

**UJIAN AKHIR EL2008 PEMECAHAN MASALAH DENGAN C
DATABASE KEPENDUDUKAN**

Oleh:

Kevin Naoko (13218046)

Dosen:

Arif Sasongko S.T., M.T., Ph.D.

Dr. Reza Darmakusuma S.T., M.T.

Muhammad Ogin Hasanuddin S.T.,M.T.



**PROGRAM STUDI TEKNIK ELEKTRO
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2020**

1. SPESIFIKASI

Pada tugas kali ini, akan dibuat suatu *database* kependudukan yang memuat parameter berikut ini:

1. Nomor Induk Kependudukan (NIK)
2. Nama Lengkap
3. Tempat Lahir
4. Tanggal Lahir
5. Umur
6. Jenis Kelamin
7. Golongan Darah
8. Status Perkawinan
9. Pekerjaan

Khusus untuk parameter umur, program diharapkan dapat menghitung umur dari tiap penduduk secara *real-time*. Program juga diharapkan dapat melakukan beberapa tindakan dasar seperti layaknya *database* pada umumnya, yakni pembuatan data, penghapusan data, perubahan data, mengimport *database* dari file eksternal, pencarian data, serta pengurutan data. Pada realisasinya, ditambahkan pula oleh pengguna fitur penyimpanan kembali *database* yang terdapat pada program ke file eksternal, serta penampilan database dengan bentuk tabel.

2. ANALISIS PERMASALAHAN

Untuk mempermudah pengolahan data secara keseluruhan, akan dibuat dua tipe data baru, yaitu ‘data’ dan ‘date’. Akan digunakan pula satu variabel global “filename” yang akan mempermudah proses pemilihan overwrite file. Rincian dari kedua tipe data ini dapat dilihat di gambar 2-1.

Setelah itu, untuk mempermudah perhitungan usia, digunakan library time.h yang dapat meng-fetch tanggal dari komputer secara *realtime*.

```
#define MAX 30
#define ENTRY_MAX 1024

typedef struct database{
    char nik[MAX];
    char nama[MAX];
    char tempat_lahir[MAX];
    char tanggal_lahir[MAX];
    int umur;
    char jen_kel[MAX];
    char gol_dar[MAX];
    char status[MAX];
    char pekerjaan[MAX];
} data;

typedef struct date_sys{
    int day;
    int month;
    int year;
} date;

char filename[128];
```

Gambar 2-1. Struktur Data yang Digunakan

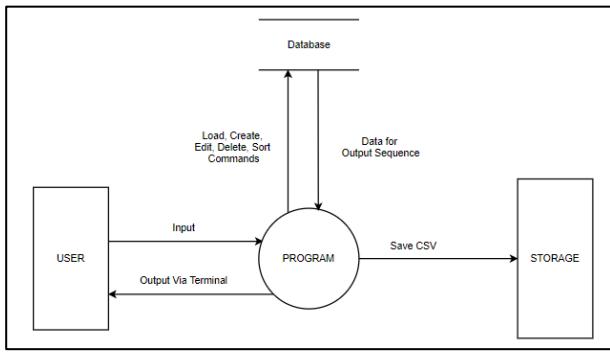
Sebelum dimulai proses implementasi permasalahan ke bahasa C, perlu diberi beberapa asumsi yang mendasari pembuatan program ini. Asumsi yang akan disajikan telah melewati proses pertimbangan yang matang, mengingat ketersediaan waktu yang sangat minim untuk menyelesaikan project ini.

1. Seluruh data yang pengguna masukkan ke dalam *database* adalah valid. Namun tetap tersedia validator untuk proses navigasi pada menu program
2. *Database* akan dirancang dengan model struktur data static array, yang memiliki kapasitas maksimal 1024. Namun apabila diinginkan jumlah penampungan yang lebih banyak, pengguna dapat mengubah define “ENTRY_MAX” menyesuaikan dengan jumlah yang diperlukan
3. Semua string pada tipe data ‘data’ memiliki panjang maksimal 30, yang telah didefinisikan dengan define “MAX”. Namun pada proses display, terdapat limitasi jumlah karakter maksimal yang dapat ditampilkan. Bila limitasi ini dilanggar, hanya akan mengakibatkan penurunan kualitas estetika tabel. Diputuskan bahwa jumlah karakter nama dan tempat lahir maksimal 20 karakter, dan pekerjaan maksimal 15 karakter.

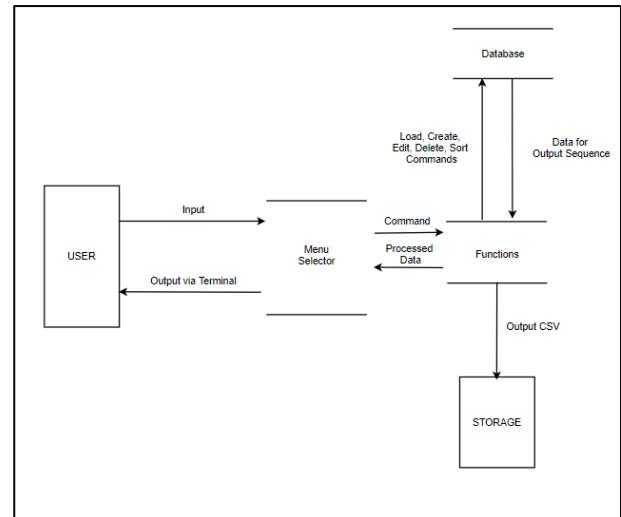
3. FLOWCHART RANCANGAN ALGORITMA

Pada bagian pemaparan flowchart tidak dilakukan pembagian dua kolom untuk mempermudah pembaca dalam melihat alur pemrograman pada setiap fungsi. Bagian ini akan dibagi menjadi 3 bagian, dimana bagian pertama memaparkan Data Flow Diagram dari program, bagian kedua memaparkan flowchart dari fungsi utama, bagian ketiga memaparkan flowchart dari fungsi komplementer, dan bagian keempat memaparkan flowchart dari fungsi processing.

3.1. DATA FLOW DIAGRAM



Gambar 3-1. Data Flow Diagram L0



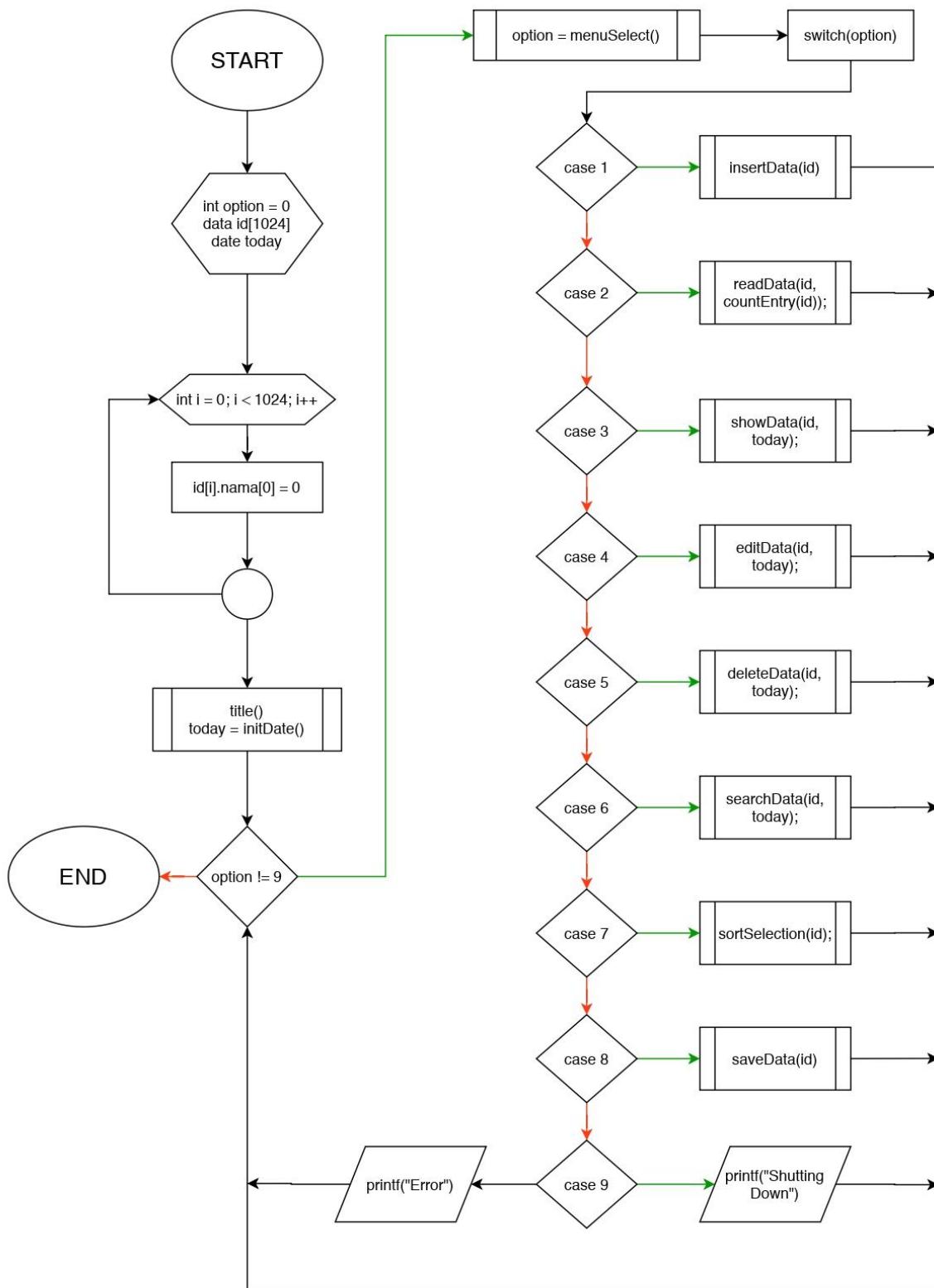
Gambar 3-2 Data Flow Diagram L1

Secara garis besar, user akan memberi input sesuai permintaan program, dan program akan menjalankan command sesuai masukkan pengguna, dan pengguna akan menerima keluaran dalam terminal. Struktur data 'database' diperlukan untuk melakukan operasi yang diperintahkan oleh pengguna

Secara lebih detil, user akan memasukkan pilihan yang diinginkan ke selektor pada program. Selektor ini kemudian akan memilih fungsi apa yang dijalankan. Fungsi ini dapat berinteraksi langsung dengan database untuk keperluan *commands* yang ada, dan dapat pula mengeluarkan output file ke storage

