formulir
        if (
            !formulirTerisi(penduduk) && cancelForm() || // keluar saat belum selesai isi
            formulirTerisi(penduduk)                  // keluar saat sudah selesai isi
        ) {
            openPage = false;
            break;
        }

        page += 1;

        // ke halaman selanjutnya
    } else if (key == RIGHT && page == 1) {
        page = 2;
    }
}

```

```

} else if (key == RIGHT && page == 2) {
    page = 3;

} // end if

} // end while

}

bool cancelForm() {
    // var yg digunakan
    bool openNotif = true;
    short int ya1 = SIGNED, ya2,
        tidak1 = UNSIGNED, tidak2;

    while (openNotif) {
        // warna tanda pada opsi
        ya2 = (ya1 == SIGNED) ? SELECT : UNSELECT;
        tidak1 = (ya1 == UNSIGNED) ? SIGNED : UNSIGNED;
        tidak2 = (ya2 == UNSELECT) ? SELECT : UNSELECT;

        // batas atas
        gotoxy(17, 7);
        for (int i=0; i<32; i++) {
            if (i%2 == UNSIGNED) { karakter(177, 2, 9); }
            else { cout << " "; }
        }

        // batas bawah
        gotoxy(17, 17);
        for (int i=0; i<32; i++) {
            if (i%2 == UNSIGNED) { karakter(177, 2, 9); }
            else { cout << " "; }
        }
    }
}

```

```

// tampilan
color(RESET); gotoxy(16, 8);
cout << " +-----+ \n\t\t";
cout << " | \n\t\t| \n\t\t";
cout << " | Keluar saat ini tidak akan menyimpan | \n\t\t";
cout << " | data yang telah Anda isi. | \n\t\t";
cout << " | \n\t\t| \n\t\t";
cout << " | Apakah Anda yakin ingin keluar? | \n\t\t";
cout << " | \n\t\t| \n\t\t";
cout << " | \n\t\t| \n\t\t";
cout << " +-----+ \n\t\t";

// opsi ya atau tidak
gotoxy(28, 14); color(ya1); cout << char(16); color(ya2); cout << " Ya";
gotoxy(44, 14); color(tidak1); cout << char(16); color(tidak2); cout << " Tidak";

gotoxy(0, 22);
color(RESET); cout << endl << endl;

// respon user
unsigned char opsi, key;
opsi = getch();

switch (opsi) {

    // tekan ENTER
    case ENTER:

        // tidak
        if (tidak2 == SELECT) {
            openNotif = false;
            break;

        // ya
        } else if (ya2 == SELECT) {
            return true;
        }
}

```

```

        break;

    // tekan RIGHT || LEFT
    case 224:
        key = getch();

        // tunjuk ya
        if (tidak1 == SIGNED && key == LEFT) {
            ya1 = SIGNED;

            // tunjuk tidak
        } else if (ya1 == SIGNED && key == RIGHT) {
            ya1 = UNSIGNED;
        }

        break;

    } // end switch case

} // end while

return false;

}

void tampilanData(Penduduk penduduk) {
    bool openPage = true;

    while (openPage) {
        system("cls");
        cout << endl << endl;

        // judul
        cout << "\t"; karakter(177, 5, 9); // dekorasi atas
        cout << " Biodata Penduduk "; karakter(177, 45, 9);
        cout << endl << endl << endl;

        // jika belum isi formulir
    }
}

```

```

if (!formulirTerisi(penduduk)) {
    color(14);
    cout << endl << endl << endl << endl
        << "\t\t\tHarap isi formulir terlebih dahulu." << endl;

    goto kembali; //pergi ke kembali, lewatkan tampilan data
}

// tampilan data
cout << "\t Nama : " << penduduk.namaLengkap << endl;
cout << "\t NIK : " << penduduk.nik << endl;
cout << "\t Tempat/Tanggal Lahir : " << penduduk.ttl << endl;
cout << "\t Usia : " << penduduk.usia << endl
    << endl;
cout << "\t Alamat " << endl;
cout << "\t Jalan : " << penduduk.alamat.jalan << endl;
cout << "\t Kelurahan : " <<

KELURAHAN[penduduk.alamat.kecamatan][penduduk.alamat.kelurahan] << endl;
    cout << "\t Kecamatan : " << KECAMATAN[penduduk.alamat.kecamatan]
<< endl
    << endl;
    cout << "\t No. HP : " << penduduk.telepon << endl;
    cout << "\t Jenis Kelamin : " << GENDER[penduduk.gender] << endl;
    cout << "\t Agama : " << AGAMA[penduduk.agama] << endl;
    cout << "\t Golongan Darah : " << GOLDAR[penduduk.golDar] << endl;
    cout << "\t Status Perkawinan : " << STATUSKAWIN[penduduk.statusKawin]
<< endl;
    cout << "\t Status Hidup : " << STATUSHIDUP[penduduk.statusHidup] <<
endl;

kembali: //goto
gotoxy(0, 23);
color(SELECT); cout << "\t<= Kembali" << endl; color(RESET);

unsigned char opsi, key;
opsi = getch();

// jika tekan arrow keys

```

```

if (opsi == 224) {
    key = getch();

    // keluar halaman
    if (key == LEFT) {
        openPage = false;
    } // end if
} // end if

} // end while
}

void ubahDataDiri(Penduduk &penduduk) {

    // var yg dipakai
    bool openPage = true,
        editing = false;
    short int index,
    warna[11] = { // warna opsi
        SELECT, UNSELECT, UNSELECT, UNSELECT, UNSELECT,
        UNSELECT, UNSELECT, UNSELECT, UNSELECT, UNSELECT,
        UNSELECT
    },
    sign[11] = { // warna tanda
        SIGNED, UNSIGNED, UNSIGNED, UNSIGNED, UNSIGNED,
        UNSIGNED, UNSIGNED, UNSIGNED, UNSIGNED, UNSIGNED,
        UNSIGNED
    };

    string opsi[11] = { // opsi
        "Nama Lengkap", "NIK", "Password", "Tempat, Tanggal Lahir", "Jenis Kelamin",
        "Telepon", "Alamat", "Agama", "Golongan Darah", "Status Perkawinan",
        "Kembali"
    };

    while (openPage) {
        system("cls");

```

```

cout << endl << endl;

// judul halaman
cout << "\t"; karakter(177, 5, 9);
cout << " Mengubah Biodata Diri "; karakter(177, 40, 9);
cout << endl << endl << endl;

// jika belum isi formulir
if (!formulirTerisi(penduduk)) {
    color(14);
    cout << endl << endl << endl << endl
        << "\t\t\tHarap isi formulir terlebih dahulu." << endl;

// tunjuk kembali
warna[10] = SELECT; sign[10] = SIGNED;

} else {

// data yg ditampilkan
string dataTerisi[] = {
    penduduk.namaLengkap, // 0
    penduduk.nik, // 1
    penduduk.password, // 2
    penduduk.ttl, // 3
    GENDER[penduduk.gender], // 4
    penduduk.telepon, // 5
    penduduk.alamat.jalan, // 6
    KELURAHAN[penduduk.alamat.kecamatan][penduduk.alamat.kelurahan], // 7
    KECAMATAN[penduduk.alamat.kecamatan], // 8
    AGAMA[penduduk.agama], // 9
    GOLDAR[penduduk.golDar], // 10
    STATUSKAWIN[penduduk.statusKawin] // 11
};

// tampilkan opsi data yg ingin diubah
for (int i=0; i<10; i++) {
}

```



```

        } // end if
        } // end if
    } // end for

    // jika sedang mengedit
} else {

    // ganti tempat tanggal lahir
    if (index == 3) {

        // var untuk pengisian formulir
        string tempat,
        keterangan[3] = {"Tanggal", "Bulan", "Tahun"};
        int jawaban[3];

        // tempat
        cout << "\n\t\tTempat \t: ";

        // isi tempat
        while (isSpace(tempat)) {
            gotoxy(34, 18); hapusTulisan();
            gotoxy(34, 18); getline(cin, tempat); fflush(stdin);
        }

        // tanggal, bulan, tahun
        for (int i=0; i<3; i++) {
            cout << "\t\t " << keterangan[i] << " \t: ";

            // isi tanggal / bulan / tahun
            while (true) {
                gotoxy(34, 19+i); hapusTulisan();
                gotoxy(34, 19+i); cin >> jawaban[i];

                // jika isian telah benar, berhenti isi
                if (
                    !cin.fail() &&
                    jawaban[i] > 0 &&

```

```

(
    i == 0 && isTanggal(jawaban[i]) || // ketentuan tanggal
    i == 1 && isBulan(jawaban[i]) || // ketentuan bulan
    i == 2 && isTahun(jawaban[i]) // ketentuan tahun
)
) { clearCin(); break; }

clearCin();

}

}

// simpan ttl
penduduk.ttl = tempat + ", "
+ to_string(jawaban[0]) + "-" // tanggal
+ to_string(jawaban[1]) + "-" // bulan
+ to_string(jawaban[2]); // tahun

// ubah usia
int tahunIni = timeNow()>tm_year + 1900;
penduduk.usia = tahunIni - jawaban[2];

