

UTS EL2008 - Pemecahan Masalah dengan C

Nama: Michelle Angelina

NIM: 18320007

1. Diberikan suatu spesifikasi representasi integer dalam bentuk 16 digit bit.

- a. Berapa nilai tertinggi serta terendah yang dapat direpresentasikan spesifikasi tersebut baik sebagai unsigned maupun 2's-complement?

$$2^{16} = 65.536$$

Integer berukuran 16 digit bit yang unsigned dapat menyimpan nilai dengan rentang 0 sampai 65.535, sedangkan integer berbentuk 2's-complement dapat menyimpan nilai dengan rentang -32.768 sampai 32.767.

- b. Ambil nilai digit kedua dan ketiga dari NIM Anda, kemudian gabungkan dengan tiga digit terakhir dari NIM Anda (misal: 13207021 -> 32021). Nyatakan bilangan tersebut sebagai suatu bilangan desimal dan representasikan dalam biner

NIM: 18320007, digit kedua adalah 8 dan digit ketiga adalah 3, sehingga menghasilkan angka 83007. Bentuk bilangan desimal adalah 83007.

Jika dikonversikan menjadi bilangan biner: 1 0100 0100 0011 1111,

sedangkan untuk 2's Complement menghasilkan:
0000 0000 0000 0001 0100 0100 0011 1111.

- c. Ambil least significant byte dari nilai di atas dan simpan sebagai suatu variabel A.

Bilangan biner NIM: 1 0100 0100 0011 1111

1's complement-nya menjadi: 0 1011 1011 1100 0000,

2's complement-nya dibuat dengan menambah digit 1 ke 1's complement menjadi:
0 1011 1011 1100 0001,

Maka LSB dari nilai tersebut adalah 1100 0001, sehingga variabel A = 1100 0001.

- d. Kemudian lakukan perhitungan di bawah ini. Notasi operator diberikan sesuai penamaan dalam bahasa C. Hasil memuat 8-bit bilangan biner. x dan y diinisiasi sebagai signed. Nyatakan interpretasi hasilnya dalam bentuk biner, heksadesimal, *unsigned*, serta *2's complement*!

x	y	Operasi	Binary	Hexa	Unsigned	Signed 2C
1010 0101	1110 1111	$(A \& y)^x$	0110 0100	64	0000 0000 0110 0101	1001 1010
1101 1011	1111 0011	$(\sim(y \gg 4) \& x) + A$	1100 0001	C1	0000 0000 1100 0001	0011 1111
0010 1111	1010 0101	$((x \gg 2) y) * A$	1100 0001	C1	0000 0000 1100 0001	0011 1111

$$1) (1100\ 0001 \& 1110\ 1111)^{(1010\ 0101)} = (1100\ 0001)^{(1010\ 0101)} = 0110\ 0100$$

$$2) (\sim(1111\ 0011 \gg 4) \& 1101\ 1011) + 1100\ 0001 = (\sim(0011\ 1111) \& (1101\ 1011)) + 1100\ 0001 = 0 + 1100\ 0001 = 1100\ 0001$$

$$3) ((0010\ 1111 \gg 2) | 1010\ 0101) * 1100\ 0001 = (1100\ 1011) | (1010\ 0101) * 1100\ 0001 = 1110\ 1111 * 1100\ 0001 = 1100\ 0001$$

- e. Buat program dalam Bahasa C, yang menerima input seperti pada b, dan menghasilkan nilai-nilai pada d.