3.2. FUNGSI UTAMA

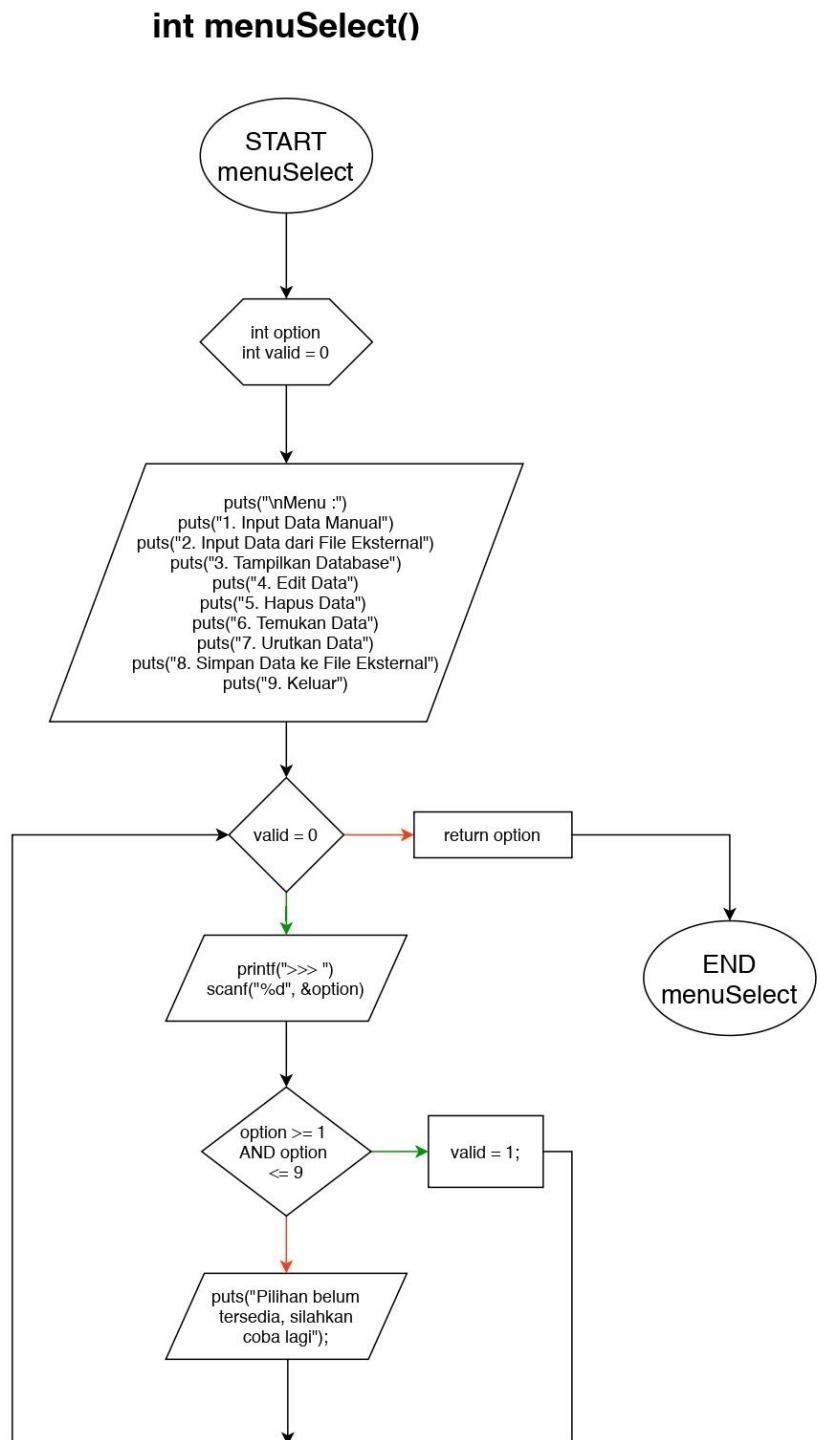
Main Function



Gambar 3-3. Flowchart dari Fungsi Utama

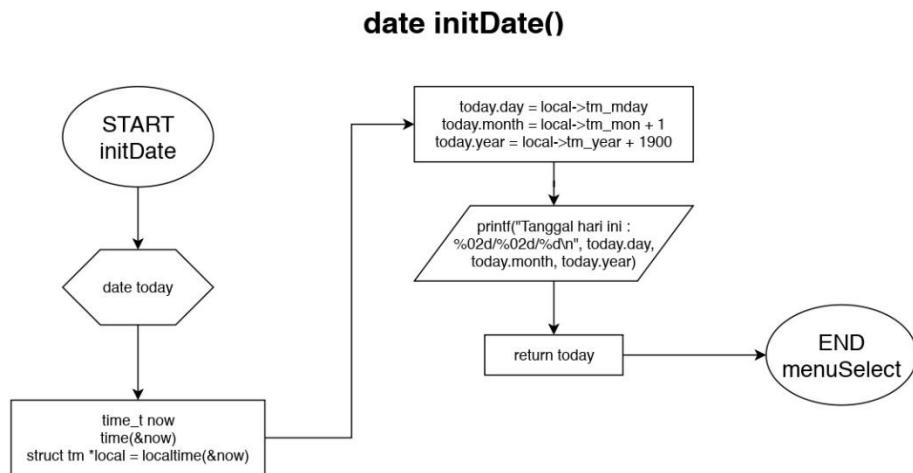
3.3. FUNGSI PENUNJANG

3.3.1. MENUSELECT()



Gambar 3-4. Flowchart dari Fungsi menuSelect

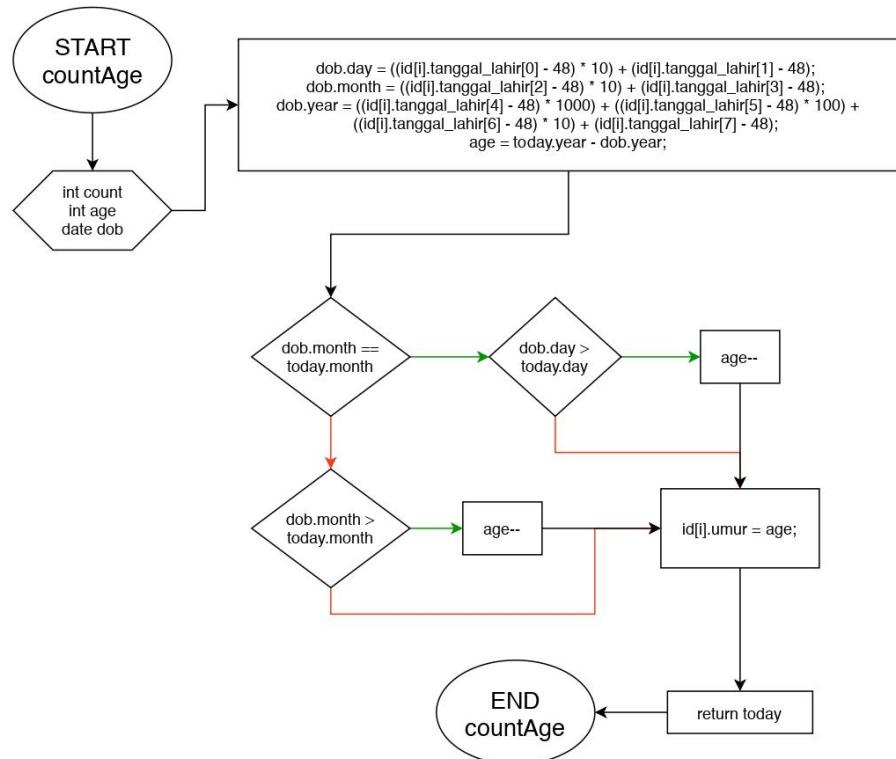
3.3.2. INITDATE ()



Gambar 3-5. Flowchart dari Fungsi initDate

3.3.3. COUNTAGE ()

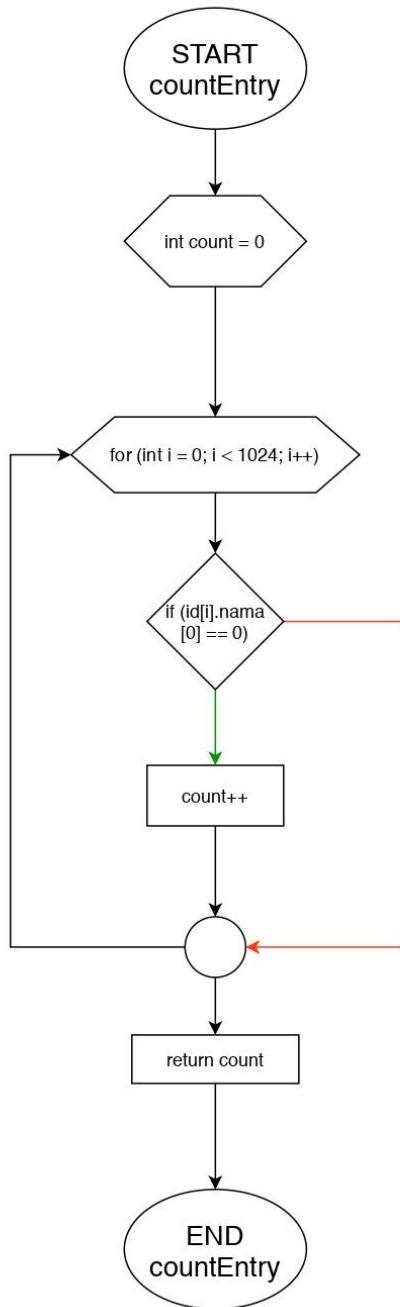
int countAge(data id[], date today, int i)



Gambar 3-6. Flowchart dari Fungsi countAge

3.3.4. COUNTENTRY ()

```
int countEntry(data id[], date today, int i)
```

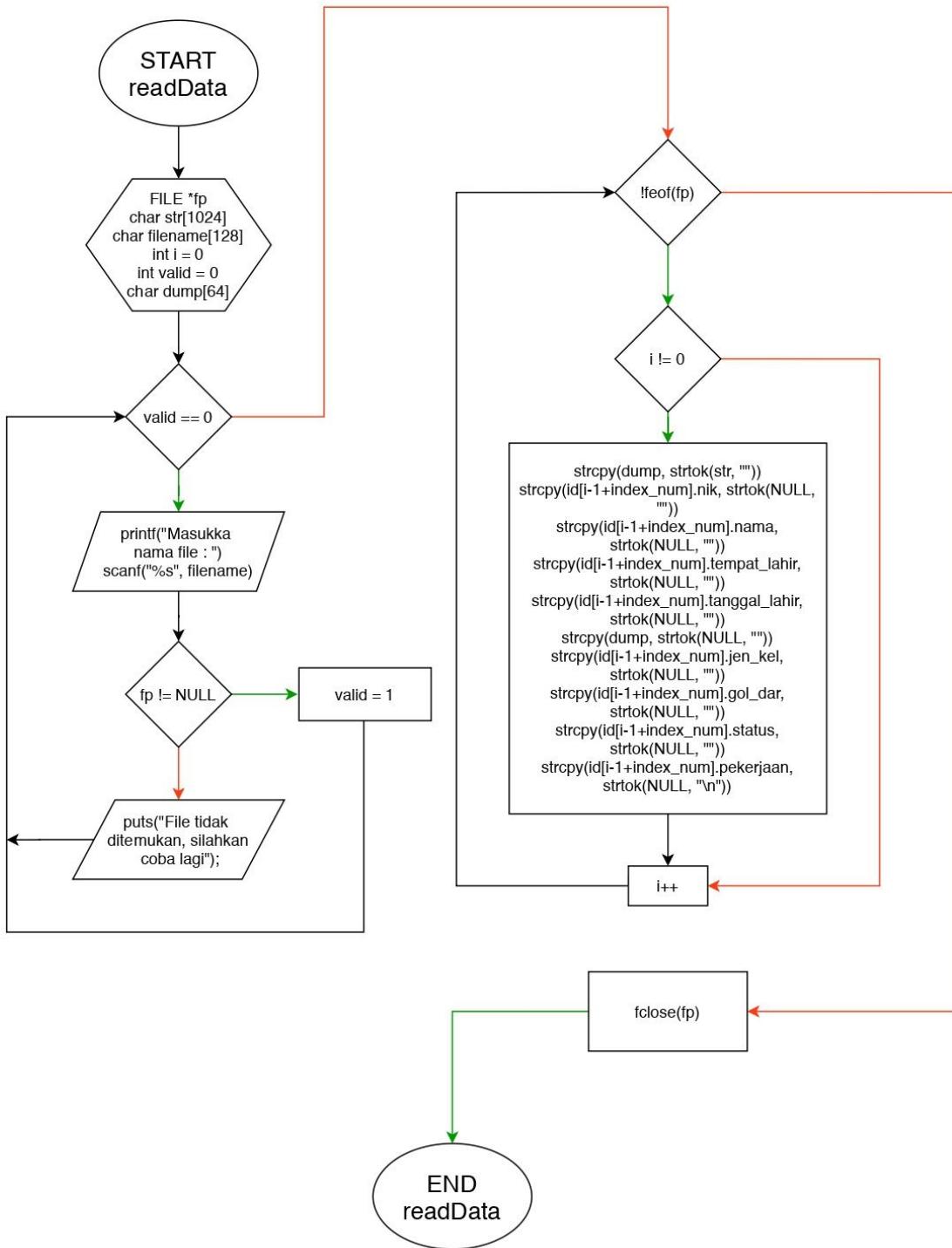


Gambar 3-7. Flowchart dari Fungsi countEntry

3.4. FUNGSI PROCESSING

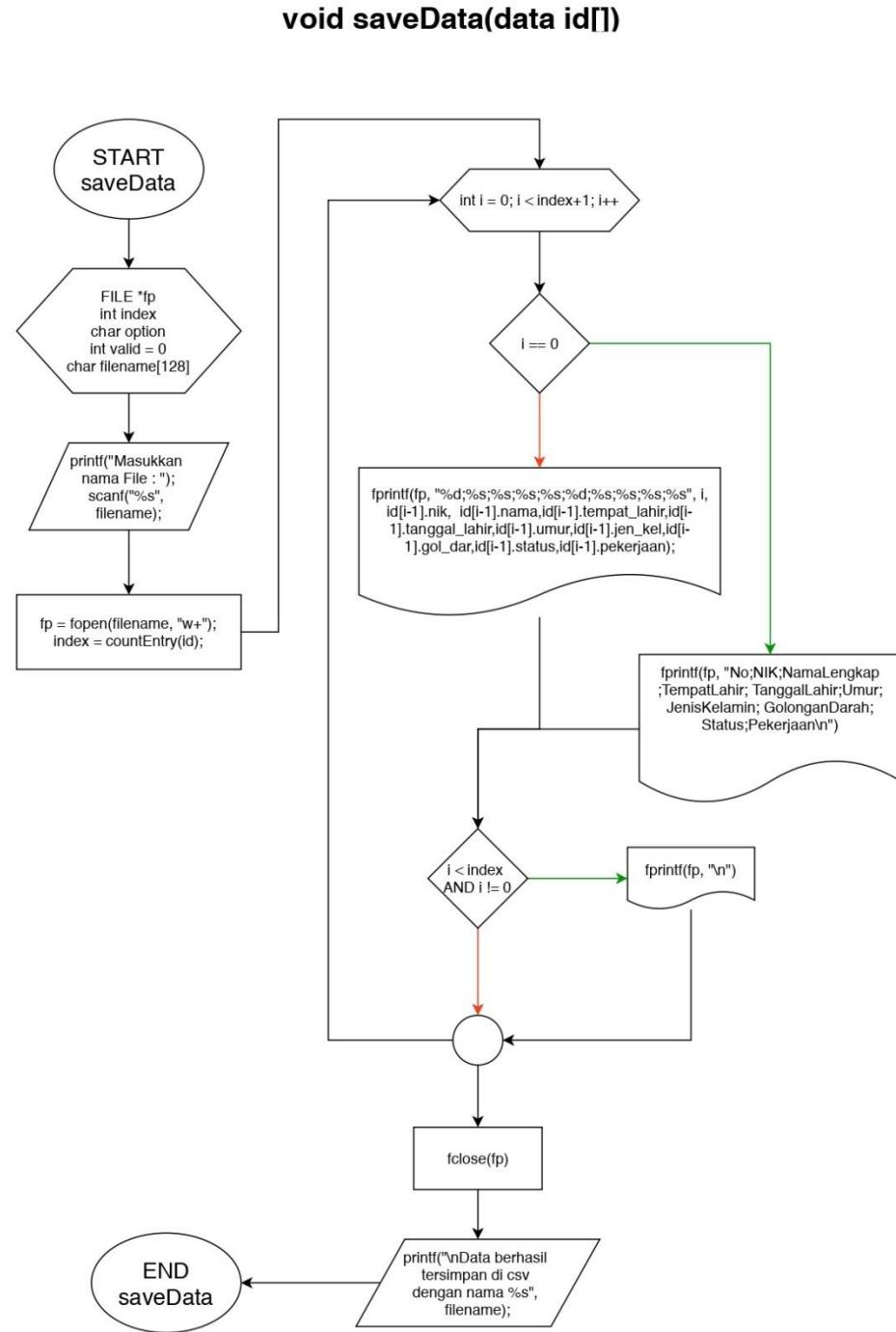
3.4.1. FUNGSI READDATA

void readData(data id[], int index_num)