// ganti alamat
} else if (index == 6) {

    // var untuk pengisian
    string jalan;
    short int noKec, noKel;

    //jalan
    cout << "\n\t\t Jalan \t: ";
    while (isSpace(jalan)) {
        gotoxy(34, 18); hapusTulisan();
        gotoxy(34, 18); getline(cin, jalan); fflush(stdin);
    }

    // kecamatan
    gotoxy(0, 17);
}

```

```

tampilkanOpsi("Kecamatan");

// isi kecamatan
while (true) {
    gotoxy(15, 23); hapusTulisan();
    gotoxy(15, 23); cin >> noKec;

    // jika isian sudah benar, berhenti isi;
    if (!cin.fail() && noKec >= 1 && noKec <= 10) {
        clearCin; break;
    }
    clearCin();

}

// kelurahan
gotoxy(10, 22); hapusTulisan(100);
gotoxy(0, 17); tampilkanOpsi(KECAMATAN[noKec-1]);
cout << endl; hapusTulisan();

// isi kelurahan
while (true) {
    gotoxy(15, 22); hapusTulisan();
    gotoxy(15, 22); cin >> noKel;

    // jika isian sudah benar, berhenti isi
    if (
        !cin.fail() &&
        noKel >= 1 &&
        (
            noKec == 1 && noKel <= 5 ||
            noKec == 2 && noKel <= 6 ||
            noKec == 3 && noKel <= 8 ||
            noKec == 4 && noKel <= 5 ||
            noKec == 5 && noKel <= 8 ||
            noKec == 6 && noKel <= 7 ||
            noKec == 7 && noKel <= 5 ||
            noKec == 8 && noKel <= 5 ||

```

```

        noKec == 9 && noKel <= 5 ||
        noKec == 10 && noKel <= 5
    )
) { clearCin(); break; }

clearCin();
}

// ubah alamat
penduduk.alamat.jalan = jalan; // simpan jalan
penduduk.alamat.kecamatan = noKec-1; // simpan kecamatan
penduduk.alamat.kelurahan = noKel-1; // simpan kelurahan

// gender, agama, golongan darah, status perkawinan
} else if (index == 4 || index == 7 || index == 8 || index == 9){

int isiInt;           // deklarasi var
tampilkanOpsi(opsi[index]); // tampilkan formulir

while (true) {

// letak isian agama
if(index == 7) {
    gotoxy(15, 21); hapusTulisan();
    gotoxy(15, 21); cin >> isiInt;

// letak isian gender, golongan darah, status perkawinan
} else {
    gotoxy(15, 20); hapusTulisan();
    gotoxy(15, 20); cin >> isiInt;

}

// jika isian sudah benar, berhenti isi;
if(
    !cin.fail() &&
    isiInt >= 1 &&
(

```

```

        index == 4 && isiInt <= 2 || // syarat jenis kelamin
        index == 7 && isiInt <= 6 || // syarat agama
        index == 8 && isiInt <= 4 || // syarat golongan darah
        index == 9 && isiInt <= 4    // syarat status perkawinan
    )
) { clearCin; break; }
clearCin();

}

if (index == 4) { penduduk.gender = isiInt-1; } // simpan gender
else if (index == 7) { penduduk.agama = isiInt-1; } // simpan agama
else if (index == 8) { penduduk.golDar = isiInt-1; } // simpan goldar
else if (index == 9) { penduduk.statusKawin = isiInt-1; } // simpan status
kawin

// nama, password, telepon, nik
} else {

cout << "\n\t\t" << opsi[index] << " \t: "; // tampilkan formulir

// nama, password, telepon
if (index == 0 || index == 2 || index == 5) {
    string isiStr; // deklarasi var

    // isi nama || password || telepon
    while (isSpace(isiStr) || index == 5 && !isTelp(isiStr)) {
        gotoxy(34, 18); hapusTulisan();
        gotoxy(34, 18); getline(cin, isiStr); fflush(stdin);
    }

    if (index == 0) { penduduk.namaLengkap = isiStr; } // simpan nama
lengkap
    else if (index == 2) { penduduk.password = isiStr; } // simpan password
    else if (index == 5) { penduduk.telepon = isiStr; } // simpan telepon

    // nik
} else {
}

```

```

char nik[17]; // deklarasi var

// isi nik
while (!isNIK(nik) || searchNIK(nik) != -1 && string(nik) != string(penduduk.nik)) {
    gotoxy(27, 18); hapusTulisan();
    gotoxy(27, 18); cin.get(nik, 17); clearCin();
}
clearCin();

strcpy(penduduk.nik, nik); // simpan nik

} // end if

} // end if - ubah data atribut

editing = false;

// update
updateToTxt(penduduk);

warna[index] = SELECT;
sign[index] = SIGNED;
continue;

} // end if - editing

} // end if - formulir terisi

gotoxy(0, 23);

color(warna[10]); cout << "\tKembali" << endl;
color(sign[10]); cout << "\t <= " << endl;

cout << endl << endl; color(RESET);

// navigasi -----
unsigned char opsi, key;

```

```

opsi = getch();

switch (opsi) {

    // tekan ENTER
    case ENTER:

        index = indexElemen(warna, SELECT, 11);
        warna[index] = UNSELECT;
        sign[index] = MARKED;
        editing = true;

        break;

    // tekan UP || RIGHT || DOWN || LEFT
    case 224:
        key = getch();

        index = indexElemen(warna, SELECT, 11);

        // tekan DOWN
        if (key == DOWN && index+1 <= 10) {
            warna[index] = UNSELECT;
            sign[index] = UNSIGNED;

            if (index == 4) { index += 6; }
            else { index += 1; }

        // tekan RIGHT
        } else if (key == RIGHT && index+5 < 10) {
            warna[index] = UNSELECT;
            sign[index] = UNSIGNED;

            index += 5;

        // tekan UP
        } else if (key == UP && index-1 >= 0) {
            warna[index] = UNSELECT;
}

```

```

sign[index] = UNSIGNED;

if (index == 10) { index -= 6; }
else { index -= 1; }

// tekan LEFT
} else if (key == LEFT && index-5 >= 0) {
    if (index == 10) { openPage = false; break; }

warna[index] = UNSELECT;
sign[index] = UNSIGNED;

index -= 5;

// keluar halaman
} else if (key == LEFT && warna[10] == SELECT) {
    openPage = false;
    break;
}

warna[index] = SELECT;
sign[index] = SIGNED;

break;

} // end switch

}

}

// pemerintah -----
=====

void logInPemerintah() {
    system("cls"); cout << endl << endl;

    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 18); // batas atas
    cout << "\t"; karakter(177, 70, 9); gotoxy(0, 5); // batas bawah

```

```

cout << "\t\tLOGIN PEMERINTAH -*- \n\n\n" << endl; // judul

// masukkan username dan password
string username, password;
cout << "\t\t Username : "; color(SELECT); cin >> username; clearCin();
color(RESET);
cout << "\t\t Password : "; color(SELECT); cin >> password; clearCin();
color(RESET);
cout << endl << endl << endl;

// berhasil masuk
if (username == "pemerintah" && password == "123") {

    color(SUCCESS); cout << "\t\t\t Log in berhasil! \n\n\n" << endl;
    color(RESET); getch();

    menuPemerintah();

    // gagal masuk
} else {

    color(FAIL); cout << "\t\t\t Log in gagal! \n\n\n" << endl;
    color(RESET); getch();
}

void menuPemerintah() {

    // var yg digunakan
    bool openPage = true;
    short int opsi[5] = {SELECT, UNSELECT, UNSELECT, UNSELECT,
    UNSELECT}, // warna tulisan menu
    index; // index warna yg di-SELECT

    while(openPage) {

```

```

// tampilan -----
// warna tanda bergantung dengan warna opsi
int sign[5] = {
    (opsi[0] == SELECT) ? SIGNED : UNSIGNED, // tampilkan data penduduk
    (opsi[1] == SELECT) ? SIGNED : UNSIGNED, // tampilkan diagram
    (opsi[2] == SELECT) ? SIGNED : UNSIGNED, // perbarui status hidup
    (opsi[3] == SELECT) ? SIGNED : UNSIGNED, // hapus data penduduk
    (opsi[4] == SELECT) ? SIGNED : UNSIGNED // keluar
};

system("cls");

// judul dan dekorasi
gotoxy(0, 2); cout << "\t"; karakter(177, 26, 9); // dekorasi atas
cout << " PEMERINTAH "; karakter(177, 26, 9);
gotoxy(0, 18); cout << "\t"; karakter(177, 70, 9); // dekorasi bawah

gotoxy(0, 4); // reset posisi

// judul
cout << "\n\t Selamat datang, Pemerintah! " << char(2)
    << "\n\t Total penduduk : " << banyakData() << "."
    << endl << endl << endl;