Source code:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>

int binaryconvert(int decimal) {

// REFERENSI:
https://www.javatpoint.com/c-program-to-convert-decimal-to-binary

    int binary[17]; // Biner yang digunakan adalah biner 16 bit
    int i;

    for (i = 0; decimal > 0; i++) {

        binary[i] = decimal % 2;

// Pengurangan angka karena basenya 2

        decimal /= 2;
    }
}
```

```

    }

    // Output
    for(i = i-1; i >= 0; i--) {
        printf("%d", binary[i]);
    }

    return binary[17];
}

int hexaconvert(long long decimal) {

                                                                    // REFERENSI:
https://www.sanfoundry.com/c-program-convert-decimal-hex/

    long long hasilbagi;
    long sisa;
    char hex[10];
    int j;

    hasilbagi = decimal;

    while (hasilbagi != 0) {
        sisa = hasilbagi % 16; // Basis 16
        int j;

        if (sisa < 10) {
            hex[j] = 48 + sisa;
            ++j;
        }
        else {
            hex[j] = 55 + sisa;
            hasilbagi /= 16;
        }
    }
}

```

```

        ++j;
    }
}

// Output
for (int i = j; i >= 0; i--) {
    printf("%c", hex[i]);
}

return hex;
}

int decimalconvert(int binary) {
    int dec = 0;
    int remain = 0;
    int i;

    while (binary != 0) {
        remain = binary % 10;
        binary /= 10;
        dec += remain * pow(2, i);
        i++;
    }

    return dec;
}

int main() {
    int digit[17];
    int a;

```

```

int i;

// Input nilai

printf("Masukkan nilai digit kedua dan ketiga NIM Anda\nKemudian
gabungkan tiga digit terakhir dari NIM Anda (misal: 13207021 ->
32021): ");

scanf("%d",&digit);

// Konversi

printf("\nHasil konversi desimal ke biner: ");

binaryconvert(digit);

// Konversi ke 1's complement

for (i = 0; i < 17; i++) {
    if (digit[i] == 0) {
        digit[i] = 1;
    }
    else {
        digit[i] = 0;
    }
}

int carry = 1; // Penambahan 1 ke biner

// Konversi ke 2's complement

for (i = 0; i < 17; i++) {
    if (digit[i] == 1 && carry == 1) {
        digit[i] = 0;
        carry = 1;
    }
    else if (digit[i] == 0 && carry == 1) {
        digit[i] = 1;
    }
}

```

```

        carry = 0;
    }

    else {
        digit[i] = digit[i];
    }
}

printf("\nHasil konversi ke 2's Complement: %d", digit);

// Mencari variabel A, a.k.a Least Significant Byte = 8 bit
for (a = 7; a >= 0; a--) {
    a = a + (digit[a]*pow(10, a));
}

int deca = 0;

deca = decimalconvert(a);

// Operasi biner
int x1, y1, x2, y2, x3, y3;

x1 = 193; //10100101
y1 = 239; // 11101111
x2 = 219; // 11011011
y2 = 243; // 11110011
x3 = 47; // 00101111
y3 = 165; // 10100101

int op1;
int op2;
int op3;

```

```

op1 = (deca & y1) ^ x1;
op2 = (~(y2 >> 4) && 2) + deca;
op3 = ((x3 >> 2) | y3) * deca;

int pil = 0;

// Pilihan menu untuk output tabel
printf("\n\n1. (A&y)^x");
printf("\n2. ~(y>>4) &&x+A");
printf("\n3. ((x>>2)|y)*A");
printf("\nPilih jenis operasi (1/2/3): ");
scanf("%d", &pil);

if (pil == 1) {
    printf("Hasil (A&y)^x: %d", binaryconvert(op1));
}

else if (pil == 2) {
    printf("Hasil ~(y>>4) &&x+A: %d", binaryconvert(op2));
}

else if (pil == 3) {
    printf("Hasil ((x>>2)|y)*A: %d", binaryconvert(op3));
}

else {
    printf("Pilihan tidak valid.");
}

printf("\n");