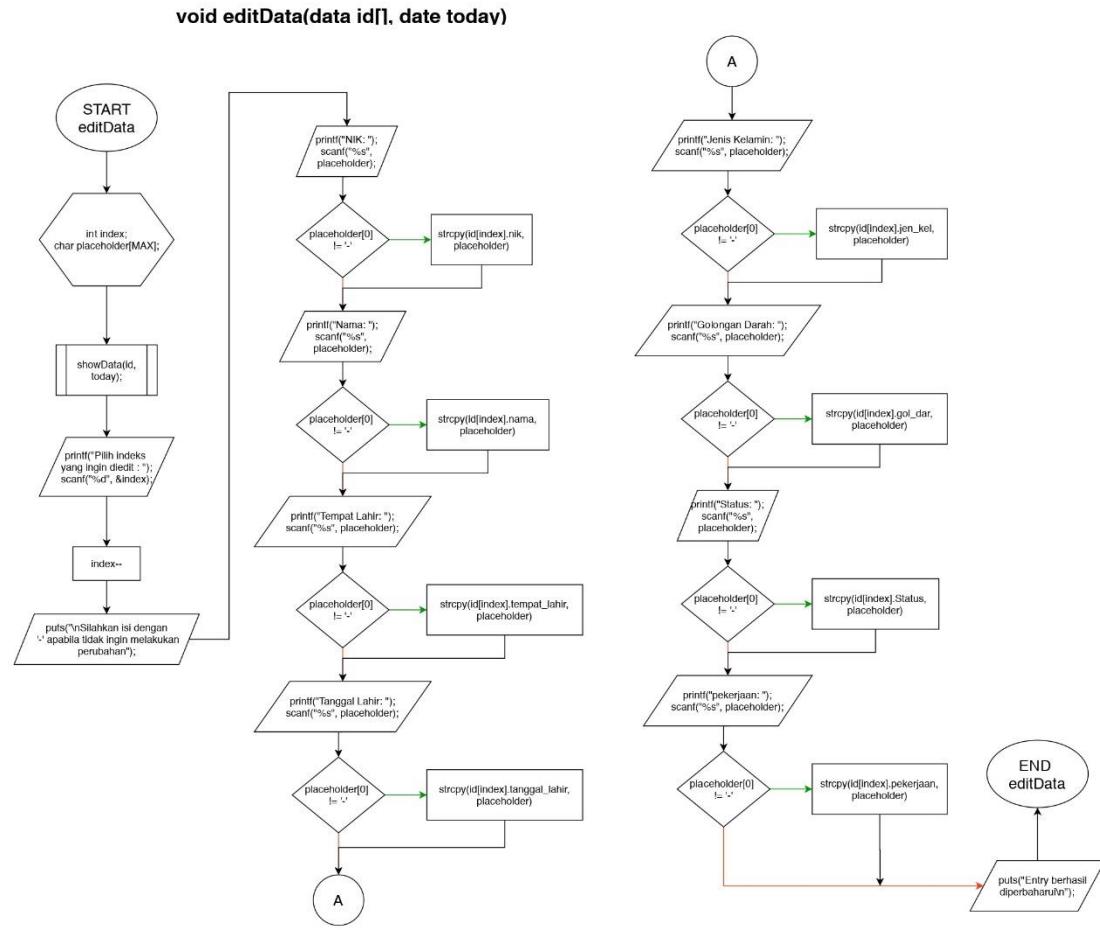
Gambar 3-8. Flowchart dari Fungsi countEntry

3.4.2. FUNGSI SAVEDATA



Gambar 3-10. Flowchart dari Fungsi saveData

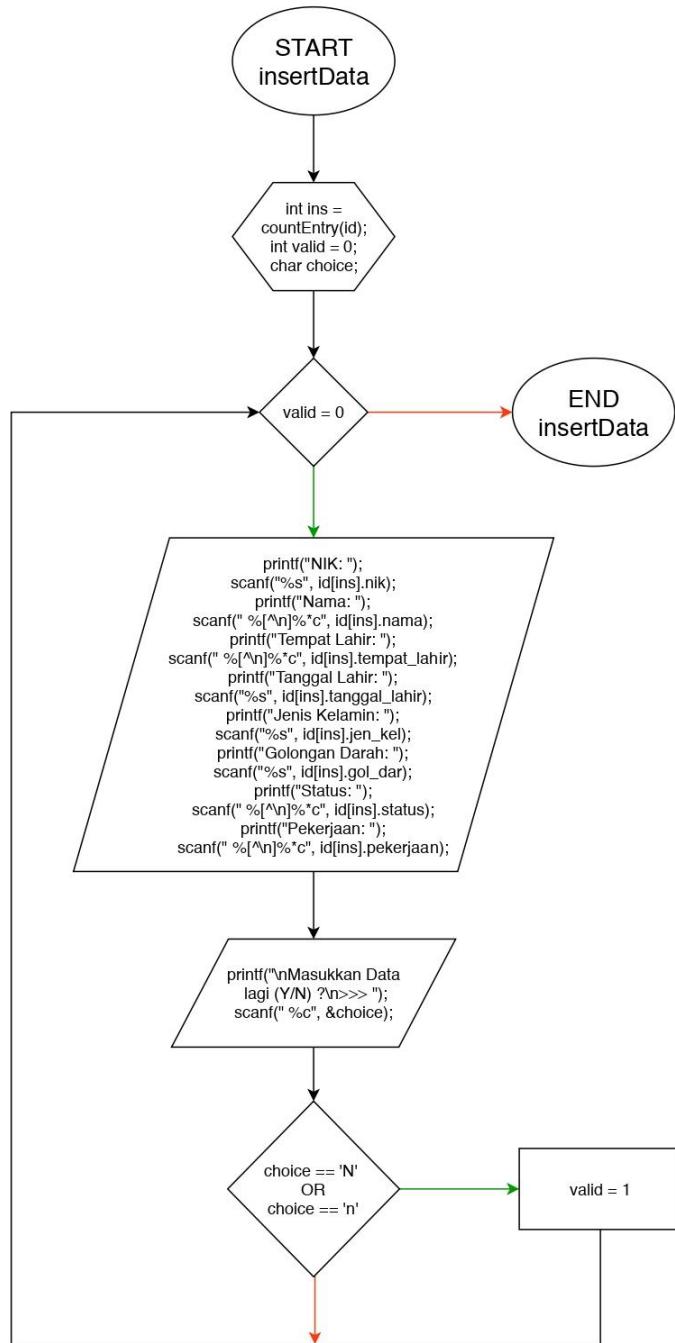
3.4.3. FUNGSI EDITDATA



Gambar 3-11. Flowchart dari Fungsi editData

3.4.4. FUNGSI INSERTDATA

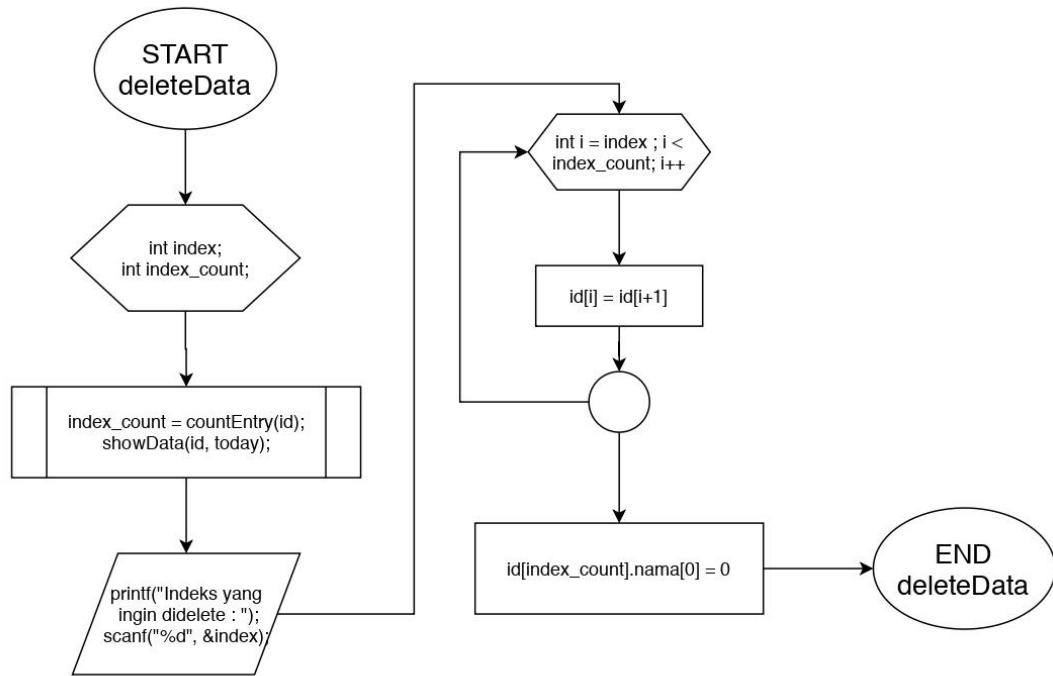
void insertData(data id[])



Gambar 3-12. Flowchart dari Fungsi insertData

3.4.5. FUNGSI DELETEDATA

void deleteData(data id[], date today)