// menu
color(sign[0]); cout << "\t\t " << char(16); // opsi[0]
color(opsi[0]); cout << " Tampilkan data" << "\n\t\t "
    << " penduduk " << "\n\n";

color(sign[1]); cout << "\t\t " << char(16); // opsi[1]
color(opsi[1]); cout << " Tampilkan hasil" << "\n\t\t "
    << " pendataan " << "\n\n";

gotoxy(0, 9);
color(sign[2]); cout << "\t\t\t\t\t\t " << char(16); // opsi[2]
color(opsi[2]); cout << " Perbarui status" << "\n\t\t\t\t\t\t "
    << " penduduk " << "\n\n";

```

```

color(sign[3]); cout << "\t\t\t\t\t\t " << char(16); // opsi[3]
color(opsi[3]); cout << " Hapus data " << "\n\t\t\t\t\t\t "
<< " penduduk " << "\n\n";

color(sign[4]); cout << "\t\t\t\t\t " << char(16); // opsi[4]
color(opsi[4]); cout << " Keluar " << "\n\n";
color(RESET);

// navigasi -----
unsigned char menu, key;
menu = getch();

switch (menu) {
    // tekan ENTER
    case ENTER:

        // tampilkan data penduduk
        if (opsi[0] == SELECT) {
            tampilkanDataPenduduk();

        // tampilkan diagram data
        } else if (opsi[1] == SELECT) {
            tampilkanDiagram();

        // perbarui status
        } else if (opsi[2] == SELECT) {
            ubahStatusHidup();

        // hapus data penduduk
        } else if (opsi[3] == SELECT) {
            hapusDataPenduduk();

        // kembali
        } else if (opsi[4] == SELECT) {
            openPage = false;

        }
}

```

```

break;

// tekan UP || RIGHT || DOWN || LEFT
case 224:

    // key yg ditekan
    key = getch();

    // opsi yang di-SELECT
    index = indexElemen(opsi, SELECT, 5);

    // tunjuk opsi[0] - tampilkan data penduduk
    if (
        key == UP && opsi[1] == SELECT ||
        key == LEFT && opsi[2] == SELECT
    ) {
        opsi[0] = SELECT;

        // tunjuk opsi[1] - tampilkan diagram
    } else if (
        key == DOWN && opsi[0] == SELECT ||
        key == LEFT && opsi[3] == SELECT ||
        key == UP && opsi[4] == SELECT ||
        key == LEFT && opsi[4] == SELECT
    ) {
        opsi[1] = SELECT;

        // tunjuk opsi[2] - perbarui status penduduk
    } else if (
        key == RIGHT && opsi[0] == SELECT ||
        key == UP && opsi[3] == SELECT
    ) {
        opsi[2] = SELECT;

        // tunjuk opsi[3] - hapus data penduduk
    } else if (
        key == DOWN && opsi[2] == SELECT ||

```

```

        key == RIGHT && opsi[1] == SELECT ||
        key == RIGHT && opsi[4] == SELECT
    ) {
        opsi[3] = SELECT;

        // tunjuk opsi[4] - keluar
    } else if (
        key == DOWN && (opsi[1] == SELECT || opsi[3] == SELECT)
    ) {
        opsi[4] = SELECT;
    }

    // opsi yg di-SELECT sebelumnya di-UNSELECT
    opsi[index] = UNSELECT;
    if (indexElemen(opsi, SELECT, 5) == -1) {
        opsi[index] = SELECT;
    }

    break;

} // end switch

} // end while

}

void tampilanDiagram() {

    // deklarasi var yg digunakan
    bool openPage = true;
    string kategori = "Jenis Kelamin";
    short int warna[8] = { // warna opsi
        SELECT, UNSELECT, UNSELECT, UNSELECT,
        UNSELECT, UNSELECT, UNSELECT, UNSELECT
    }, tanda[8] = { // warna tanda
        SIGNED, UNSIGNED, UNSIGNED, UNSIGNED,
        UNSIGNED, UNSIGNED, UNSIGNED, UNSIGNED
    }
}

```

```

}, index = 0;           // index di-SELECT

while (openPage) {
    system("cls");
    cout << endl << endl;

    // judul halaman
    cout << "\t";           karakter(177, 5, 9);
    cout << " Diagram Data Penduduk "; karakter(177, 40, 9);
    cout << endl << endl;

    // kategori diagram
    cout << "\t" ; color(tanda[0]);
    cout << char(16); color(warna[0]); cout << " Kecamatan";

    cout << " " ; color(tanda[1]);
    cout << char(16); color(warna[1]); cout << " Status Hidup ";

    cout << " " ; color(tanda[2]);
    cout << char(16); color(warna[2]); cout << " Jenis Kelamin";

    cout << "\t " ; color(tanda[3]);
    cout << char(16); color(warna[3]); cout << " Usia";

    cout << "\n\t" ; color(tanda[4]);
    cout << char(16); color(warna[4]); cout << " Agama";

    cout << "\t\t" ; color(tanda[5]);
    cout << char(16); color(warna[5]); cout << " Golongan Darah";

    cout << " " ; color(tanda[6]);
    cout << char(16); color(warna[6]); cout << " Status Perkawinan";

    color(RESET); cout << endl << endl;

    // jika belum ada yg mengisi formulir
    if (banyakDataTerisi() == 0) {
        color(14);

```

```

cout << endl << endl
<< "\t\t\tBelum ada data penduduk yang tersimpan" << endl;

// jika sudah ada yg mengisi formulir
} else {

    // tampilkan diagram
    if (warna[0] == SELECT || tanda[0] == MARKED) { diagramKecamatan(); }
    else if (warna[1] == SELECT || tanda[1] == MARKED) { diagramStatusHidup(); }
    else if (warna[2] == SELECT || tanda[2] == MARKED) { diagramGender(); }
    else if (warna[3] == SELECT || tanda[3] == MARKED) { diagramUsia(); }
    else if (warna[4] == SELECT || tanda[4] == MARKED) { diagramAgama(); }
    else if (warna[5] == SELECT || tanda[5] == MARKED) { diagramGolDar(); }
    else if (warna[6] == SELECT || tanda[6] == MARKED) { diagramStatusKawin(); }

}

}

gotoxy(0, 18); color(RESET); // reset posisi dan warna

// banyak data tersimpan
cout << "\tPenduduk terdaftar : " << banyakData() << endl
<< "\tPenduduk terverifikasi : " << banyakDataTerisi() << endl << endl << endl;

// ganti halaman
color(warna[7]); cout << "\tKembali\n";
color(tanda[7]); cout << "\t <= ";
color(RESET); cout << endl << endl << endl;

// navigasi -----

unsigned char opsi, key;
opsi = getch();

// tekan UP || RIGHT || DOWN || LEFT
if (opsi == 224) {
    key = getch();
}

```

```
index = indexElemen(warna, SELECT, 8);

// UNSELECT opsi yg sudah tidak ditunjuk
warna[index] = UNSELECT;
tanda[index] = UNSIGNED;

// tekan key RIGHT
if (key == RIGHT) {

    index += 1;
    if (index > 6) { index = 3; }

    // tekan key LEFT
} else if (key == LEFT) {

    // keluar halaman
    if (index == 7) {
        openPage = false;
    }

    index -= 1;
    if (index < 0) { index = 0; }

    // tekan DOWN
} else if (key == DOWN) {

    // tunjuk kembali
    if (index > 3) {
        tanda[index] = MARKED;
        index = 7;

    } else if (index == 3) {
        index = 6;

    } else {
        index += 4;
    }
}
```

```

// tekan UP
} else if (key == UP) {

    if (index == 7) {
        index = indexElemen(tanda, MARKED, 8);

    } else if (index > 3) {
        index -= 4;
    }

}

// SELECT opsi yg ditunjuk
warna[index] = SELECT;
tanda[index] = SIGNED;

} // end if

} // end while

}

void tampilanDataPenduduk() {
    // deklarasi var yg digunakan
    bool openPage = true;
    string mode = "ASCENDING";
    short int sort1 = SIGNED, sort2 = UNSIGNED, sort3 = UNSIGNED,
            cari1 = UNSIGNED, cari2 = UNSELECT,
            nav = UNSELECT,
            indexAwal = 0, page;

    while (openPage) {

        // halaman
        page = indexAwal/10 + 1;

        system("cls");
    }
}

```

```

cout << endl << endl;

// warna tanda search nik
cari2 = (cari1 == SIGNED) ? SELECT : UNSELECT;

// judul halaman
cout << "\t"; karakter(177, 5, 9);
cout << " Data Penduduk "; karakter(177, 48, 9);
cout << endl << endl;

cout << "\t\t\t\t" << mode << endl; color(8);
cout << "\t\t\t\t\b[Tekan SPACE untuk mengubah mode sorting]" << endl;
color(RESET);

// sort berdasarkan nama lengkap
if (sort1 == SIGNED || sort1 == MARKED) {
    sortNamaLengkap(0, banyakData()-1, mode);

// sort berdasarkan NIK
} else if (sort2 == SIGNED || sort2 == MARKED) {
    sortNIK(mode);

// sort berdasarkan tanggal diperbarui
} else if (sort3 == SIGNED || sort3 == MARKED) {
    sortTanggal(mode);
}