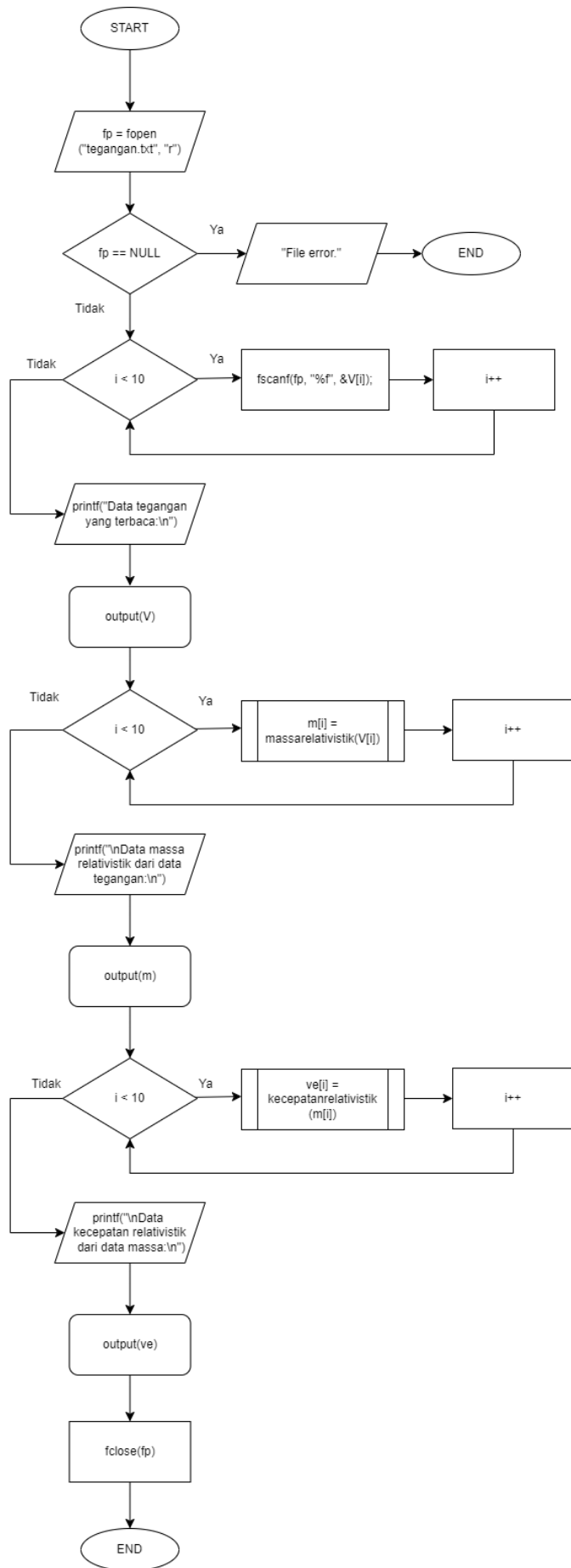
```

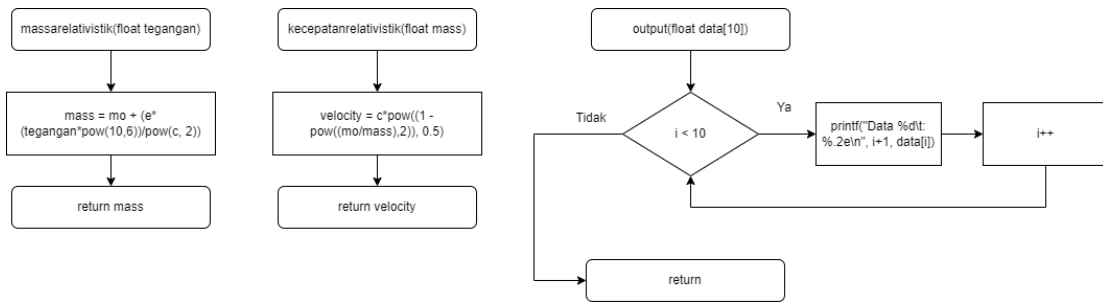
```
    return 0;  
}
```

f. **Bandingkan hasilnya.**

Seharusnya hasil yang ditampilkan oleh program dengan perhitungan manual akan memberikan hasil yang sama, menggunakan bitwise operator.

2. Sebuah elektron dipercepat oleh tegangan V dalam sebuah electron gun memperoleh energi $V_e = mc^2 - m_0c^2$, di mana 1.602×10^{-19} coulomb adalah muatan elektron, $m_0 = 9.109 \times 10^{-31}$ kg adalah massa diam, m adalah massa relativistik dalam kg, dan $c = 2.9979 \times 10^8$ m/s adalah kecepatan cahaya. Kecepatan elektron v bermassa relativistik m diperoleh dari $m/m_0 = [1 - (\frac{v}{c})^2]^{-1/2}$
- a. Buatlah flowchart dan program dalam Bahasa C yang membaca beberapa tegangan dari file eksternal dan menghitung massa relativistik dan kecepatan elektron yang dipercepat oleh tegangan tersebut.





Source Code:

```
#include <stdio.h>

#include <stdlib.h>

#include <math.h>
```

```
#define e 1.602*pow(10,-19)

#define c 2.9979*pow(10,8)

#define mo 9.109*pow(10,-31)
```

```
float massarelativistik(float tegangan) {

    // V = mc^2 - moc^2,
    // maka V = c^2 (m-mo)
    // sehingga m = mo + V/c^2

    float mass;

    mass = mo + (e*(tegangan*pow(10,6))/pow(c, 2));

    return mass;

}
```

```
float kecepatanrelativistik(float mass) {

    // m/mo = (1-(v/c)^2)^(-1/2)
    // m/mo = 1/(1-(v/c)^2)^(1/2)
    // m^2/mo^2 = 1/(1-(v/c)^2)
    // m^2 * (1-(v/c)^2) = mo^2
    // (1-(v/c)^2) = (mo/m)^2
    // 1 - (mo/m)^2 = (v/c)^2
    // c^2 * (1 - (mo/m)^2) = v^2
```

```

//  $c(1 - (m_0/m)^2)^{1/2} = v$ 

float velocity;

velocity = c*pow((1 - pow((m0/mass),2)), 0.5);

return velocity;
}

void output(float data[10]) {
    for (int i = 0; i < 10; i++) {
        printf("Data %d\t: %.2e\n", i+1, data[i]);
    }
}

int main() {
    float ve[10]; // Array of kecepatan, 10 data
    float m[10]; // Array of mass, 10 data
    float V[10]; // Array of tegangan, 10 data
    int i;

    // REFERENSI: Praktikum EL2208 Modul 02 Praktikum Pemecahan
    Masalah dengan C

    FILE* fp;

    fp = fopen("tegangan.txt", "r");

    if (fp == NULL) {
        printf("File error.");
        return 0;
    }

    for (i=0; i<10; i++) {
        fscanf(fp, "%f", &V[i]);
    }
}

```

```

    }

    // Print tegangan dari tegangan.txt
    printf("Data tegangan yang terbaca:\n");
    output(V);

    // Menghitung massa relativistik
    for (i = 0; i < 10; i++) {
        m[i] = massarelativistik(V[i]);
    }

    // Print massa relativistik
    printf("\nData massa relativistik dari data tegangan:\n");
    output(m);

    // Menghitung kecepatan relativistik
    for (i = 0; i < 10; i++) {
        ve[i] = kecepatanrelativistik(m[i]);
    }

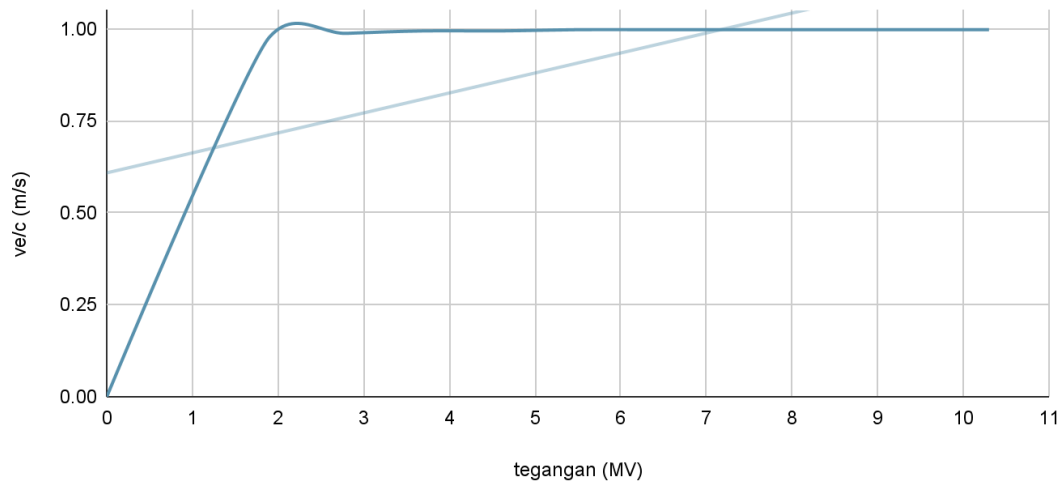
    // Print kecepatan relativistik
    printf("\nData kecepatan relativistik dari data massa:\n");
    output(ve);