Gambar 3-13. Flowchart dari Fungsi deleteData

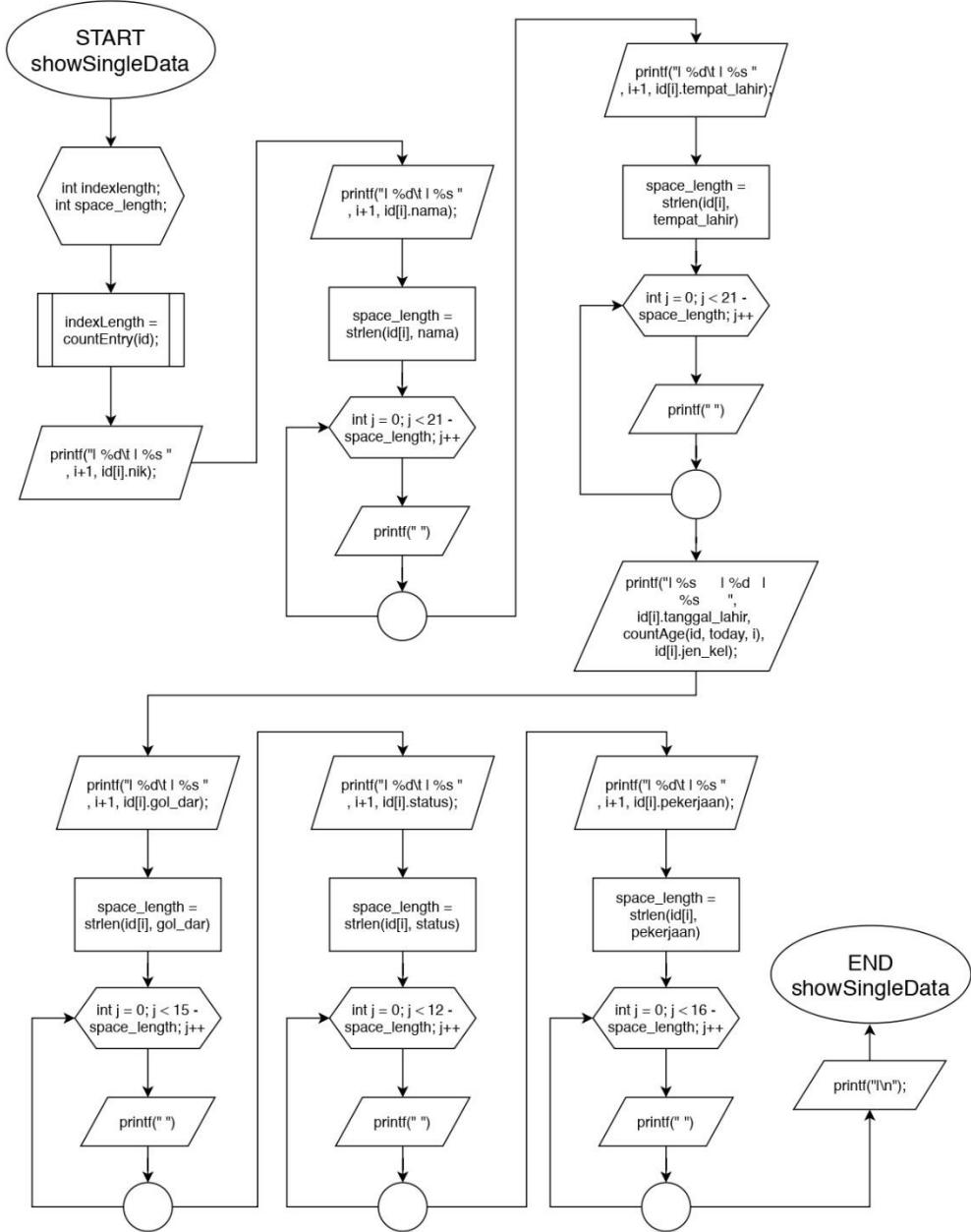
3.4.6. FUNGSI SHOWDATA



Gambar 3-14. Flowchart dari Fungsi showData

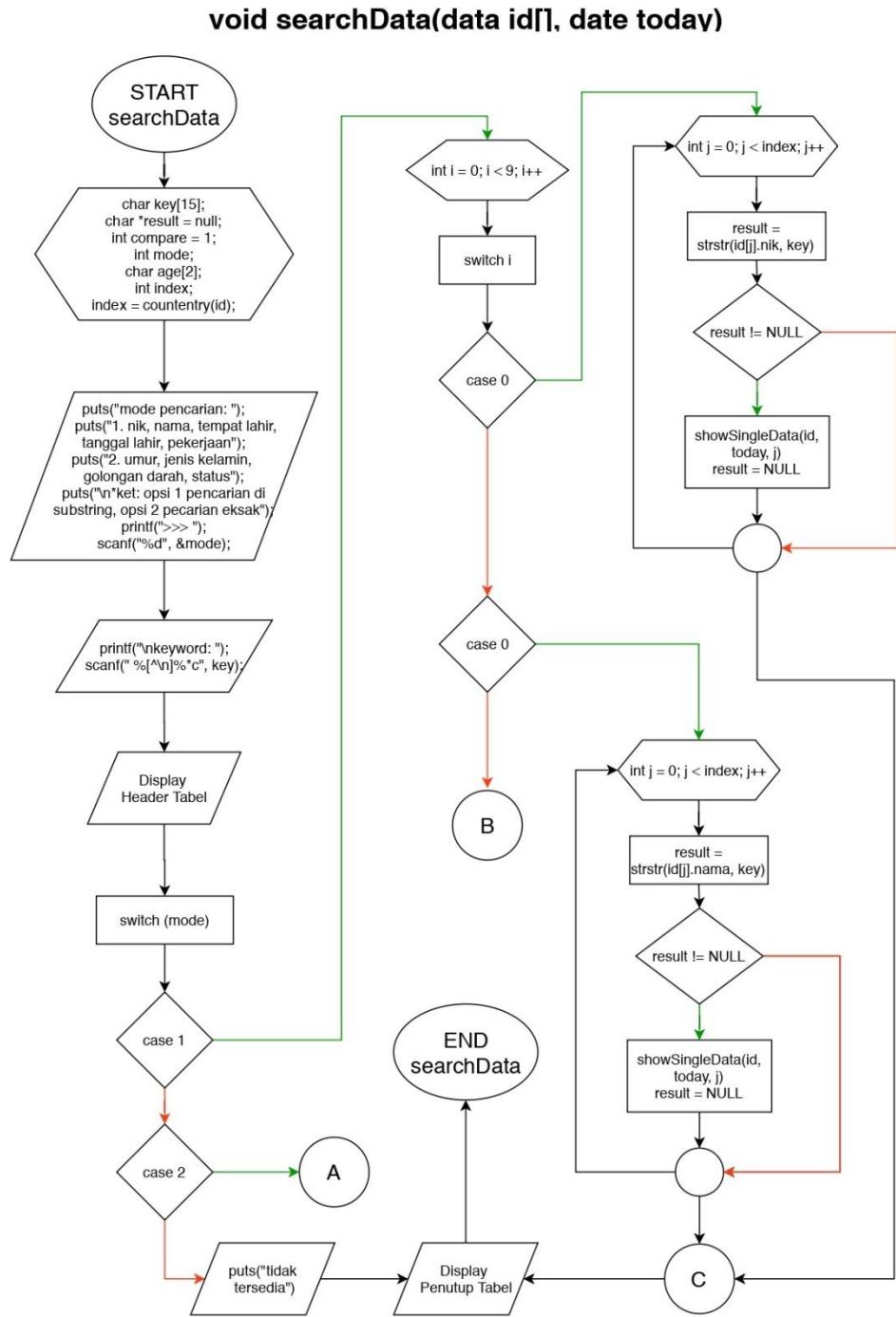
3.4.7. FUNGSI SHOWSINGLEDATA

void showSingleData(data id[], date today)

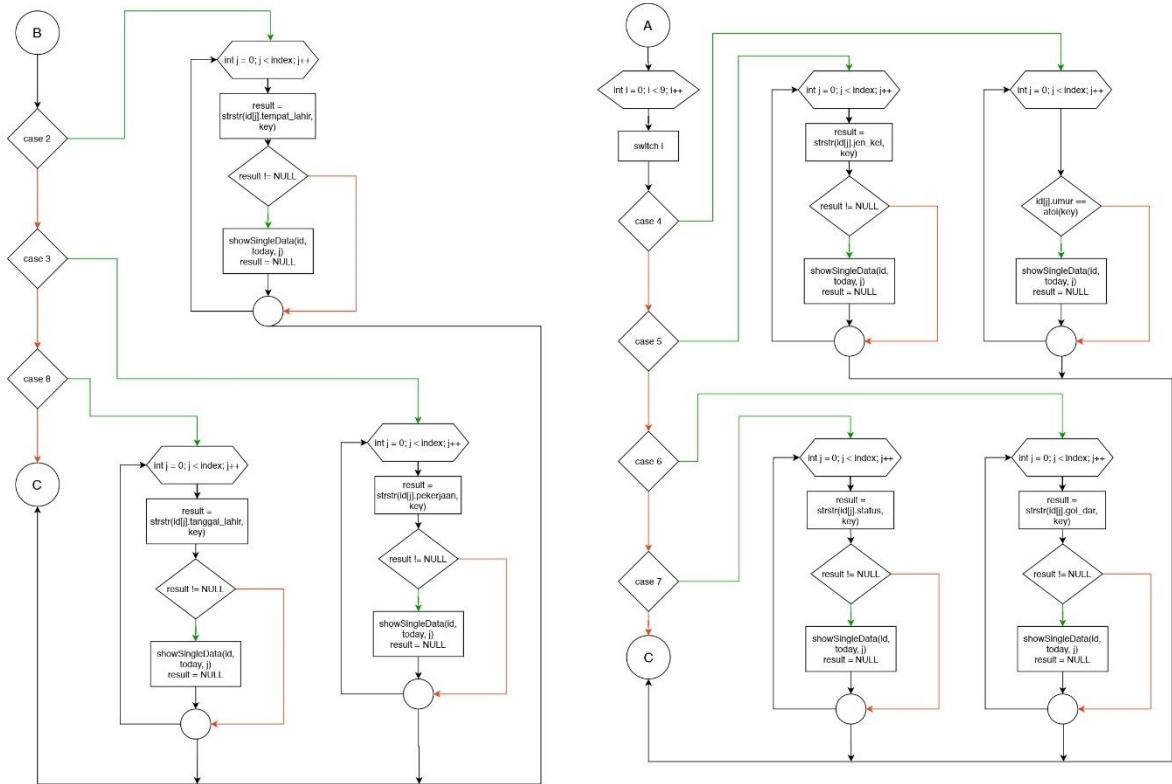


Gambar 3-15. Flowchart dari Fungsi showSingleData

3.4.8. FUNGSI SEARCHDATA



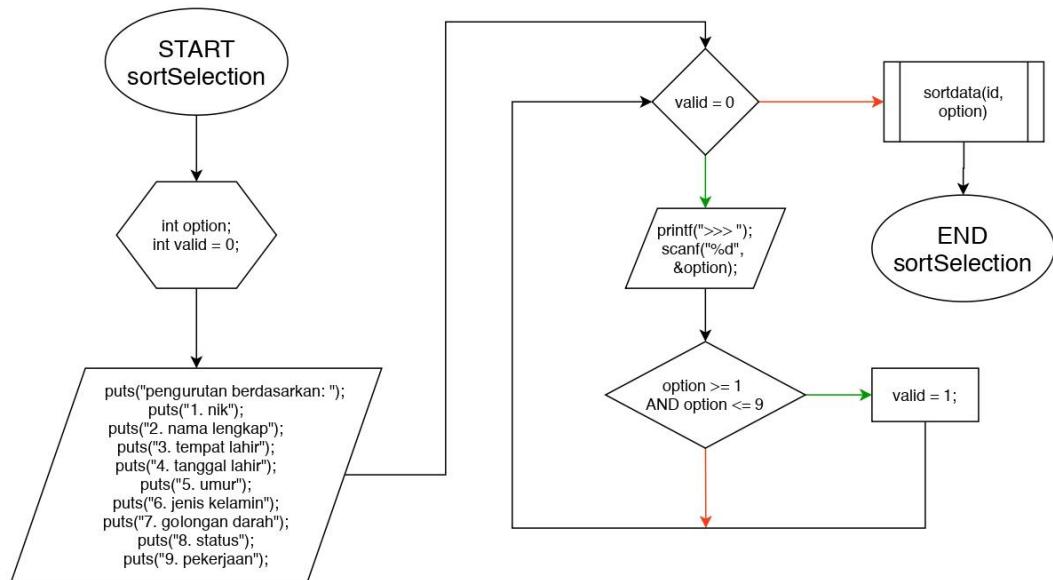
Gambar 3-16a. Flowchart dari Fungsi searchData



Gambar 3-16b. Lanjutan Flowchart dari Fungsi searchData

3.4.9. FUNGSI SORTSELECTION

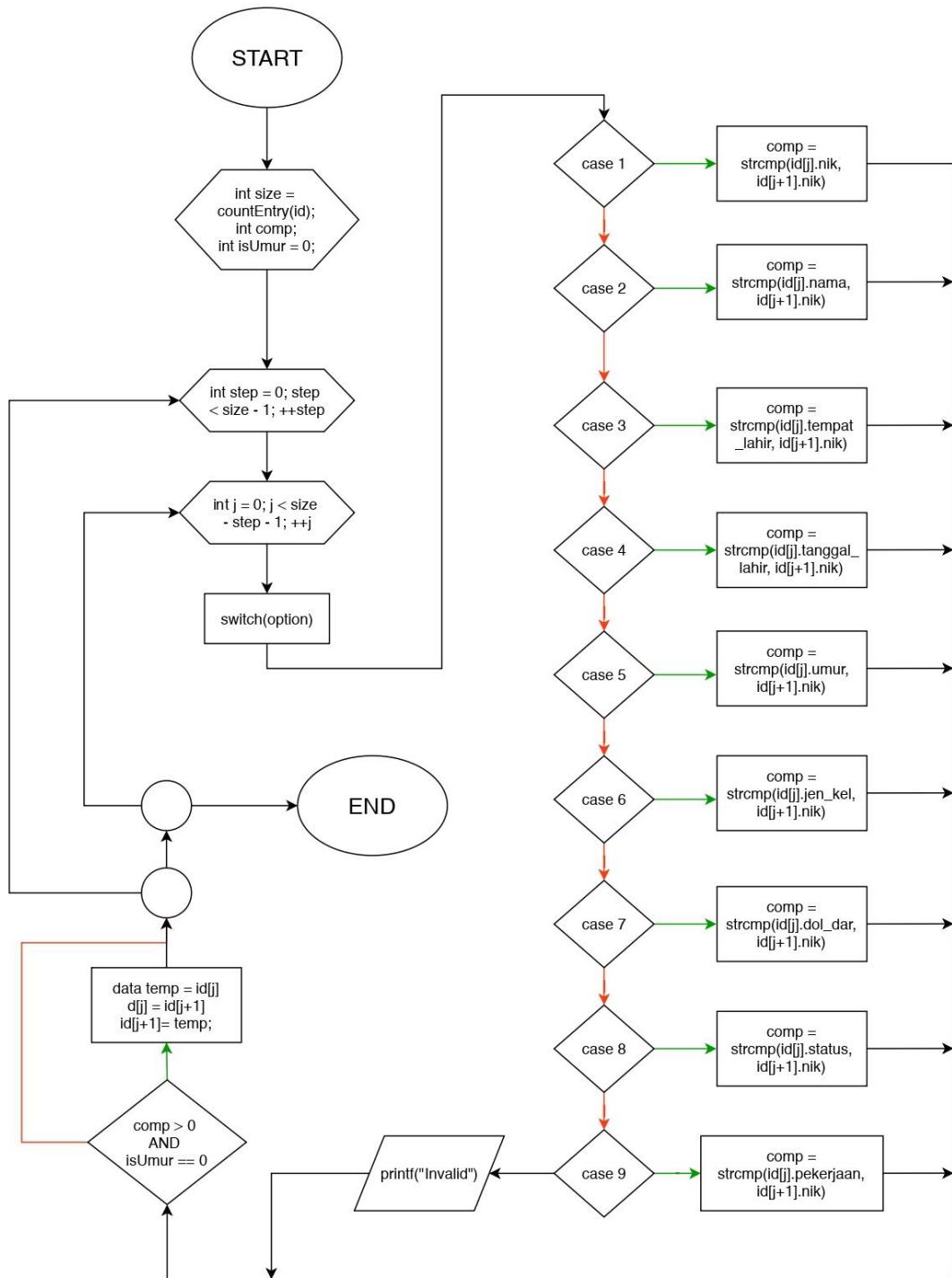
void sortSelection(data id[])



Gambar 3-17. Flowchart dari Fungsi sortSelection

3.4.10. FUNGSI SORTDATA

void sortData(data id[], int sort_by)



Gambar 3-18. Flowchart dari Fungsi sortData

4. DESAIN UJI FUNGSI

Pada bab ini, hanya akan diuji fungsi komplementer yang tidak terlihat secara riil pada program utama. Hasil pengujian dari fungsi lainnya dapat dilihat di bab pengujian, atau bab 6.