// tampilkan data dalam tabel
int jarak = 0;
for (int i=indexAwal; i<10+indexAwal; i++) {
    gotoxy(10, 9+jarak); cout << i+1;
    gotoxy(16, 9+jarak); cout << " " << dataPenduduk[i].namaLengkap;
    gotoxy(39, 9+jarak); cout << " " << dataPenduduk[i].nik;
    gotoxy(58, 9+jarak); cout << " " << dataPenduduk[i].tanggalPembaruan;

    jarak++;
}

```

```

// tampilkan tabel
tabel(sort1, sort2, sort3);

// cari nik
char nik[17];
color(cari1); cout << "\n\t" << char(16);
color(cari2); cout << " Masukkan NIK : ";

// menunjuk ke cari nik
if (cari1 == SIGNED) {
    color(8); cout << "[Tekan ENTER untuk mengisi]";

// mode cari nik
} else if (cari1 == MARKED) {
    color(SELECT); cin.get(nik, 17); clearCin();
    cari1 = SIGNED;

// searching NIK
int indexNIK = searchNIK(nik);

// jika ditemukan
if (indexNIK != -1) {
    tampilkanData(dataPenduduk[indexNIK]);
}

continue;
}

color(RESET); cout << endl << endl;

// ganti halaman
color(nav); cout << "\t<= Kembali " << "\t\t\t\t\t\t"
             << " Berikutnya => ";
color(RESET); cout << endl << endl << endl;

// navigasi -----
unsigned char opsi, key;

```

```

opsi = getch();

switch (opsi) {
    // tekan SPACE
    case ' ':
        if (mode == "ASCENDING") {
            mode = "DESCENDING";
        } else {
            mode = "ASCENDING";
        }

        break;

    // tekan ENTER
    case ENTER:
        if (cari1 == SIGNED) {
            cari1 = MARKED;
        }

        break;

    // tekan UP || RIGHT || DOWN || LEFT
    case 224:
        key = getch();

        // tunjuk nama lengkap
        if (
            key == LEFT && sort2 == SIGNED ||
            key == UP   && sort1 == MARKED && cari1 == SIGNED
        ) {
            sort1 = SIGNED;
            sort2 = UNSIGNED; cari1 = UNSIGNED;

        // tunjuk nik
        } else if (
            key == RIGHT && sort1 == SIGNED ||
            key == LEFT  && sort3 == SIGNED ||

```

```

key == UP  && sort2 == MARKED && cari1 == SIGNED
) {
    sort2 = SIGNED;
    sort1 = UNSIGNED; sort3 = UNSIGNED; cari1 = UNSIGNED;

    // tunjuk tanggal diperbarui
} else if (
    key == RIGHT && sort2 == SIGNED ||
    key == UP  && sort3 == MARKED && cari1 == SIGNED
) {
    sort3 = SIGNED; sort2 = UNSIGNED; cari1 = UNSIGNED;

    // tunjuk cari
} else if (
    key == DOWN && (sort1 == SIGNED || sort2 == SIGNED || sort3 ==
SIGNED) ||
    key == UP && nav == SELECT
) {
    if    (sort1 == SIGNED) { sort1 = MARKED; }
    else if(sort2 == SIGNED) { sort2 = MARKED; }
    else if(sort3 == SIGNED) { sort3 = MARKED; }

    cari1 = SIGNED;
    nav = UNSELECT;

    // tunjuk arah halaman
} else if (key == DOWN && cari1 == SIGNED) {
    nav = SELECT;
    cari1 = UNSIGNED;

    // ke halaman sebelumnya
} else if (key == LEFT && nav == SELECT) {
    indexAwal -= 10;

    // jika di halaman pertama
    if(indexAwal < 0) {
        openPage = false;
        break;
    }
}

```

```

        }

        // ke halaman selanjutnya
    } else if (key == RIGHT && nav == SELECT) {
        indexAwal += 10;

        // jika di halaman terakhir
        if (indexAwal > banyakData()-1) {
            indexAwal -= 10;
            continue;
        }

    } // end if
    break;
} // end switch
} // end while

}

void ubahStatusHidup() {

    // deklarasi var
    bool openPage = true;
    int index = -1;

    while (openPage) {

        system("cls");
        cout << endl << endl;

        // judul halaman
        cout << "\t";           karakter(177, 5, 9);
        cout << " Ubah Status Hidup Penduduk "; karakter(177, 35, 9);
        cout << endl << endl;

        // searching NIK
        char ubahNIK[17];

```

```

cout << "\tMasukkan NIK yang ingin diubah : ";
color(SELECT); cin >> ubahNIK; color(RESET); clearCin();

cout <<
"\n\t====="
=====" << endl;

// proses searching
index = searchNIK(ubahNIK);

cout << "\tHasil pencarian "; color(14);
cout << ubahNIK; color(RESET);
cout << " ditemukan." << endl << endl;

// jika NIK tidak ditemukan
if (index == -1) {
    cout <<
"\t====="
===== \n" << endl; color(FAIL);
    cout << "\t      NIK Penduduk yang Anda Cari Tidak Ada      " <<
endl << endl << endl;

// jika NIK ditemukan
} else {
    cout << "\t      Apakah Anda yakin ingin mengubah status hidup      "
<< endl
    << "\t      data penduduk berikut ini?      " << endl <<
endl;

// tampilkan data yg ingin diubah
cout << "\t Nama : " <<
dataPenduduk[index].namaLengkap << endl;
cout << "\t NIK : " << dataPenduduk[index].nik <<
endl;
cout << "\t Status Hidup : " <<
STATUSHIDUP[dataPenduduk[index].statusHidup] << endl;

```

```

cout <<
"\n\t=====
===== " << endl;
cout << "\n\t      Tekan [Enter] untuk Melanjutkan . . ." << endl;

// penanda editing
bool editing = false, // sedang mengubah
      edited = false; // sudah diubah
short int updateStatusHidup;

unsigned char respon = getch();
cout << endl;

// jika mengubah
if (respon == ENTER) {
    editing = true;

    short int notif = SIGNED,
              opsi1 = SELECT, sign1, // wafat
              opsi2 = UNSELECT, sign2, // hidup
              opsi3 = UNSELECT, sign3; // batalkan

    // selama mengubah data
    while (editing) {
        gotoxy(0, 18);

        // proses updating

        sign1 = (opsi1 == SELECT) ? SIGNED : UNSIGNED;
        sign2 = (opsi2 == SELECT) ? SIGNED : UNSIGNED;
        sign3 = (opsi3 == SELECT) ? SIGNED : UNSIGNED;

        cout << "\t\t\t Ubah status hidup menjadi : \n\n"; // judul
        color(notif); cout << "\t\t\t --- \n\n";
    }
}

// opsi yang tersedia
color(sign1); cout << "\t\t\t " << char(16); // opsi 1
color(opsi1); cout << " Wafat" ;

```

```

color(sign2); cout << "\t\t\t\t " << char(16); // opsi 2
color(opsi2); cout << " Hidup\n";

color(sign3); cout << "\t\t\t\t " << char(16); // opsi 3
color(opsi3); cout << " Batalkan \n\n";

color(RESET);

// navigasi -----

// pilih mode
unsigned char opsi, key;
opsi = getch();
string jump;

// opsi yg tersedia untuk mengedit
switch (opsi) {

    // tekan ENTER
    case ENTER:

        // tekan opsi wafat
        if (opsi1 == SELECT) {
            updateStatusHidup = 0; // wafat
            edited = true;
            editing = false;

        // tekan opsi hidup
    } else if (opsi2 == SELECT) {
        updateStatusHidup = 1; // hidup
        edited = true;
        editing = false;

    // kembali
} else if (opsi3 == SELECT) {
    editing = false;
}
}

```

```

break;

// tekan UP || RIGHT || DOWN || LEFT
case 224:
    key = getch();

    // pilih opsi 1
    if (key == LEFT || key == UP && opsi3 == SELECT) {
        opsi1 = SELECT;
        opsi2 = UNSELECT;
        opsi3 = UNSELECT;

    // pilih opsi 2
    } else if (key == RIGHT) {
        opsi2 = SELECT;
        opsi1 = UNSELECT; opsi3 = UNSELECT;

    // pilih opsi 3
    } else if (key == DOWN) {
        opsi3 = SELECT;
        opsi1 = UNSELECT; opsi2 = UNSELECT;
    }

    break;

// opsi tidak ada
default:
    notif = FAIL;
    continue; // lanjutkan openPage

    break;

} // end switch

notif = SIGNED;

} // end while

```

```

}

// jika status hidup diubah
if (edited) {

    // ubah status hidup
    dataPenduduk[index].statusHidup = updateStatusHidup;
    updateToTxt(dataPenduduk[index]);

    gotoxy(0, 22); color(SUCCESS);
    cout << "\t" Status hidup pada data penduduk ini " << endl
    << "\t" telah berhasil diubah. " << endl << endl;

// jika batal mengubah
} else {
    if (respon == 224) { getch(); }
    gotoxy(0, 18); color(FAIL);
    cout << "\t" Aktivitas dibatalkan. " << endl;

    for (int i=0; i<5; i++) {
        cout << "\t" " << endl;
    }
}

color(RESET);