4.1. MENUSELECT

```
Menu :  
1. Input Data Manual  
2. Input Data dari File Eksternal  
3. Tampilkan Database  
4. Edit Data  
5. Hapus Data  
6. Temukan Data  
7. Urutkan Data  
8. Simpan Data ke File Eksternal  
9. Keluar  
>>> 10  
Pilihan belum tersedia, silahkan coba lagi  
>>> 11  
Pilihan belum tersedia, silahkan coba lagi  
>>> 12  
Pilihan belum tersedia, silahkan coba lagi  
>>> 0  
Pilihan belum tersedia, silahkan coba lagi  
>>> 2  
  
Masukka nama file : |
```

Gambar 4-1. Pengujian fungsi MenuSelect

Dari gambar 4-1 terlihat bahwa menuSelect tidak menerima inputan diluar angka 1 - 9 sebagai input yang valid sehingga meminta pengguna untuk mencoba input kembali. Dapat disimpulkan menuSelect bekerja dengan baik

4.2. INITDATE

```
Uji Fungsi Inisialisasi Tanggal  
Tanggal hari ini : 07/05/2020
```

Gambar 4-2. Pengujian fungsi initDate

Dari gambar 4-2 terlihat bahwa initDate menunjukkan tanggal hari ini, yaitu 7 Mei 2020, hari pengumpulan proyek ini. Dapat disimpulkan initDate bekerja dengan baik

4.3. COUNTAGE

```
Uji Kalkulasi Umur  
Tanggal Lahir = 01 01 1980  
Tanggal Hari ini = 07 05 2020  
Usia = 40
```

Gambar 4-3. Pengujian fungsi countAge

Dari gambar 4-3 terlihat bahwa countAge berhasil menghitung usia dari test case dengan akurat. Dapat disimpulkan countAge bekerja dengan baik

4.4. COUNTENTRY

Dari gambar 4-4 terlihat bahwa countEntry menunjukkan jumlah indeks yang tepat, yaitu 20 indeks. Dapat disimpulkan initDate bekerja dengan baik

No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan
1	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta
2	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta
3	1002861756	Sofia Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS
4	1005912127	Elly Sugisi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS
5	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta
6	1012046732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS
7	1015119023	Michael Tambuan	Tegal	01071986	33	L	O	Kawin	Wiraswasta
8	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta
9	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta
10	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta
11	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS
12	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta
13	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS
14	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS
15	0911960898	Inggrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta
16	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS
18	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS
19	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta
20	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta

Gambar 4-4. Pengujian fungsi countEntry

5. IMPLEMENTASI

Pada bab ini, akan ditampilkan realisasi yang telah dibuat dari permasalahan dalam bentuk *code* pada bahasa C, juga akan dibahas secara singkat mengenai *time complexity* dari *code*. Perlu diketahui bahwa akan terdapat sedikit perbedaan antara kode yang terlampir disini dibanding dengan kode yang dikumpulkan. Namun perbedaan tersebut hanya terdapat pada bagian estetikanya saja seperti pemberian enter tambahan pada terminal, atau notifikasi telah selesai menjalankan suatu program. Untuk logikanya, dapat diasumsikan sama. Pada bagian ini, tidak akan dilakukan fo

5.1. FUNGSI UTAMA

```
int main() {
    //Deklarasi
    int option = 0;
    data id[ENTRY_MAX];
    date today;

    //Inialisasi
    for (int i = 0; i < ENTRY_MAX;
i++) {
        id[i].nama[0] = 0;
    }
    title();
    today = initDate();
    while (option != 9){
        option = menuSelect();
        reset();
        puts("");
        switch(option){
            case 1:
                insertData(id);
                break;
            case 2:
                readData(id,
countEntry(id));
                break;
            case 3:
                showData(id, today);
                break;
            case 4:
                editData(id, today);
                break;
            case 5:
                deleteData(id, today);
                break;
            case 6:
                searchData(id, today);
                break;
            case 7:
                sortSelection(id);
                break;
            case 8:
                saveData(id);
                break;
            case 9:
```

```
printf("Shutting
Down... ");
break;
default:
puts("nani? ");
break;
}
}
```

Fungsi utama hanya mengandung switch case yang menerima input case dari fungsi menuSelect untuk memanggil fungsi processing yang bersangkutan.

5.2. FUNGSI PENUNJANG

5.2.1. MENUSELECT

```
int menuSelect(){
    int option;
    int valid = 0;

    puts("\nMenu :");
    puts("1. Input Data Manual");
    puts("2. Input Data dari File
Eksternal");
    puts("3. Tampilkan Database");
    puts("4. Edit Data");
    puts("5. Hapus Data");
    puts("6. Temukan Data");
    puts("7. Urutkan Data");
    puts("8. Simpan Data ke File
Eksternal");
    puts("9. Keluar");

    while (valid == 0){
        printf(">>> ");
        scanf("%d", &option);
        if (option >= 1 && option
<= 9){
            valid = 1;
        } else{
            puts("Pilihan belum
tersedia, silahkan coba lagi");
        }
    }

    return option;
}
```

Fungsi ini bertujuan untuk mengeluarkan pilihan menu utama, sekaligus sebagai validator dari pilihan masukkan pengguna. Proses validasi sederhana dilakukan pada looping while yang apabila nilai yang diinput

pengguna tidak terdapat di pilihan, program akan meminta masukkan kembali.

Kompleksitas dari fungsi ini ialah $O(n)$, mengingat bahwa terdapat validator yang akan terus berjalan selama pengguna memasukkan input sebanyak n kali hingga program menganggap input valid

5.2.2. INITDATE

```
date initDate() {
    date today;

    time_t now;
    time(&now);
    struct tm *local =
    localtime(&now);

    today.day = local->tm_mday;
    // get day of month (1 to 31)
    today.month = local->tm_mon +
    1; // get month of year (0 to
    11)
    today.year = local->tm_year +
    1900; // get year

    printf("Tanggal hari ini :
%02d/%02d/%d\n", today.day,
today.month, today.year);
    return today;
}
```

Dengan memanfaatkan `time.h`, fungsi ini dapat meng-generate tanggal secara *realtime*, sehingga usia penduduk dapat dikalkulasi dengan akurat. Fungsi ini juga akan mem-print tanggal hari ini pada tiap pemanggilannya.

Kompleksitas dari fungsi ini ialah $O(1)$ karena program tidak memiliki pengulangan, sehingga runtime akan selalu konstan.

5.2.3. COUNTAGE

```
int countAge(data id[], date today,
int i){
    int count;
    int age;
    date dob; //date of birth

    count = countEntry(id);

    dob.day =
    ((id[i].tanggal_lahir[0] - 48) *
    10) + (id[i].tanggal_lahir[1] -
```

```
48);
    dob.month =
    ((id[i].tanggal_lahir[2] - 48) *
    10) + (id[i].tanggal_lahir[3] -
    48);
    dob.year =
    ((id[i].tanggal_lahir[4] - 48) *
    1000) + ((id[i].tanggal_lahir[5] -
    48) * 100) +
    ((id[i].tanggal_lahir[6] - 48) *
```

Fungsi ini akan menghitung usia dari tiap penduduk dengan bantuan tanggal *realtime* yang diberi oleh fungsi sebelumnya yaitu `initDate`. Sama halnya seperti fungsi sebelumnya, kompleksitas dari fungsi ini ialah $O(1)$ karena program tidak memiliki pengulangan, sehingga runtime akan selalu konstan.

5.2.4. COUNTENTRY

```
int countEntry(data id[]){
    int count = 0;

    for (int i = 0; i < ENTRY_MAX;
    i++) {
        if (id[i].nama[0] == 0){
            break;
        }
        count++;
    }

    return count;
}
```

Fungsi ini merupakan salah satu fungsi vital pada program ini, karena berperan pada hampir seluruh fungsi processing lainnya untuk memberi informasi jumlah entry yang memiliki data yang valid. Sebagai contoh, agar pengguna dapat memasukkan data via file external berkali kali, fungsi ini akan mencegah file eksternal yang dibaca menimpa file yang sudah ada pada program. Kegunaan lain fungsi ini akan dijelaskan secara sporadis pada tiap fungsi processing (bila digunakan)

Kompleksitas dari fungsi ini ialah $O(n)$ dengan n ialah jumlah entry yang dimiliki oleh *database* karena memang fungsi dari program ini ialah melakukan looping dan bertindak sebagai counter terhadap banyaknya indeks pada *database*

5.3. FUNGSI PROCESSING

5.3.1. READDATA

```
void readData(data id[], int index_num){  
    FILE *fp;  
    char str[1024];  
    char filename[128];  
    int i = 0;  
    int valid = 0;  
    char dump[64];  
  
    while (valid == 0){  
        printf("Masukkan nama file : ");  
        scanf("%s", filename);  
        fp = fopen(filename,"r");  
        if (fp != NULL){  
            valid = 1;  
        }  
        else{  
            puts("File tidak ditemukan,  
silahkan coba lagi");  
        }  
  
        while(!feof(fp)){  
  
            fgets(str, 1024, fp);  
            if (i != 0){  
                strcpy(dump, strtok(str, ";"));  
                strcpy(id[i-1+index_num].nik,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].nama,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].tempat_lahir,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].tanggal_lahir,  
strtok(NULL, ";"));  
                strcpy(dump, strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].jen_kel,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].gol_dar,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].status,  
strtok(NULL, ";"));  
                strcpy(id[i-1+index_num].pekerjaan,  
strtok(NULL, "\n"));  
            }  
            i++;  
        }  
  
        fclose(fp);  
    }  
}
```

Fungsi `readData` akan menerima masukkan pengguna berupa nama file yang ingin dibaca, yang telah diperlengkapi dengan validator. Selanjutnya program akan memasukkan data dari file eksternal ke *database* melalui *string tokenization*. Token yang digunakan ialah titik koma untuk menyesuaikan dengan spesifikasi. Dengan bantuan `countEntry` yang akan dijadikan argumen bagi fungsi ini, fungsi ini dapat menghindari penumpukan data yang akan merusak *database*.

Kompleksitas dari fungsi ini ialah $O(n)$ dengan n ialah jumlah baris yang dimiliki oleh file eksternal karena fungsi ini memerlukan looping tiap baris untuk *parsing* data kedalam *database*.

5.3.2. SAVEDATA

```
void saveData(data id[]){  
    FILE *fp;  
    int index;  
    char option;  
    int valid = 0;  
    char filename[128];  
  
    FILE *fp;  
    int index;  
    char choice;  
    int valid = 0;  
    char date[8];  
    char day[2];  
    char month[2];  
    char year[4];  
  
    puts("Overwrite File Input Terakhir  
(Y/N)?");  
    while (valid == 0){  
        printf("">>>> ");  
        scanf(" %c", &choice);  
        if (choice == 'N' || choice ==  
'n'){  
            printf("Masukkan nama File  
(tanpa extension): ");  
            scanf("%s", filename);  
  
            itoa(today.day, day, 10);  
            itoa(today.month, month,  
10);  
            itoa(today.year, year, 10);  
            if (today.day <= 9){  
                date[0] = '0';  
                date[1] = day[0];  
            }  
            else{  
                date[0] = day[0];  
                date[1] = day[1];  
            }  
  
            if (today.month <= 9){  
                date[2] = '0';  
                date[3] = today.month +  
48;  
            }  
            else{  
                date[2] = month[0];  
                date[3] = month[1];  
            }  
            date[4] = year[0];  
            date[5] = year[1];  
            date[6] = year[2];  
            date[7] = year[3];  
            date[8] = '\0';  
            strcat(filename, date);  
            strcat(filename, ".csv");  
            valid = 1;  
        }  
    }  
}
```

```

        }
        else if (choice == 'Y' || choice == 'y'){
            valid = 1;
        }
        else{
            puts("Pilihan tidak tersedia, silahkan coba lagi\n");
        }
    }

    fp = fopen(filename, "w+");
    index = countEntry(id);
    for (int i = 0; i < index+1; i++){
        if (i == 0){
            fprintf(fp,
"No;NIK;NamaLengkap;TempatLahir;Tanggal
lLahir;Umur;JenisKelamin;GolonganDarah
;Status;Pekerjaan\n");
        }
        else{
            fprintf(fp,
"%d;%s;%s;%s;%d;%s;%s;%s;%s", i,
id[i-1].nik,
id[i-1].nama,
id[i-1].tempat_lahir,
id[i-1].tanggal_lahir,
id[i-1].umur,
id[i-1].jen_kel,
id[i-1].gol_dar,
id[i-1].status,
id[i-1].pekerjaan);
        }

        if (i < index && i != 0){
            fprintf(fp, "\n");
        }
    }
    fclose(fp);
    printf("\nData berhasil tersimpan di csv dengan nama %s", filename);
}

```

Fungsi saveData ini akan menerima masukkan pengguna berupa nama file untuk output save. Bagian ini tidak diperlengkapi oleh validator. Kerja dari fungsi ini cukup jelas, yaitu melakukan *printing* ke file eksternal sesuai dengan format csv pada spesifikasi, yaitu separator titik koma.

Sama halnya dengan fungsi readData, kompleksitas dari fungsi ini ialah O(n) dengan n ialah jumlah entry yang dimiliki oleh database karena program memerlukan looping

untuk menulis data dari tiap indeks ke file eksternal.

5.3.3. EDITDATA

```

void editData(data id[], date today){
    int index;
    char placeholder[MAX];
    showData(id, today);

    printf("Pilih indeks yang ingin diedit : ");
    scanf("%d", &index);
    index--;

    puts("\nSilahkan isi dengan '-' apabila
tidak ingin melakukan perubahan");
    printf("NIK: ");
    scanf("%s", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].nik, placeholder);
    }

    printf("Nama: ");
    scanf(" %[^\n]*c", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].nama, placeholder);
    }

    printf("Tempat Lahir: ");
    scanf(" %[^\n]*c", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].tempat_lahir,
placeholder);
    }

    printf("Tanggal Lahir: ");
    scanf("%s", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].tanggal_lahir,
placeholder);
    }

    printf("Jenis Kelamin: ");
    scanf("%s", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].jen_kel,
placeholder);
    }

    printf("Golongan Darah: ");
    scanf("%s", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].gol_dar,
placeholder);
    }

    printf("Status: ");
    scanf(" %[^\n]*c", placeholder);
    if (placeholder[0] != '-'){
        strcpy(id[index].status,
placeholder);
    }
}

```

Fungsi editData ini dapat mengubah data yang tersimpan pada *database* untuk suatu indeks tertentu. Pertama, pengguna akan memasukkan indeks pilihan nya untuk diedit. Masukkan ini tidak diperlengkapi dengan validator. Fungsi ini telah diperlengkapi untuk

mengedit semua parameter, namun bila pengguna hanya ingin mengubah beberapa parameter saja, fungsi juga diperlengkapi dengan pengandaian apabila pengguna memasukkan input ‘-’, program menganggap pengguna tidak ingin mengubah data pada parameter tersebut dan program akan maju ke parameter selanjutnya.