} // end if

gotoxy(0, 23); Sleep(1000); color(SELECT);
cout << "\t"= Kembali" << endl; color(RESET);

unsigned char respon, key;
respon = getch();

if (respon == 224) {
    key = getch();
}

```

```

        if (key == LEFT) { openPage = false; break;}

    } // end if

} // end while

}

void hapusDataPenduduk() {
    system("cls");
    cout << endl << endl;

// judul halaman
cout << "\t";           karakter(177, 5, 9);
cout << " Hapus Data Penduduk "; karakter(177, 42, 9);
cout << endl << endl;

// searching NIK
char hapusNIK[17];
cout << "\tMasukkan NIK yang ingin dihapus : ";
color(SELECT); cin >> hapusNIK; color(RESET); clearCin();

cout <<
"\n\t=====
====="
<< endl;

// proses searching
int index = searchNIK(hapusNIK);

cout << "\tHasil pencarian "; color(14);
cout << hapusNIK << "."
<< endl   << endl;   color(RESET);

// jika NIK tidak ditemukan
if (index == -1) {

```

```

cout<<
"\t=====
\n" << endl; color(FAIL);
    cout<<"\t      NIK Penduduk yang Anda Cari Tidak Ada      " <<
endl;

// jika NIK ditemukan
} else {
    cout << "\t      Apakah Anda yakin ingin menghapus data      " <<
endl
    << "\t      penduduk berikut ini?      " << endl << endl;

// tampilkan data yg ingin dihapus
cout << "\t Nama : " << dataPenduduk[index].namaLengkap <<
endl;
cout << "\t NIK : " << dataPenduduk[index].nik << endl;
cout << "\t Status Hidup : " <<
STATUSHIDUP[dataPenduduk[index].statusHidup] << endl << endl;

// masukkan inputan
char hapus[17];
cout << "\tKetik "; color(14); cout << dataPenduduk[index].nik; color(RESET);
cout << " untuk menghapus. \n"
    << "\t : ";

color(SELECT); cin.get(hapus, 17); clearCin();
color(RESET);

cout <<
"\n\t=====
" << endl;

// benar-benar menghapus
if (string(hapus) == string(dataPenduduk[index].nik)) {
    // proses deleting
    deleteFromTxt(dataPenduduk[index]);

    color(SUCCESS);
}

```

```

cout << "\t           Sukses... Data Berhasil Terhapus.          " << endl;

// batal hapus
} else {

    color(FAIL);
    cout << "\t           Menghapus data dibatalkan.          " << endl;
}

}

color(RESET);
cout <<
"\t=====
====="
<< endl;

}

getch(); color(RESET);

}

// diagram
void diagramKecamatan() {

// var yg digunakan
float total[10] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
      totalData = banyakDataTerisi();
short int warna[10] = {1, 2, 15, 5, 12, 6, 3, 8, 13, 14};

// atur angka belakang koma
cout << setiosflags(ios::fixed) << setprecision(1);

for (int i=0; i<10; i++) {

    // hitung jumlah data per gender
    for (int j=0; j<banyakData(); j++) {
        if (dataPenduduk[j].alamat.kecamatan == i) {
            total[i]++;
    }
}
}
}

```

```

        }

    }

    // tampilkan diagram
    diagram((total[i]*30)/(totalData), warna[i]);
    cout << " " << (total[i]*100)/(totalData) << "%" << endl;

}

// keterangan warna
for (int i=0; i<10; i++) {
    gotoxy(55, 7+i);
    color(warna[i]); cout << char(254) << " ";
    color(RESET); cout << KECAMATAN[i];
}

cout << endl << endl;
}

void diagramStatusHidup() {
    float total[2] = {0, 0},
        totalData = banyakData();
    short int warna[2] = {8, 14};

    // atur angka belakang koma
    cout << setiosflags(ios::fixed) << setprecision(1);

    // hitung jumlah data per status
    for (int i=0; i<banyakData(); i++) {
        if (dataPenduduk[i].statusHidup == 0) { total[0]++; }
        else if (dataPenduduk[i].statusHidup == 1) { total[1]++; }
    }

    // tampilkan diagram
    for (int i=0; i<10; i++) {

        if (i == 2) {          // hidup
            diagram((total[1]*30)/(totalData), warna[1]);
        }
    }
}

```

```

cout << " " << (total[1]*100)/(totalData) << "%" << endl;

} else if (i == 7) { //wafat
    diagram((total[0]*30)/(totalData), warna[0]);
    cout << " " << (total[0]*100)/(totalData) << "%" << endl;

} else {
    diagram(0); cout << endl;
}

}

// keterangan warna
gotoxy(55, 9); color(warna[1]); cout << char(254) << " "; color(RESET); cout <<
STATUSHIDUP[1];
gotoxy(55, 14); color(warna[0]); cout << char(254) << " "; color(RESET); cout <<
STATUSHIDUP[0];

cout << endl << endl << endl;

}

void diagramGender() {
float total[2] = {0, 0},
    totalData = banyakDataTerisi();
short int warna[2] = {9, 13};

// atur angka belakang koma
cout << setiosflags(ios::fixed) << setprecision(1);

// hitung jumlah data per gender
for (int j=0; j<banyakData(); j++) {
    if (dataPenduduk[j].gender == 0) { total[0] += 1; } // laki-laki
    else if (dataPenduduk[j].gender == 1) { total[1] += 1; } // perempuan
}

// tampilkan diagram
for (int i=0; i<10; i++) {

```

```

if (i == 2) {          // laki-laki
    diagram((total[0]*30)/(totalData), warna[0]);
    cout << " " << (total[0]*100)/(totalData) << "%" << endl;

} else if (i == 7) {   // perempuan
    diagram((total[1]*30)/(totalData), warna[1]);
    cout << " " << (total[1]*100)/(totalData) << "%" << endl;

} else {
    diagram(0); cout << endl;
}

}

// keterangan warna
gotoxy(55, 9); color(warna[0]); cout << char(254) << " "; color(RESET); cout <<
GENDER[0];
gotoxy(55, 14); color(warna[1]); cout << char(254) << " "; color(RESET); cout <<
GENDER[1];

}

void diagramUsia() {
float total[7] = {0, 0, 0, 0, 0, 0, 0},
      totalData = banyakDataTerisi();
short int warna[7] = {8, 9, 10, 14, 12, 13, 6};

cout << setiosflags(ios::fixed) << setprecision(1);

// hitung jumlah data per usia
for (int j=0; j<banyakData(); j++) {
    if (dataPenduduk[j].usia >= 0 && dataPenduduk[j].usia <= 9) { total[0]++; }
    else if (dataPenduduk[j].usia >= 10 && dataPenduduk[j].usia <= 19) { total[1]++; }
    else if (dataPenduduk[j].usia >= 20 && dataPenduduk[j].usia <= 29) { total[2]++; }
    else if (dataPenduduk[j].usia >= 30 && dataPenduduk[j].usia <= 39) { total[3]++; }
    else if (dataPenduduk[j].usia >= 40 && dataPenduduk[j].usia <= 49) { total[4]++; }
    else if (dataPenduduk[j].usia >= 50 && dataPenduduk[j].usia <= 59) { total[5]++; }
    else if (dataPenduduk[j].usia >= 60)           { total[6]++; }
}

```

```

    }

    // tampilkan diagram
    for (int i=0; i<9; i++) {

        if (i == 0 || i == 8) {
            diagram(0); cout << endl;

        } else {
            diagram(total[i-1]*30/totalData, warna[i-1]);
            cout << " " << total[i-1]*100/totalData << "%" << endl;
        }
    }

    // keterangan
    gotoxy(55, 8); color(warna[0]); cout << char(254) << " "; color(RESET); cout << "0-9
tahun";
    gotoxy(55, 9); color(warna[1]); cout << char(254) << " "; color(RESET); cout << "10-
19 tahun";
    gotoxy(55, 10); color(warna[2]); cout << char(254) << " "; color(RESET); cout <<
"20-29 tahun";
    gotoxy(55, 11); color(warna[3]); cout << char(254) << " "; color(RESET); cout <<
"30-39 tahun";
    gotoxy(55, 12); color(warna[4]); cout << char(254) << " "; color(RESET); cout <<
"40-49 tahun";
    gotoxy(55, 13); color(warna[5]); cout << char(254) << " "; color(RESET); cout <<
"50-59 tahun";
    gotoxy(55, 14); color(warna[6]); cout << char(254) << " "; color(RESET); cout <<
"60+ tahun";

    cout << endl;
}

void diagramAgama() {
    float total[6] = {0, 0, 0, 0, 0, 0},

```

```

totalData = banyakDataTerisi();

short int warna[6] = {10, 13, 8, 14, 9, 12};

cout << setiosflags(ios::fixed) << setprecision(1);

// hitung jumlah data per agama
for (int i=0; i<banyakData(); i++) {
    if (dataPenduduk[i].agama == 0) { total[0]++; }
    else if (dataPenduduk[i].agama == 1) { total[1]++; }
    else if (dataPenduduk[i].agama == 2) { total[2]++; }
    else if (dataPenduduk[i].agama == 3) { total[3]++; }
    else if (dataPenduduk[i].agama == 4) { total[4]++; }
    else if (dataPenduduk[i].agama == 5) { total[5]++; }
}