Kompleksitas dari program ini ialah O(1) dikarenakan tidak adanya validator serta pengulangan pada fungsi

5.3.4. INSERTDATA

```
void insertData(data id[]){
    int ins = countEntry(id);
    int valid = 0;
    char choice;

    while (valid == 0){
        printf("NIK: ");
        scanf("%s", id[ins].nik);
        printf("Nama: ");
        scanf(" %[^\n]*c", id[ins].nama);
        printf("Tempat Lahir: ");
        scanf("%s", id[ins].tempat_lahir);
        printf("Tanggal Lahir: ");
        scanf("%s", id[ins].tanggal_lahir);
        printf("Jenis Kelamin: ");
        scanf("%s", id[ins].jen_kel);
        printf("Golongan Darah: ");
        scanf("%s", id[ins].gol_dar);
        printf("Status: ");
        scanf(" %[^\n]*c", id[ins].status);
        printf("Pekerjaan: ");
        scanf(" %s", id[ins].pekerjaan);

        printf("\nMasukkan Data lagi (Y/N)
?>> ");
        scanf(" %c", &choice);

        if (choice == 'N' || choice ==
'n'){
            valid = 1;
        }
        ins++;
    }
}
```

Fungsi ini berperan untuk menerima masukkan berupa parameter *database* dari pengguna dan menempatkannya ke array paling akhir. Fungsi ini tidak diperlengkapi oleh validator sama sekali, mengingat ketersediaan waktu yang kurang.

Kompleksitas dari fungsi ini ialah O(n) dengan n ialah jumlah pemasukkan data yang dilakukan oleh pengguna karena pada fungsi ini terdapat looping yang terus dilakukan

hingga pengguna menulis N atau n untuk menghentikan proses input data.

5.3.5. DELETEDATA

```
void deleteData(data id[], date today){
    int index;
    int index_count;
    index_count = countEntry(id);

    showData(id, today);
    printf("Indeks yang ingin didelete
: ");
    scanf("%d", &index);
    index--;
    for (int i = index ; i <
index_count; i++){
        id[i] = id[i+1];
    }

    id[index_count].nama[0] = 0;
}
```

Fungsi ini berperan untuk menerima masukkan berupa indeks dari *database* yang ingin pengguna hapus dari *database*. Fungsi ini juga tidak memiliki validator sehingga diasumsikan indeks yang dipilih valid dan tersedia pada tabel

Kompleksitas dari fungsi ini ialah O(n) dengan n ialah jumlah entry yang akan digeser pada *database* yang diakibatkan oleh pen-delete-an indeks tertentu dari suatu array. Apabila digunakan linked list, kompleksitas akan menjadi O(1). Namun karena keterbatasan waktu, digunakan static array pada implementasi ini.

5.3.6. SHOWDATA

```
void showData(data id[], date today){
    int indexLength;
    int space_length;
    indexLength = countEntry(id);
    yellow();
    ======HEADER TABEL(diimplementasikan
di kode)=====
    for (int i = 0; i < indexLength; i++){
        //No dan NIK
        printf("| %d\t | %s ", i+1,
id[i].nik);

        //Nama
        printf(" | %s", id[i].nama);
        space_length = strlen(id[i].nama);
        for (int j = 0; j < 21 -
space_length; j++){
            printf(" ");
        }

        //Tempat Lahir
        printf(" | %s", id[i].tempat_lahir);
    }
}
```

```

        space_length =
strlen(id[i].tempat_lahir);
    for (int j = 0; j < 21 -
space_length; j++){
        printf(" ");
    }

    //Tanggal Lahir, Umur, Jenis
    Kelamin
    printf("| %s | %d | %s
", id[i].tanggal_lahir, countAge(id, today,
i), id[i].jen_kel);

    //Golongan Darah
    printf("| %s", id[i].gol_dar);
    space_length =
strlen(id[i].gol_dar);
    for (int j = 0; j < 15 -
space_length; j++){
        printf(" ");
    }

    //Status
    printf("| %s", id[i].status);
    space_length =
strlen(id[i].status);
    for (int j = 0; j < 12 -
space_length; j++){
        printf(" ");
    }

    //Pekerjaan
    printf("| %s", id[i].pekerjaan);
    space_length =
strlen(id[i].pekerjaan);
    for (int j = 0; j < 16 -
space_length; j++){
        printf(" ");
    }

    //New Line
    printf("| \n");
}
=====TAIL TABEL(diimplementasikan
di kode)=====
}

```

Fungsi ini berperan untuk meng-generate tabel pada *database* sehingga pengguna dapat lebih mudah untuk melakukan rekayasa lebih lanjut pada *database*. Pada pemanggilan fungsi ini pula data parameter usia pengguna diisi dan diupdate, sehingga apabila pengguna telah menjalankan program ini selama berhari-hari dan ingin mengunggah *database* ke file eksternal, disarankan untuk memanggil fungsi ini (terdapat pada main menu) untuk *refresh* umur dari penduduk.

Fungsi ini juga dapat menyesuaikan karakter spasi yang digunakan pada tiap *cell*, asalkan jumlah karakter dari tiap parameter tidak melewati batas maksimal sistem (silahkan merujuk pada asumsi awal di bab 2)

Kompleksitas dari fungsi ini ialah $O(n^2)$ dengan n ialah jumlah entry yang dimiliki oleh *database*, karena terdapat nested for pada fungsi ini yang bertujuan untuk meng-generate space character berdasarkan jumlah karakter pada tiap *cell*.

5.3.7. SHOWSINGLEDATA

Fungsi ini pada dasarnya sama seperti fungsi *showData* namun fungsi ini hanya menampilkan 1 baris saja sesuai masukkan pada argumen fungsi. Fungsi ini nantinya akan berguna pada penampilan hasil pencarian oleh fungsi *searchData*.

Kompleksitas dari fungsi ini ialah $O(n)$ dengan n ialah jumlah spasi total yang perlu dicetak oleh sistem karena fungsi ini merupakan implementasi fungsi sebelumnya.

5.3.8. SORTSELECTION

```

void sortSelection(data id[]){
    int option;
    int valid = 0;

    puts("Pengurutan berdasarkan: ");
    puts("1. NIK");
    puts("2. Nama Lengkap");
    puts("3. Tempat Lahir");
    puts("4. Tanggal Lahir");
    puts("5. Umur");
    puts("6. Jenis Kelamin");
    puts("7. Golongan Darah");
    puts("8. Status");
    puts("9. Pekerjaan");

    while (valid == 0){
        printf("">>>> ");
        scanf("%d", &option);
        if (option >= 1 && option <=
9) {
            valid = 1;
        }
    }

    sortData(id, option);
}

```

Fungsi ini merupakan fungsi persiapan untuk memanggil fungsi *sortData*. Maksud dari persiapan disini ialah pencetakan pilihan pengurutan berdasarkan parameter, serta validator agar program dapat berjalan dengan baik.

Kompleksitas dari fungsi ini ialah O(n), mengingat bahwa terdapat validator yang akan terus berjalan selama pengguna

memasukkan input sebanyak n kali hingga program menganggap input valid.

5.3.9. SEARCHDATA

Pada pemamparan fungsi ini dan fungsi selanjutnya, tidak akan dibelah menjadi dua untuk memperlihatkan case yang terdapat pada kode dengan jelas

```
void searchData(data id[], date today){
    char key[15];
    char *result = NULL;
    int compare = 1;
    int mode;
    char age[2];
    int index;
    index = countEntry(id);

    puts("Mode Pencarian: ");
    puts("1. Pencarian di bagian NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan");
    puts("2. Pencarian di bagian Umur, Jenis Kelamin, Golongan Darah, dan Status");
    puts("\n*ket: Opsi 1 pencarian di substring, Opsi 2 pecarian eksak");
    printf(">>> ");
    scanf("%d", &mode);

    printf("\nKeyword: ");
    scanf(" %[^\n]*c", key);

=====HEADER TABEL(diimplementasikan di kode)=====

switch(mode){
    case 1:
        for (int i = 0; i < 9; i++){
            switch(i){
                case 0:
                    for (int j = 0; j < index; j++){
                        result = strstr(id[j].nik, key);
                        if (result != NULL){
                            showSingleData(id, today, j);
                            result = NULL;
                        }
                    }
                    result = NULL;
                    break;
                case 1:
                    for (int j = 0; j < index; j++){
                        result = strstr(id[j].nama, key);
                        if (result != NULL){
                            showSingleData(id, today, j);
                            result = NULL;
                        }
                    }
                    result = NULL;
                    break;
                case 2:
                    for (int j = 0; j < index; j++){
                        result = strstr(id[j].tempat_lahir, key);
                        if (result != NULL){
                            showSingleData(id, today, j);
                            result = NULL;
                        }
                    }
                    result = NULL;
                    break;
            }
        }
}
```

```

        case 3:
            for (int j = 0; j < index; j++) {
                result = strstr(id[j].tanggal_lahir, key);
                if (result != NULL) {
                    showSingleData(id, today, j);
                    result = NULL;
                }
            }
            result = NULL;
            break;
        case 4:
            break;
        case 5:
            break;
        case 6:
            break;
        case 7:
            break;
        case 8:
            for (int j = 0; j < index; j++) {
                result = strstr(id[j].pekerjaan, key);
                if (result != NULL) {
                    showSingleData(id, today, j);
                    result = NULL;
                }
            }
            result = NULL;
            break;
        default:
            break;
    }
}
break;
case 2:
for (int i = 0; i < 9; i++) {
switch(i) {
    case 0:
        break;
    case 1:
        break;
    case 2:
        break;
    case 3:
        break;
    case 4:
        //umur
        for (int j = 0; j < index; j++) {
            if (id[j].umur == atoi(key)) {
                showSingleData(id, today, j);
            }
        }
        break;
    case 5:
        //jenkel
        for (int j = 0; j < index; j++) {
            compare = strcmp(id[j].jen_kel, key);
            if (compare == 0) {
                showSingleData(id, today, j);
                compare = 1;
            }
        }
}
}

```

```

        }
        break;
    case 6:
        //golongan darah
        for (int j = 0; j < index; j++){
            compare = strcmp(id[j].gol_dar, key);
            if (compare == 0){
                showSingleData(id, today, j);
                compare = 1;
            }
        }
        break;
    case 7:
        for (int j = 0; j < index; j++){
            compare = strcmp(id[j].status, key);
            if (compare == 0){
                showSingleData(id, today, j);
                compare = 1;
            }
        }
        result = NULL;
        break;
    case 8:
        break;
    default:
        break;
    }
}
break;
default:
    puts("tidak tersedia");
    break;
}
puts("-----");
}

```

Fungsi ini bertujuan untuk mencari suatu data dari *database* pada program ini. Sebelum melakukan pencarian, pengguna akan diminta oleh program untuk memasukkan parameter apa yang ingin dicari. Pilihan ini dibagi 2, yang pertama ialah pencarian pada parameter NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan. Dan yang kedua ialah pencarian pada parameter Umur, Jenis Kelamin, Golongan Darah, dan Status.

Penulis memilih untuk memisahkan kedua pilihan ini karena pada pilihan pertama, telah diterapkan pencarian dengan metode *substring*, sehingga tidak perlu *match whole word* untuk menampilkan apa yang dicari. Sedangkan untuk pilihan kedua, penulis menerapkan pencarian dengan metode *exact*, karena misalkan bila ingin dicari golongan

darah A dengan metode *substring*, program akan menunjukkan semua data yang memiliki karakter "A" didalamnya, tentunya hal ini tidak baik, sehingga untuk pilihan kedua telah diterapkan pencarian dengan metode *exact*. Dua pilihan ini terbagi oleh dua case yang terdapat pada kode, dan tiap case hanya mencari pada parameter yang telah dijabarkan

Kompleksitas dari fungsi ini ialah $O(n)$ dengan n ialah jumlah entry yang dimiliki oleh *database* dikali dengan konstanta 5, karena tiap pilihan pada program akan mencari data pada 5 parameter yang berbeda. Meskipun terdapat nested for pada program ini, salah satu pengulangan for yang digunakan memiliki batasan konstan sehingga hanya dianggap terdapat satu loop yang memiliki batas looping yang berubah ubah.

5.3.10. SORTDATA

```
void sortData(data id[], int sort_by){  
    //tes sort nama  
    int size = countEntry(id);  
    int comp;  
    int isUmur = 0;  
  
    for (int step = 0; step < size - 1; ++step) {  
        for (int j = 0; j < size - step - 1; ++j) {  
            switch(sort_by){  
                case 1:  
                    comp = strcmp(id[j].nik, id[j+1].nik);  
                    break;  
                case 2:  
                    comp = strcmp(id[j].nama, id[j+1].nama);  
                    break;  
                case 3:  
                    comp = strcmp(id[j].tempat_lahir, id[j+1].tempat_lahir);  
                    break;  
                case 4:  
                    comp = strcmp(id[j].tanggal_lahir, id[j+1].tanggal_lahir);  
                    break;  
                case 5:  
                    comp = strcmp(id[j].jen_kel, id[j+1].umur);  
                    break;  
                case 6:  
                    comp = strcmp(id[j].jen_kel, id[j+1].jen_kel);  
                    break;  
                case 7:  
                    comp = strcmp(id[j].gol_dar, id[j+1].gol_dar);  
                    break;  
                case 8:  
                    comp = strcmp(id[j].status, id[j+1].status);  
                    break;  
                case 9:  
                    comp = strcmp(id[j].pekerjaan, id[j+1].pekerjaan);  
                    break;  
                default:  
                    puts("invalid");  
            }  
  
            if (comp > 0 && isUmur == 0){  
                //swap  
                data temp = id[j];  
                id[j] = id[j+1];  
                id[j+1]= temp;  
            }  
        }  
    }  
}
```

Fungsi ini akan melakukan sorting dengan algoritma dasar *Bubble Sort* pada parameter yang telah dipilih pada fungsi sebelumnya. Dapat dilihat bahwa bila bagian switch case diabaikan, algoritma pada fungsi ini ialah algoritma *bubble sort* sederhana dengan *swapping* dasar. Banyaknya case yang digunakan diatas hanya untuk mengisi variabel "comp" dari hasil komparasi antara kedua string. Fungsi *strcmp* yang digunakan

dapat melakukan sorting secara lexicografikal, dimana sorting dilakukan secara alfabetis / numeris seperti layaknya pada kamus.