// tampilkan diagram
for (int i=0; i<10; i++) {
    if (i < 2 || i > 7) {
        diagram(0); cout << endl;
    } else {
        diagram((total[i-2]*30)/totalData, warna[i-2]);
        cout << " " << (total[i-2]*100)/totalData << "%" << endl;
    }
}

// keterangan warna
for (int i=0; i<6; i++) {
    gotoxy(55, 9+i);
    color(warna[i]); cout << char(254) << " ";
    color(RESET); cout << AGAMA[i];
}

void diagramGolDar() {
    float total[4] = {0, 0, 0, 0},
          totalData = banyakDataTerisi();
}

```

```

short int warna[4] = {12, 3, 8, 15};

cout << setiosflags(ios::fixed) << setprecision(1);

// hitung jumlah data per agama
for (int i=0; i<banyakData(); i++) {
    if (dataPenduduk[i].golDar == 0) { total[0]++; }
    else if (dataPenduduk[i].golDar == 1) { total[1]++; }
    else if (dataPenduduk[i].golDar == 2) { total[2]++; }
    else if (dataPenduduk[i].golDar == 3) { total[3]++; }
}

// tampilkan diagram
for (int i=0; i<9; i++) {

    if (i%2 == 0) {
        diagram(0); cout << endl;

    } else {
        diagram((total[i/2]*30)/totalData, warna[i/2]);
        cout << " " << (total[i/2]*100)/totalData << "%" << endl;
    }
}

// keterangan warna
int jarak = 0;
for (int i=0; i<4; i++) {

    gotoxy(55, 8+jarak);
    color(warna[i]); cout << char(254) << " ";
    color(RESET); cout << GOLDAR[i];

    jarak += 2;
}
}

```

```

void diagramStatusKawin() {
    float total[4] = {0, 0, 0, 0},
        totalData = banyakDataTerisi();

    short int warna[4] = {3, 10, 6, 13};

    cout << setiosflags(ios::fixed) << setprecision(1);

    // hitung jumlah data per status kawin
    for (int i=0; i<banyakData(); i++) {
        if (dataPenduduk[i].statusKawin == 0) { total[0]++; }
        else if (dataPenduduk[i].statusKawin == 1) { total[1]++; }
        else if (dataPenduduk[i].statusKawin == 2) { total[2]++; }
        else if (dataPenduduk[i].statusKawin == 3) { total[3]++; }
    }

    // tampilkan diagram
    for (int i=0; i<9; i++) {

        if (i%2 == 0) {
            diagram(0); cout << endl;

        } else {
            diagram((total[i/2]*30)/totalData, warna[i/2]);
            cout << " " << (total[i/2]*100)/totalData << "%" << endl;
        }
    }

    // keterangan warna
    int jarak = 0;
    for (int i=0; i<4; i++) {

        gotoxy(55, 8+jarak);
        color(warna[i]); cout << char(254) << " ";
        color(RESET); cout << STATUSKAWIN[i];

        jarak += 2;
    }
}

```

```

}

//pengisian formulir
void tampilkanOpsi(string opsi) {
    if (opsi == "Kecamatan") {
        cout << "\n\t Kecamatan
        << "\n\t [1] Palaran      [5] Samarinda Utara  [9] Samarinda Kota "
        << "\n\t [2] Samarinda Seberang [6] Sungai Kunjang  [10] Loa Janan Ilir "
        << "\n\t [3] Samarinda Ulu     [7] Sambutan          "
        << "\n\t [4] Samarinda Ilir    [8] Sungai Pinang          "
        << "\n\t : ";
    } else if (opsi == "Palaran") { // 1
        cout << "\n\t Kelurahan
        << "\n\t [1] Rawa Makmur    [4] Simpang Pasir          "
        << "\n\t [2] Handil Bakti   [5] Bantuan           "
        << "\n\t [3] Bukuan          "
        << "\n\t : ";
    } else if (opsi == "Samarinda Seberang") { // 2
        cout << "\n\t Kelurahan
        << "\n\t [1] Sungai Keledang  [4] Mangkupalas          "
        << "\n\t [2] Baqa            [5] Gunung Panjang          "
        << "\n\t [3] Mesjid           "
        << "\n\t : ";
    } else if (opsi == "Samarinda Ulu") { // 3
        cout << "\n\t Kelurahan
        << "\n\t [1] Teluk Lerong Ilir  [4] Sidodadi    [7] Gunung Kelua  "
        << "\n\t [2] Jawa             [5] Air Hitam    [8] Bukit Pinang "
        << "\n\t [3] Air Putih       [6] Dadi Mulya          "
        << "\n\t : ";
    } else if (opsi == "Samarinda Ilir") { // 4
        cout << "\n\t Kelurahan
        << "\n\t [1] Selili          [4] Sidodamai          "
        << "\n\t [2] Sungai Dama     [5] Pelita           "
        << "\n\t [3] Sidomulyo         "
        << "\n\t : ";
    } else if (opsi == "Samarinda Utara") { // 5
}

```

```

cout << "\n\t Kelurahan"
    << "\n\t [1] Sempaja Selatan [4] Sempaja Utara [7] Sempaja Timur "
    << "\n\t [2] Lempake [5] Tanah Merah [8] Budaya Pampang "
    << "\n\t [3] Sungai Siring [6] Sempaja Barat "
    << "\n\t : ";
} else if (opsi == "Sungai Kunjang") { // 6
    cout << "\n\t Kelurahan"
        << "\n\t [1] Loa Bakung [4] Lok Bahu [6] Karang Asam "
        << "\n\t [2] Loa Buah [5] Telok Lerong Ilir "
        << "\n\t [3] Karang Asam Ulu Ulu [7] Karang Anyar "
        << "\n\t : ";
} else if (opsi == "Sambutan") { // 7
    cout << "\n\t Kelurahan"
        << "\n\t [1] Sungai Kapih [4] Sindang Sari "
        << "\n\t [2] Sambutan [5] Pulau Atas "
        << "\n\t [3] Makroman "
        << "\n\t : ";
} else if (opsi == "Sungai Pinang") { // 8
    cout << "\n\t Kelurahan"
        << "\n\t [1] Temindung Permai [3] Gunung Lingai "
        << "\n\t [2] Sungai Pinang [4] Mugirejo "
        << "\n\t Dalam [5] Bandara "
        << "\n\t : ";
} else if (opsi == "Samarinda Kota") { // 9
    cout << "\n\t Kelurahan"
        << "\n\t [1] Karang Mumus [4] Bugis "
        << "\n\t [2] Pelabuhan [5] Sungai Pinang "
        << "\n\t [3] Pasar Pagi Luar "
        << "\n\t : ";
} else if (opsi == "Loa Janan Ilir") { // 10
    cout << "\n\t Kelurahan"
        << "\n\t [1] Simpang Tiga [4] Harapan Baru "
        << "\n\t [2] Tani Aman [5] Rapak Dalam "
        << "\n\t [3] Sengkotek "
        << "\n\t : ";
} else if (opsi == "Jenis Kelamin") {
    cout << "\n\t Jenis Kelamin \n"
        << "\t [1] Laki-laki [2] Perempuan "
            "\n";
}

```

```

<< "\t : \n";
} else if (opsi == "Agama") {
    cout << "\n\tAgama \n"
        << "\t [1] Islam    [3] Katolik    [5] Buddha      \n"
        << "\t [2] Kristen   [4] Hindu     [6] Konghuchu    \n"
        << "\t : \n";
} else if (opsi == "Golongan Darah") {
    cout << "\n\tGolongan Darah           \n"
        << "\t [1] A      [2] B      [3] AB     [4] O           \n"
        << "\t : \n";
} else if (opsi == "Status Perkawinan") {
    cout << "\n\tStatus Perkawinan          \n"
        << "\t [1] Belum kawin  [2] Kawin  [3] Cerai hidup  [4] Cerai mati \n"
        << "\t : \n";
} else if (opsi == "Status Hidup") {
}
}

// tabel
void tabel(short int nama, short int nik, short int tanggal) {
    gotoxy(0, 6);
    cout << "\t+-----+-----+-----+-----+" << endl;

    cout << "\t| No ";
    cout << "| Nama Lengkap "; color(nama); cout << char(31) << " ";
    color(RESET);
    cout << "| NIK "; color(nik); cout << char(31) << " "; color(RESET);
    cout << "| Diperbarui Pada "; color(tanggal); cout << char(31) << "
"; color(RESET);
    cout << "|" << endl;

    cout << "\t+-----+-----+-----+-----+" << endl;

    for (int i=0; i<10; i++) {
        gotoxy(8, 9+i); cout << "|";
        gotoxy(15, 9+i); cout << "|";
    }
}

```

```

        gotoxy(38, 9+i); cout << "|";
        gotoxy(57, 9+i); cout << "|";
        gotoxy(77, 9+i); cout << "|";
    }

    cout << "\n\t-----+-----+-----+" << endl;
}

/* ----- FILE TXT ----- */

// ## Modul 7 - File
void importFromTxt() {
    ifstream input;
    input.open("data.txt");

    // deklarasi var
    Penduduk penduduk;
    string temp;
    int index = 0;

    // selama belum di akhir file, masukkan ke struct array
    while (!input.eof()) {
        getline(input, temp, '|');

        if (temp == "") { break; }           // jika masih kosong, berhenti
        else { penduduk.tanggalPembaruan = temp; } // tanggal diperbarui

        getline(input, penduduk.namaLengkap, '|'); // nama lengkap

        getline(input, temp, '|');
        strcpy(penduduk.nik, temp.c_str()); // nik

        getline(input, penduduk.password, '|'); // password
        getline(input, penduduk.ttl, '|'); // tempat tanggal lahir
        getline(input, penduduk.alamat.jalan, '|'); // alamat.jalan

        getline(input, temp, '|');
        penduduk.alamat.kecamatan = stoi(temp); // alamat.kecamatan
    }
}

```

```

getline(input, temp, '|');
penduduk.alamat.kelurahan = stoi(temp); // alamat.kelurahan

getline(input, penduduk.telepon, '|'); // no telepon

getline(input, temp, '|');
penduduk.usia = stoi(temp); // usia

getline(input, temp, '|');
penduduk.gender = stoi(temp); // gender

getline(input, temp, '|');
penduduk.agama = stoi(temp); // agama

getline(input, temp, '|');
penduduk.golDar = stoi(temp); // golongan darah

getline(input, temp, '\n');
penduduk.statusKawin = stoi(temp); // status perkawinan

getline(input, temp, '\n');
penduduk.statusHidup = stoi(temp); // status hidup

dataPenduduk[index] = penduduk; // masukkan array
index++;

}

input.close();
}

void appendToTxt(Penduduk penduduk) {
    ofstream file;
    file.open("data.txt", ios::app);

    // tambahkan ke file eksternal
    file << penduduk.tanggalPembaruan << "|"

```

```

<< penduduk.namaLengkap    << "|"
<< penduduk.nik          << "|"
<< penduduk.password     << "|"
<< penduduk.ttl          << "|"
<< penduduk.alamat.jalan << "|"
<< penduduk.alamat.kecamatan << "|"
<< penduduk.alamat.kelurahan << "|"
<< penduduk.telepon      << "|"
<< penduduk.usia         << "|"
<< penduduk.gender        << "|"
<< penduduk.agama         << "|"
<< penduduk.golDar        << "|"
<< penduduk.statusKawin   << "|"
<< penduduk.statusHidup   << "\n";

file.close();

importFromTxt();
}

void deleteFromTxt(Penduduk penduduk) {
    int length = banyakData(),
    index = searchNIK(penduduk.nik);

    ofstream output;
    output.open("data.txt");

    // tulis ulang seluruh data ke file eksternal
    for (int i=0; i<STORAGE; i++) {

        // jika adalah data yg ingin dihapus, jangan ditulis
        if (string(dataPenduduk[i].nik) == string(penduduk.nik)) {
            continue;

        // jika sampai di data terakhir
        } else if (string(dataPenduduk[i].nik) == "") {
            break;
        }
    }
}

```

```

// tambahkan ke file eksternal
appendToTxt(dataPenduduk[i]);
}

output.close();

// ambil ulang data dari file eksternal
importFromTxt();
}

void updateToTxt(Penduduk penduduk) {
penduduk.tanggalPembaruan = hariIni(); // tgl data diri diubah

deleteFromTxt(penduduk); // hapus data dari file eksternal
appendToTxt(penduduk); // tambahkan data baru ke file eksternal

}

/* ----- SORT & SEARCH ----- */

// ## Modul 5 - Sorting
// sort berdasarkan nik - insertion sort
void sortNIK(string mode) {
Penduduk key;
int i, j,
length = banyakData();

for (i = 1; i < length; i++) {

key = dataPenduduk[i];
j = i-1;

// ascending
if (mode == "ASCENDING") {
while (j >= 0 && string(dataPenduduk[j].nik).compare(key.nik) > 0){
dataPenduduk[j+1] = dataPenduduk[j];
j = j-1;
}
}
}
}

```

```

    }

    // descending
} else {
    while (j >= 0 && string(dataPenduduk[j].nik).compare(key.nik) < 0){
        dataPenduduk[j+1] = dataPenduduk[j];
        j = j-1;
    }
}

dataPenduduk[j+1] = key;
}
}

// ## Modul 2 - Array dan Pointer
// tukar elemen penduduk
void swap(Penduduk *elemen1, Penduduk *elemen2) {

    Penduduk temp = *elemen1;
    *elemen1 = *elemen2;
    *elemen2 = temp;
}

// sorting berdasarkan nama lengkap - quick sort
int partition(int low, int high, string mode) {
    string pivot = dataPenduduk[high].namaLengkap;
    int i = (low - 1);
    for (int j = low; j <= high- 1; j++) {

        // ascending
        if (
            dataPenduduk[j].namaLengkap.compare(pivot) <= 0
            && mode == "ASCENDING"
        ) {
            i++;
            swap(&dataPenduduk[i], &dataPenduduk[j]);
        }
    }
}

```

```

// descending
if (
    dataPenduduk[j].namaLengkap.compare(pivot) >= 0
    && mode == "DESCENDING"
) {
    i++;
    swap(&dataPenduduk[i], &dataPenduduk[j]);
}
}
swap(&dataPenduduk[i + 1], &dataPenduduk[high]);
return (i + 1);
}

// quick sort
void sortNamaLengkap(int low, int high, string mode) {

    // ## Modul 8 - Rekursif

    if (low < high) {
        int pi = partition(low, high, mode);
        sortNamaLengkap(low, pi - 1, mode);
        sortNamaLengkap(pi + 1, high, mode);
    }
}

// sorting berdasarkan tanggal - shell sort
void sortTanggal(string mode) {
    Penduduk temp;
    int n = banyakData();

    for (int gap=n/2; gap > 0; gap /= 2) {
        for (int i = gap; i < n; i++) {
            for (int j=i-gap; j >= 0; j-=gap) {

                if (mode == "ASCENDING") {
                    if (
                        ambilTanggal(dataPenduduk[j + gap].tanggalPembaruan) >=
                        ambilTanggal(dataPenduduk[j].tanggalPembaruan) &&

```

```

        ambilBulan(dataPenduduk[j + gap].tanggalPembaruan) >=
ambilBulan(dataPenduduk[j].tanggalPembaruan) &&
        ambilTahun(dataPenduduk[j + gap].tanggalPembaruan) >=
ambilTahun(dataPenduduk[j].tanggalPembaruan)
    ) {

        break;

} else {

    temp = dataPenduduk[j];
    dataPenduduk[j] = dataPenduduk[j + gap];
    dataPenduduk[j + gap] = temp;

}

} else {

    if (
        ambilTanggal(dataPenduduk[j + gap].tanggalPembaruan) <=
ambilTanggal(dataPenduduk[j].tanggalPembaruan) &&
        ambilBulan(dataPenduduk[j + gap].tanggalPembaruan) <=
ambilBulan(dataPenduduk[j].tanggalPembaruan) &&
        ambilTahun(dataPenduduk[j + gap].tanggalPembaruan) <=
ambilTahun(dataPenduduk[j].tanggalPembaruan)
    )
        break;

    else {
        temp = dataPenduduk[j + gap];
        dataPenduduk[j + gap] = dataPenduduk[j];
        dataPenduduk[j] = temp;
    }

} // end if
} // end for
} // end for
} // end for

```

```

}

// ## Modul 6 - Searching
// cari index NIK - binary search
int searchNIK(string nik) {
    int beg = 0, end = banyakData()-1, mid;
    sortNIK("ASCENDING");

    while( beg <= end ){

        mid = (end + beg) / 2;

        if (dataPenduduk[mid].nik == nik) {
            return mid;

        } else {

            if(nik.compare(dataPenduduk[mid].nik) > 0){
                beg = mid + 1;

            } else {
                end = mid - 1;

            } // end if
        } // end if
    } // end while

    return -1;
}

// cari index elemen - sequential search
int indexElemen(short int array[], int cari, int length) {
    for (int i=0; i<length; i++) {
        if (array[i] == cari) {
            return i;
        }
    }
}

```