Kompleksitas dari fungsi ini ialah $O(n^2)$ karena fungsi ini mengimplementasikan algoritma *bubble sort* sederhana yang menggunakan nested loops, yang bertujuan untuk melakuakn sorting pada *database*.

6. HASIL PENGUJIAN DAN ANALISIS

Pada pengujian program, akan ditampilkan tangkapan layar hasil kompilasi melalui VSCode dengan warna *custom* pada UI nya. Namun kode yang telah dilampirkan sebelumnya dan yang telah dikumpulkan tidak akan menunjukkan warna dikarenakan warna ini tidak dapat muncul dengan konsisten pada terminal penulis, sehingga penulis tidak bisa menjamin *User Experience* yang maksimal bila mengumpulkan kode yang mengimplementasi warna tersebut. Namun bila pembaca hendak mencoba fitur ini, pembaca dapat melakukan *uncomment* pada fungsi warna yang terdapat pada program dan mengkompilasi ulang, atau untuk kemudahan pembaca, penulis juga telah melampirkan program bernama ‘main_color.exe’ yang hanya memiliki perbedaan pada bagian kode yang ditunjukkan pada gambar 6-1.

```
113 //aesthetics
114 void red(){
115     printf("\033[1;31m");
116 }
117
118 void yellow(){
119     printf("\033[1;33m");
120 }
121
122 void blue(){
123     printf("\033[1;34m");
124 }
125
126 void reset(){
127     printf("\033[0m");
128 }
```

Gambar 6-1. Fungsi yang perlu di *uncomment* agar dapat menampilkan warna

Konten dari bab ini akan dibagi per spesifikasi yang terdapat pada program, yang dapat dilihat pada subbab subbab dibawah ini

6.1. INPUT DATA MANUAL

Gambar 6-2 menunjukkan tampilan terminal saat pertama kali dijalankan, serta pada saat memilih mode 1. Pengguna akan memasukkan parameter yang diminta oleh program, lalu data tersebut akan tersimpan di *database*. Pengguna dapat memasukkan data berulang kali, namun untuk pemaparan ini hanya dimasukkan satu kali. Hasil dari input pengguna dapat dilihat pada gambar 6-3. Gambar 6-3 ialah tabel data yang digenerasi oleh program secara otomatis. Untuk pengujian lebih detailnya dapat dilihat pada subbab 6.3 dan penjabaran dari sintaks pada subbab 5.3.6



Gambar 6-2. Tampilan terminal saat memilih menu 1

Tabel Data Penduduk Pada Sistem									
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan
1	1234567890	Kevin Naoko	Jakarta	06142000	19	L	AB	Tidak Kawin	Mahasiswa

Gambar 6-3. Tampilan tabel *database* setelah memilih menu 1

6.3. INPUT DATA DARI FILE EKSTERNAL

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 2

Masukka nama file (tanpa extension): DataPenduduk
File DataPenduduk.csv berhasil diload
  
```

Gambar 6-4. Tampilan terminal saat memilih menu 2

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	1234567890	Kevin Naoko	Jakarta	06142000	19	L	AB	Tidak Kawin	Mahasiswa	
2	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta	
3	099820635	Cintia Damank	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
4	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
5	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
6	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
7	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
8	1015119023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
9	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
10	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
11	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
12	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
13	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
14	0982161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
15	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
16	0911960898	Ingrid Yusti	Kediri	22021983	37	P	AB	Kawin	Wiraswasta	
17	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
18	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
19	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
20	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
21	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Gambar 6-5. Tampilan tabel saat sesudah meng-import data dari csv setelah memilih menu 1.

6.4. MENAMPILKAN DATABASE

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	1234567890	Kevin Naoko	Jakarta	06142000	19	L	AB Rh	Kawin	Pegawai Swasta	
2	0999620635	Cintia Damank	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
3	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
4	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
5	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
6	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
7	1015119023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
8	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
9	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
10	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
11	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
12	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
13	0982161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
14	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
15	0911960898	Ingrid Yusti	Kediri	22021983	37	P	AB	Kawin	Wiraswasta	
16	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
18	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
19	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
20	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Gambar 6-6. Tampilan terminal setelah memilih menu 3.

Apabila pengguna memilih mode 3, pengguna dapat melihat kondisi tabel setelah dilakukan operasi tertentu, yang meliputi pengeditan, penghapusan, penambahan, dan pengurutan data. Nomor indeks pada database ini selalu

Pengujian pada sbb ini berkaitan dengan sbb sebelumnya, yaitu setelah dilakukan input manual oleh pengguna, akan dilakukan input melalui CSV dengan menu 2. Hasilnya ialah tidak terjadi overlapping pada data yang sudah tersimpan di CSV. Jadi jika diinginkan, pengguna dapat memasukkan referensi dari beberapa file eksternal yang berbeda dan program masih dapat berjalan dengan baik.

statistik 1 hingga 19 meskipun telah dilakukan pengurutan untuk menjaga estetika dari program.

Sekedar informasi yang penulis anggap sedikit relevan, karena tidak adanya validator untuk semua data, kolom golongan darah dan status dapat diisi selain dari data yang valid, namun tabel masih dapat ditampilkan tanpa

6.5. MENGEDIT DATA

Mode 4 pada program mempersilahkan pengguna untuk mengedit data yang sudah terdapat di *database*. Program pada awalnya akan memanggil fungsi `showData` terlebih dahulu untuk mempermudah rekayasa data. Apabila program digunakan untuk menghandle data yang banyak, fungsi ini `showData` dapat dicomment pada fungsi `editData`. Pada gambar 6-7 terlihat bahwa terdapat beberapa karakter yang diubah menjadi '-'. Sebetulnya itu bukan diubah menjadi ',', melainkan program mengabaikan

kerusakan. Contoh dari hal ini dapat dilihat pada gambar 6-6 kolom sgolongan darah, dan gambar 6-7 dan gambar 6-8, khususnya pada baris pertama pada kolom status.

No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan
1	9999999999	Kevin Naoko	Bandung	06142000	19	L	AB	Kawin hehe	Mahasiswa
2	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta
3	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta
4	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS
5	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS
6	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta
7	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS
8	1015119823	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta
9	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta
10	1021303722	Burret Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta
11	1024410187	Udi Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta
12	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS
13	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta
14	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS
15	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS
16	0911960898	Ingrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta
17	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta
18	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS
19	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS
20	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta
21	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta

Pilih indeks yang ingin diedit : 1

Silahkan isi dengan '-' apabila tidak ingin melakukan perubahan

NIK: 5555544444

Nama: -

Tempat Lahir: -

Tanggal Lahir: -

Jenis Kelamin: P

Golongan Darah: O

Status: Masih Suci

Pekerjaan: -

Entry berhasil diperbaharui

Gambar 6-7. Tampilan terminal saat memilih menu 4.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	5555544444	Kevin Naoko	Bandung	06142000	19	P	O	Masih Suci	Mahasiswa	
2	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta	
3	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
4	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
5	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
6	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
7	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
8	1015119023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
9	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
10	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
11	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
12	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
13	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
14	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
15	0911960898	Inggrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta	
16	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
18	9026866079	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
19	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
20	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Gambar 6-7b. Tampilan tabel setelah mengedit beberapa parameter dengan menu 4

6.6. MENGHAPUS DATA

Menu 5 pada program memperbolehkan pengguna untuk menghapus 1 indeks pada database. Program pada awalnya akan memanggil fungsi showData terlebih dahulu untuk mempermudah rekayasa data. Apabila program digunakan untuk menghandle data

yang banyak, fungsi ini showData dapat dicomment pada fungsi deleteData. Dapat dilihat pada gambar 6-8b bahwa penghapusan indeks berjalan dengan baik. Dapat dilihat pada gambar 6-8c bahwa bila dipilih indeks melebihi apa yang dimiliki pada data, maka tidak terjadi apa apa pada sistem.

Menu :										
1. Input Data Manual										
2. Input Data dari File Eksternal										
3. Tampilkan Database										
4. Edit Data										
5. Hapus Data										
6. Temukan Data										
7. Urutkan Data										
8. Simpan Data ke File Eksternal										
9. Keluar										
>> 5										
Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta	
2	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
3	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
4	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
5	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
6	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
7	1015119023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
8	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
9	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
10	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
11	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
12	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
13	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
14	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
15	0911960898	Inggrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta	
16	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
18	9026866079	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
19	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
20	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Indeks yang ingin di delete : 1
Data dengan nama Adam Suseno berhasil dihapus

Gambar 6-8. Tampilan terminal saat memilih menu 5.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
2	1002861756	Sofia Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
3	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
4	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
5	1012040732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
6	1015119023	Michael Tambuan	Tegal	01071986	33	L	O	Kawin	Wiraswasta	
7	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
8	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
9	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
10	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
11	1030651494	Limbud Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
12	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
13	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
14	0911960898	Inggrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta	
15	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
16	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
17	9026866679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
18	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
19	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Gambar 6-8b. Tampilan tabel setelah menjalankan menu 5.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
Indeks yang ingin didelete : 2										
Data dengan nama berhasil dihapus										

Gambar 6-8c. Tampilan saat indeks yang dipilih melewati indeks yang tersedia

6.7. MENCARI DATA

Untuk menguji menu 6, akan digunakan data seperti pada gambar 6-9. Pembaca dapat melihat pada gambar 6-10 hingga 6-13 hasil pencarian beberapa keyword menggunakan mode pencarian yang berbeda beda. Gambar 6-10 mencari kata ‘Coldman’ dengan mode 1, gambar 6-11 mencari kata ‘Kevin’ dengan mode 1, gambar 6-12 mencari kata ‘AB’ dengan mode 2, dan gambar 6-13 mencari umur ‘38’ dengan mode 2. Apabila tidak terdapat data yang dicari, program hanya menampilkan kerangka header dan tail tabel, dengan notifikasi Not Found!

Sebagai pengingat, pencarian dengan mode 1 akan mencari *database* pada parameter NIK, Nama, Tempat Lahir, Tanggal Lahir, dan

Pekerjaan secara substring. Dan mode pencarian kedua akan mencari pada parameter Umur, Jenis Kelamin, Golongan Darah, dan Status secara eksak.

Mekanisme dari search ini dipilih oleh penulis karena seperti yang dapat dilihat oleh pembaca, terdapat suatu keyword yang bila dicari dengan mode substring akan menampilkan banyak data yang menurut penulis tidak relevan. Sebagai contoh bila ingin dicari jenis kelamin P pada *database* dengan mode substring, maka akan terjadi false positive dimana program menganggap pengguna mencari “P” pada “PNS” atau “Pegawai Swasta”. Faktor demikian lah yang menyebabkan penulis untuk memisahkan kedua mode pencarian ini.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	5555544444	Kevin Naoko	Bandung	06142000	19	P	O	Masih Suci	Mahasiswa	
2	0996788736	Adam Suseno	Jakarta	01011980	40	L	AB	Kawin	Pegawai Swasta	
3	0999828635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
4	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
5	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
6	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
7	1012049732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
8	101519023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
9	1018206777	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
10	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
11	1024410187	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
12	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
13	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
14	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
15	0911960898	Inggrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta	
16	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
18	9026866679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
19	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
20	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	

Gambar 6-9. Tampilan database pada tabel.

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 6

Mode Pencarian:
1. Pencarian di bagian NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan
2. Pencarian di bagian Umur, Jenis Kelamin, Golongan Darah, dan Status

*ket: Opsi 1 pencarian di substring, Opsi 2 pecarian eksak
>>> 1

Keyword: Coldman

```

No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan
17	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS
18	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS

Gambar 6-10. Tampilan terminal variasi 1 saat menjalankan menu 6.

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 6

Mode Pencarian:
1. Pencarian di bagian NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan
2. Pencarian di bagian Umur, Jenis Kelamin, Golongan Darah, dan Status

*ket: Opsi 1 pencarian di substring, Opsi 2 pecarian eksak
>>> 1

Keyword: Kevin

```

No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan
1	5555544444	Kevin Naoko	Bandung	06142000	19	P	O	Masih Suci	Mahasiswa

Gambar 6-11. Tampilan terminal variasi 2 saat menjalankan menu 6.

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 6

Mode Pencarian:
1. Pencarian di bagian NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan
2. Pencarian di bagian Umur, Jenis Kelamin, Golongan Darah, dan Status

*ket: Opsi 1 pencarian di substring, Opsi 2 pecarian eksak
>>> 2

Keyword: AB
-----| No | NIK | Nama Lengkap | Tempat Lahir | Tanggal Lahir | Umur | Jenis Kelamin | Golongan Darah | Status | Pekerjaan |
-----| 2 | 0996788736 | Adam Suseno | Jakarta | 01011980 | 40 | L | AB | Kawin | Pegawai Swasta |
-----| 4 | 1002861756 | Sofie Agustin | Bandung | 01031982 | 38 | P | AB | Tidak Kawin | PNS |
-----| 6 | 1008971776 | Bambang Sutikno | Sumedang | 01051984 | 36 | L | AB | Kawin | Wiraswasta |
-----| 9 | 1018206677 | Melanie Hughes | Tangerang Selatan | 15061987 | 32 | P | AB | Kawin | Wiraswasta |
-----| 11 | 1024410187 | Uda Wahab | Yogyakarta | 17081989 | 30 | L | AB | Tidak Kawin | Pegawai Swasta |
-----| 14 | 0902161017 | Scott Triswara | Banyuwangi | 20111981 | 38 | L | AB | Kawin | PNS |
-----| 15 | 0911960898 | Ingrid Yusti | Kediri | 22011983 | 37 | P | AB | Kawin | Wiraswasta |

```

Gambar 6-12. Tampilan terminal variasi 3 saat menjalankan menu 6.

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 6

Mode Pencarian:
1. Pencarian di bagian NIK, Nama, Tempat Lahir, Tanggal Lahir, dan Pekerjaan
2. Pencarian di bagian Umur, Jenis Kelamin, Golongan Darah, dan Status

*ket: Opsi 1 pencarian di substring, Opsi 2 pecarian eksak
>>> 2

Keyword: 38
-----| No | NIK | Nama Lengkap | Tempat Lahir | Tanggal Lahir | Umur | Jenis Kelamin | Golongan Darah | Status | Pekerjaan |
-----| 4 | 1002861756 | Sofie Agustin | Bandung | 01031982 | 38 | P | AB | Tidak Kawin | PNS |
-----| 14 | 0902161017 | Scott Triswara | Banyuwangi | 20111981 | 38 | L | AB | Kawin | PNS |

```

Gambar 6-13. Tampilan terminal variasi 4 saat menjalankan menu 6.

```

Keyword: lmao
-----| No | NIK | Nama Lengkap | Tempat Lahir | Tanggal Lahir | Umur | Jenis Kelamin | Golongan Darah | Status | Pekerjaan |
-----= NOT FOUND =

```

Gambar 6-14. Tampilan terminal variasi 5 saat menjalankan menu 6.

6.9. MENGURUTKAN DATA

```

Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 7

Pengurutan berdasarkan:
1. NIK
2. Nama Lengkap
3. Tempat Lahir
4. Tanggal Lahir
5. Umur
6. Jenis Kelamin
7. Golongan Darah
8. Status
9. Pekerjaan
>>> 2
Data berhasil diurutkan berdasarkan parameter ke-2
  
```

Gambar 6-14. Tampilan terminal saat menjalankan menu 7.

Untuk mengurutkan data pada *database*, pengguna dapat memilih menu 7. Menu ini akan memberi pilihan ke pengguna mengenai parameter apa yang ingin diurutkan. Fitur pengurutan ini telah diuji dan bekerja pada semua parameter yang ada, namun hanya akan ditampilkan beberapa pada laporan ini. Pembaca dapat melakukan pengujian untuk parameter lainnya dengan *source code* yang telah dilampirkan bersamaan dengan laporan ini. Gambar 6-15 menampilkan tabel hasil pengurutan dengan parameter nama. Gambar 6-16 menampilkan tabel hasil pengurutan dengan parameter golongan darah. Dan gambar 6-17 menampilkan tabel hasil pengurutan dengan parameter NIK.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
2	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
3	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
4	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
5	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
6	1012040732	Eugene Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
7	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
8	0911960898	Inggrid Yusti	Kediri	22081983	37	P	AB	Kawin	Wiraswasta	
9	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
10	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
11	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	
12	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
13	1015119023	Michael Tambuan	Tegal	01071986	33	L	O	Kawin	Wiraswasta	
14	0987047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
15	0962161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
16	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
17	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
18	1024410187	Uda Wahab	Yogyakarta	17081989	38	L	AB	Tidak Kawin	Pegawai Swasta	
19	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	

Gambar 6-15. Tampilan tabel setelah diurutkan berdasarkan Nama.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	1012040732	Eugene Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
2	1030651494	Limbad Magician	Madiun	19101991	28	L	A	Kawin	Pegawai Swasta	
3	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
4	0911960898	Inggrid Yusti	Kediri	22081983	37	P	AB	Kawin	Wiraswasta	
5	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
6	0902161017	Scott Triswara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
7	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
8	1024410187	Uda Wahab	Yogyakarta	17081989	38	L	AB	Tidak Kawin	Pegawai Swasta	
9	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
10	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
11	9021867231	Ipin Coldman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
12	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
13	9031881174	Tamtam Sutami	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
14	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
15	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
16	0916900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
17	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Pegawai Swasta	
18	1015119023	Michael Tambuan	Tegal	01071986	33	L	O	Kawin	Wiraswasta	
19	9026860679	Upin Coldman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	

Gambar 6-16. Tampilan tabel setelah diurutkan berdasarkan Golongan Darah.

Tabel Data Penduduk Pada Sistem										
No	NIK	Nama Lengkap	Tempat Lahir	Tanggal Lahir	Umur	Jenis Kelamin	Golongan Darah	Status	Pekerjaan	
1	0982161017	Scott Triwara	Banyuwangi	20111981	38	L	AB	Kawin	PNS	
2	0907047723	Mike R. Silva	Tulungagung	21121982	37	L	B	Tidak Kawin	PNS	
3	0911960898	Ingrid Yusti	Kediri	22011983	37	P	AB	Kawin	Wiraswasta	
4	0999820635	Cintia Damanik	Bandar Lampung	01021981	39	P	B	Kawin	Pegawai Swasta	
5	1002861756	Sofie Agustin	Bandung	01031982	38	P	AB	Tidak Kawin	PNS	
6	1005912127	Elly Sugigi	Cirebon	01041983	37	P	B	Tidak Kawin	PNS	
7	1008971776	Bambang Sutikno	Sumedang	01051984	36	L	AB	Kawin	Wiraswasta	
8	1012640732	Eugine Batubara	Garut	01061985	34	L	A	Tidak Kawin	PNS	
9	1015119023	Michael Tambuan	Tejal	01071986	33	L	O	Kawin	Wiraswasta	
10	1018206677	Melanie Hughes	Tangerang Selatan	15061987	32	P	AB	Kawin	Wiraswasta	
11	1021303722	Butet Sayang	Pekalongan	16071988	31	P	O	Kawin	Pegawai Swasta	
12	1024410107	Uda Wahab	Yogyakarta	17081989	30	L	AB	Tidak Kawin	Pegawai Swasta	
13	1027526102	Ayu Lestari	Surabaya	18091990	29	P	O	Tidak Kawin	PNS	
14	1030651494	Limbad Magician	Madura	19101991	28	L	A	Kawin	Pegawai Swasta	
15	9016900686	Indri Marta	Purwokerto	23021984	36	P	O	Kawin	Pegawai Swasta	
16	9021867231	Ipin Colman	Cilacap	24031985	35	L	B	Tidak Kawin	PNS	
17	9026860679	Upin Colman	Cilacap	25041984	36	L	O	Tidak Kawin	PNS	
18	9031881174	Tamtam Sumati	Cepu	26021981	39	L	B	Kawin	Wiraswasta	
19	9036928864	Maya S. Katima	Batang	30011988	32	P	O	Kawin	Wiraswasta	

Gambar 6-17. Tampilan tabel setelah diurutkan berdasarkan NIK.

6.10. MENYIMPAN DATA KE FILE EKSTERNAL

```
Menu :
1. Input Data Manual
2. Input Data dari File Eksternal
3. Tampilkan Database
4. Edit Data
5. Hapus Data
6. Temukan Data
7. Urutkan Data
8. Simpan Data ke File Eksternal
9. Keluar
>>> 8

Overwrite File Input Terakhir (Y/N)?
>>> N
Masukkan nama File (tanpa extension): DataUpdate

Data berhasil tersimpan di csv dengan nama DataUpdate07052020.csv
```

Gambar 6-18. Tampilan terminal setelah menjalankan mode 8.

```
DataPenduduk31122019 - Notepad
File Edit Form View Help
No|NIK|NamaLengkap|TempatLahir|TanggalLahir|Umur|JenisKelamin|GolonganDarah|Status|Pekerjaan
1;0982161017;Scott Triwara;Banyuwangi;20111981;38;L;AB;Kawin;PNS
2;0907047723;Mike R. Silva;Tulungagung;21121982;37;L;B;Tidak Kawin;Pegawai Swasta
3;0911960898;Ingrid Yusti;Kediri;22011983;37;P;AB;Kawin;Pegawai Swasta
4;0999820635;Cintia Damanik;Bandar Lampung;01021981;39;P;B;Kawin;Pegawai Swasta
5;1002861756;Sofie Agustin;Bandung;01031982;37;P;AB;Tidak Kawin;PNS
6;1005912127;Elly Sugigi;Cirebon;01041983;36;P;B;Tidak Kawin;PNS
7;1008971776;Bambang Sutikno;Sumedang;01051984;35;L;AB;Kawin;Wiraswasta
8;1012640732;Eugine Batubara;Garut;01061985;34;L;P;Tidak Kawin;PNS
9;1015119023;Michael Tambuan;Tejal;01071986;33;L;P;AB;Kawin;Wiraswasta
10;1021303722;Butet Sayang;Pekalongan;16071988;31;P;O;Kawin;Pegawai Swasta
11;1024410107;Uda Wahab;Yogyakarta;17081989;30;L;AB;Tidak Kawin;Pegawai Swasta
12;1027526102;Ayu Lestari;Surabaya;18091990;29;P;O;Tidak Kawin;PNS
13;09021867231;Ipin Colman;Cilacap;24031985;34;L;B;Tidak Kawin;PNS
14;09026860679;Upin Colman;Cilacap;25041984;35;L;O;Tidak Kawin;PNS
15;09031881174;Tamtam Sumati;Cepu;26021981;38;L;B;Kawin;Wiraswasta
16;09036928864;Maya S. Katima;Batang;30011988;31;P;O;Kawin;Pegawai Swasta
```

Gambar 6-19. Tampilan notepad dari file awal.

```
DataPenduduk07052020 - Notepad
File Edit Form View Help
No|NIK|NamaLengkap|TempatLahir|TanggalLahir|Umur|JenisKelamin|GolonganDarah|Status|Pekerjaan
1;0982161017;Scott Triwara;Banyuwangi;20111981;38;L;AB;Kawin;PNS
2;0907047723;Mike R. Silva;Tulungagung;21121982;37;L;B;Tidak Kawin;Pegawai Swasta
3;0911960898;Ingrid Yusti;Kediri;22011983;37;P;AB;Kawin;Pegawai Swasta
4;0999820635;Cintia Damanik;Bandar Lampung;01021981;39;P;B;Kawin;Pegawai Swasta
5;1002861756;Sofie Agustin;Bandung;01031982;37;P;AB;Tidak Kawin;PNS
6;1005912127;Elly Sugigi;Cirebon;01041983;36;P;B;Tidak Kawin;PNS
7;1008971776;Bambang Sutikno;Sumedang;01051984;35;L;AB;Kawin;Wiraswasta
8;1012640732;Eugine Batubara;Garut;01061985;34;L;P;Tidak Kawin;PNS
9;1015119023;Michael Tambuan;Tejal;01071986;33;L;P;AB;Kawin;Wiraswasta
10;1021303722;Butet Sayang;Pekalongan;16071988;31;P;O;Kawin;Pegawai Swasta
11;1024410107;Uda Wahab;Yogyakarta;17081989;30;L;AB;Tidak Kawin;Pegawai Swasta
12;1027526102;Ayu Lestari;Surabaya;18091990;29;P;O;Tidak Kawin;PNS
13;09021867231;Ipin Colman;Cilacap;24031985;34;L;B;Tidak Kawin;PNS
14;09026860679;Upin Colman;Cilacap;25041984;35;L;O;Tidak Kawin;PNS
15;09031881174;Tamtam Sumati;Cepu;26021981;38;L;B;Kawin;Wiraswasta
16;09036928864;Maya S. Katima;Batang;30011988;31;P;O;Kawin;Pegawai Swasta
```

Gambar 6-19b. Tampilan notepad dari file yang disave oleh program.

Menu terakhir ini dapat melakukan *exporting* data yang terdapat di program yang sudah diedit saat *runtime*, ke csv sesuai nama yang ditentukan oleh pengguna. Program akan meng-*generate* file pada work directory yang sama seperti letak kode utama nya. Program juga akan memberi pilihan pada pengguna apakah ingin dilakukan *overwrite* atau mengenerate file baru. Dapat dilihat pada gambar 6-19 dan 6-19b bahwa terdapat perbedaan data seperti pada indeks 14. Meskipun perbedaan ini sedikit, namun dapat dikatakan cukup untuk menguji menu ini, karena apabila pengguna melakukan *bulk editing* pada program, program juga tetap dapat menyimpan perubahan yang dilakukan tanpa terkecuali.

7. KESIMPULAN

1. Telah berhasil dibuat program *database* kependudukan sederhana dengan bahasa C dan struktur data *static array* dengan asumsi awal yang telah ditentukan.
2. Program memiliki kelebihan berupa sistem penanggalan otomatis dengan bantuan *time.h*, serta dapat menampilkan tabel database dengan baik, mengikuti ukuran text pada tiap *cell* secara dinamis.
3. Program memiliki kekurangan berupa implementasi pemrograman yang

kurang efisien. Apabila diinginkan time complexity yang rendah serta pengaplikasian untuk jumlah data yang besar, program lebih baik menggunakan struktur data *linked list*.

4. Sebagian besar fungsi memiliki kompleksitas waktu $O(n)$. Namun terdapat dua fungsi yang memiliki kompleksitas waktu $O(n^2)$, yaitu fungsi sortData dan showData. Terdapat pula beberapa fungsi yang memiliki kompleksitas waktu $O(1)$ yaitu fungsi editData, initDate, dan countAge.
5. Pengujian pada program ini membuat hasil yang baik, dimana pengujian function (*function test*) dan *unit test* dapat berjalan sesuai spesifikasi yang telah dijabarkan pada bab 1. Hanya saja, belum direalisasikan suatu validator yang dapat menguji kebenaran input pengguna, khususnya pada insersi data tabel. Namun sudah diberi validasi ke mode navigasi program untuk memudahkan pengguna.