```

        return -1;
    }

/* ----- TAMBAHAN ----- */

void color(unsigned short warna) {
    HANDLE hCon = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hCon, warna);
}

void gotoxy(int x, int y) {
    COORD coord;
    coord.X = x;
    coord.Y = y;
    SetConsoleCursorPosition(GetStdHandle(STD_OUTPUT_HANDLE), coord);
}

tm *timeNow() {
    time_t now = time(0);
    tm *ltm = localtime(&now);

    return ltm;
}

string hariIni() {
    string tgl = to_string(timeNow()->tm_mday);
    string bulan = to_string(timeNow()->tm_mon + 1);
    string tahun = to_string(timeNow()->tm_year + 1900);

    return tgl + "/" + bulan + "/" + tahun;
}

void clearCin() {
    cin.clear();
    cin.ignore();
    fflush(stdin);
}

```

```

bool logInBerhasil(char nik[], string password) {
    for (int i=0; i<banyakData(); i++) {
        if (
            string(dataPenduduk[i].nik) == string(nik)
            && dataPenduduk[i].password == password
        ){
            return true;
        }
    }

    return false;
}

int banyakData(){
    for (int i=0; i<STORAGE; i++) {
        if (string(dataPenduduk[i].nik) == "") {
            return i;
        }
    }
    return STORAGE;
}

void karakter(int ascii, int jumlah, int warna) {
    color(warna);
    for (int i=0; i<jumlah; i++) {
        cout << char(ascii);
    }
    color(RESET);
}

void diagram(int jumlah, unsigned int warna) {
    cout << "\t\t" << char(221); color(warna);

    for (int i=0; i<jumlah; i++) {
        cout << char(219);
    }
}

```

```

        color(7);
    }

bool formulirTerisi(Penduduk penduduk) {
    if (
        penduduk.ttl      == "" ||
        penduduk.alamat.jalan  == "" ||
        penduduk.alamat.kecamatan == -1 ||
        penduduk.alamat.kelurahan == -1 ||
        penduduk.telepon    == "" ||
        penduduk.gender     == -1 ||
        penduduk.agama      == -1 ||
        penduduk.golDar     == -1 ||
        penduduk.statusKawin == -1
    ) {
        return false;
    }

    return true;
}

void hapusTulisan(int panjang) {
    color(0);
    for (int i=0; i<panjang; i++) {
        cout << char(219);
    }
    color(RESET);
}

int banyakDataTerisi() {
    // hitung data terverifikasi
    int total = 0;
    for (int i=0; i<banyakData(); i++) {
        if (formulirTerisi(dataPenduduk[i])) {
            total += 1;
        }
    }
}

```

```

        return total;
    }

int ambilTanggal(string str) {
    string temp;

    for (int i=0; i<str.length(); i++) {
        if (str[i] == '/') {
            break;
        }

        temp += str[i];
    }

    return stoi(temp);
}

int ambilBulan(string str) {
    string temp;

    int iAwal = str.find("/") + 1;

    for (int i=iAwal; i<str.length(); i++) {
        if (str[i] == '/') {
            break;
        }

        temp += str[i];
    }

    return stoi(temp);
}

int ambilTahun(string str) {
    string temp;

    int iBulan = str.find("/") + 1;

```

```

int iTahun = str.find("/", iBulan) + 1;

for (int i=iTahun; i<str.length(); i++) {
    temp += str[i];
}

return stoi(temp);
}

/* ----- CEK ISIAN ----- */

bool isAngka(string str) {
    int length = str.length();

    int isInteger = 0;
    for (int i=0; i<length; i++) {
        int cek = (int)str[i] - 48;

        if (0 <= cek && cek <= 9) {
            isInteger += 1;
        }
    }

    if (isInteger == length && !isSpace(str)) {
        return true;
    }

    return false;
}

bool isSpace(string str) {
    if (str == "\t" || str == "\n" || str == "") {
        return true;
    }

    int length = str.length(),
        banyakSpasi = 0;

```

```

for (int i=0; i<length; i++) {
    if (str[i] == ' ') {
        banyakSpasi++;
    }
}

if (length == banyakSpasi) {
    return true;
}

return false;
}

bool isTahun(int tahun) {
    int tahunIni = timeNow()->tm_year + 1900;

    if (tahun >= 1920 && tahun <= tahunIni) {
        return true;
    }

    return false;
}

bool isTanggal(int tanggal) {
    if (tanggal >= 1 && tanggal <= 31) {
        return true;
    }

    return false;
}

bool isBulan(int bulan) {
    if (bulan >= 1 && bulan <= 12) {
        return true;
    }

    return false;
}

```

```
bool isNIK(char nik[]) {
    string nikStr = string(nik);
    int length = nikStr.length();

    if (!isSpace(nikStr) && isAngka(nikStr) && length == 16) {
        return true;
    }

    return false;
}

bool isTelp(string telp) {
    int length = telp.length();

    if (!isSpace(telp) && isAngka(telp) && (length >= 10 && length <= 12)) {
        return true;
    }

    return false;
}
```

BAB 4

KESIMPULAN DAN SARAN

4.1 Kesimpulan

Dari hasil pembahasan tentang Program PENDATAAN PENDUDUK DI WILAYAH SAMARINDA ini, dapat kami ambil beberapa kesimpulan yang ada, seperti diantaranya :

- Dengan adanya program ini, akan memudahkan pihak Pemerintah dalam menyajikan informasi dan mengelola data kependudukan dengan efisien
- Program ini juga memudahkan Penduduk dalam pengisian data ketika berkaitan dengan kependudukan dan catatan sipil. Dengan hanya melakukan login dengan akun mereka masing-masing, para penduduk sudah dapat melakukan pengisian secara mandiri.
- Pada Program ini, menu pemerintah didukung dengan fitur CRUD dengan *update* yang hanya mempunyai akses untuk mengubah status hidup yang akan memudahkan kerja dari pihak pemerintah di dalamnya. Dimana, jika pihak pemerintah ingin melakukan suatu perubahan terhadap program ini, dapat dilakukannya secara mudah dan leluasa serta menghindari adanya aktivitas ilegal seperti manipulasi data penduduk.

4.2 Saran

Kami selaku pembuat dan penyusun program ini, menyadari bahwa hasil dari program kami ini masih sangat jauh dari kata sempurna. Oleh karena itu, kami sangat mengharapkan adanya kritik dan saran dari pihak audiens terhadap hasil dari program kami ini.

Kami juga menyadari bahwa program ini masih kurang dalam menerapkan fungsi-fungsi yang ada pada Bahasa pemrograman *C++*. Kami sangat ingin menambahkan fungsi-fungsi lainnya pada program ini, namun kami menyadari adanya kendala terhadap batasan waktu dan pengetahuan kami yang masih belum seberapa.

Kami akan menambahkan fungsi-fungsi tersebut suatu saat nanti, sehingga program yang kami buat ini akan memiliki lebih banyak fungsi secara luas yang dapat dimanfaatkan dalam kehidupan perkantoran sehari-hari.

DAFTAR PUSTAKA

- Mujahid, F. (2022). Fungsi Manipulator C++. Retrieved 3 June 2022, from <http://rhatavarium.blogspot.com/2012/04/fungsi-manipulator-c.html>
- (2015). C/C++ Common Mistake: gotoxy(). Retrieved 3 June 2022, from <https://learntocodes.wordpress.com/2015/12/11/cc-common-mistake-gotoxy/>
- Abdurrahman, M. (2015). Fungsi . Retrieved 3 June 2022, from https://jagocoding.com/blog/1162/Fungsi_system_pada_header_windows_h
- (2016). Penggunaan fungsi sleep() sebagai timer. Retrieved 3 June 2022, from <https://soc.s.binus.ac.id/2016/12/28/penggunaan-fungsi-sleep-sebagai-timer/>
- (2022). Pada bahasa pemrograman C++, dapat dibuat program dengan beberapa sub-program sesuai dengan keinginan dengan menggunakan fungsi Retrieved 3 June 2022, from <https://adoc.pub/pada-bahasa-pemrograman-c-dapat-dibuat-program-dengan-beberapa.html>
- Zhan. (2018). ctime (time.h) - Belajar C++. Retrieved 3 June 2022, from <https://www.belajarcpp.com/referensi/ctime/>
- Unknown. (2022). Program Waktu dengan C++. Retrieved 3 June 2022, from <http://www.nblognlife.com/2013/11/program-waktu-dengan-c.html>

LAMPIRAN

Tanggal Konsultasi : 11 - Mei - 2022

Uraian / Pembahasan :

- Pergantian Menu "delete" ke Mode Pemerintah.
- Penambahan opsi searching dan sorting ke dalam Mode Pemerintah, yang mana opsi sorting Meliputi Ascending dan Descending
- Hapus menu "update" di dalam Mode Pemerintah
- Ganti nama data "status" menjadi "status Kawin"
- Tambahkan "status hidup" yang bersifat opsi hidup dan mati penduduk ke dalam Mode Pemerintah.
- Berikan akses kepada pemerintah untuk Mengubah "status hidup" penduduk
- Masukkan inputan secara satu per satu ke dalam flowchart.

Asisten Lab

Nama : Fayza Virdana
Addiza

Ketua Kelompok

Nama : Aziizah Oki Shofrina

Tanggal Konsultasi : 31 - Mei - 2022

Uraian / Pembahasan :

- Kelompokkan void pemerintah dengan pemerintah dan void penduduk dengan penduduk untuk mempermudah dalam membaca ataupun memperbaiki kesalahan kodingan.
- Ganti kata "penulis" di laporan dengan kata "penyusun"
- Jadikan lembar flowchart menjadi satu halaman full agar tidak terpotong.
- Tambahkan comment di kodingan untuk menandakan sebuah perintah secara garis besar dan sebagai penanda materi modul yang digunakan.
- Perbaiki format daftar pustaka dan seuratkannya dengan ketentuan pada umumnya.

Asisten Lab



Nama : Fayza Virdana
Addiza

Ketua Kelompok



Nama : Aziizah Oki Shofrina