    fclose(fp);
    return 0;
}

```

- b. Lakukan plot antara [kecepatan/kecepatan cahaya] (sumbu y) dan [tegangan dalam juta] (sumbu x)

Plot



- c. Lakukan untuk 10 data (beri nama file eksternal tegangan.txt)

```

2.c - Codes:Blocks  "C:\Users\Cello\OneDrive - Institut Teknologi Bandung\Documents\Pemecahan Masalah dengan C\UTSno2.exe"
View Search
Help
Data tegangan yang terbaca:
Data 1 : 1.90e+000
Data 2 : 2.80e+000
Data 3 : 3.70e+000
Data 4 : 4.60e+000
Data 5 : 5.50e+000
Data 6 : 6.40e+000
Data 7 : 7.30e+000
Data 8 : 8.20e+000
Data 9 : 9.10e+000
Data 10 : 1.03e+001

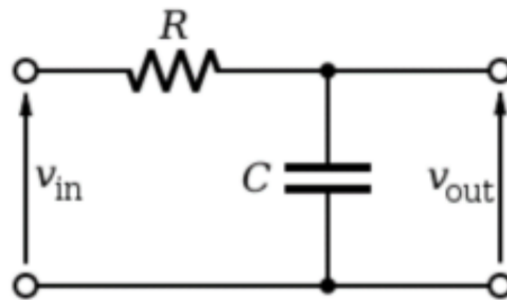
Data massa relativistik dari data tegangan:
Data 1 : 4.30e-030
Data 2 : 5.90e-030
Data 3 : 7.51e-030
Data 4 : 9.11e-030
Data 5 : 1.07e-029
Data 6 : 1.23e-029
Data 7 : 1.39e-029
Data 8 : 1.55e-029
Data 9 : 1.71e-029
Data 10 : 1.93e-029

Data kecepatan relativistik dari data massa:
Data 1 : 2.93e+008
Data 2 : 2.96e+008
Data 3 : 2.98e+008
Data 4 : 2.98e+008
Data 5 : 2.99e+008
Data 6 : 2.99e+008
Data 7 : 2.99e+008
Data 8 : 2.99e+008
Data 9 : 2.99e+008
Data 10 : 2.99e+008
  
```

```

TP04program.c x tegangan.txt x new 1 x UTSno2.c x
1 1.9
2 2.8
3 3.7
4 4.6
5 5.5
6 6.4
7 7.3
8 8.2
9 9.1
10 10.3
  
```

3. Diketahui sebuah rangkaian penapis (filter) RC seperti gambar di bawah ini:



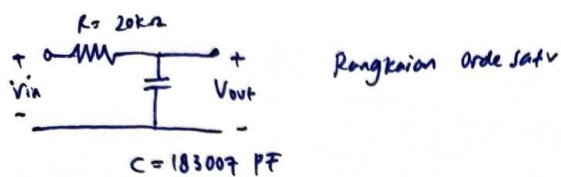
Apabila nilai R dan C didapatkan dari NIM anda (contoh NIM : XXXYYZZZ)

$R = YY \text{ k}\Omega$, $C = XXXZZZ \text{ pF}$, $V_{in} = 5 \text{ Volt}$

a. Turunkan persamaan fungsi respon (transien dan tunak) tegangan output (V_{out}) dari filter tersebut dan analisis. Jelaskan.

NIM: 183 20 007

$R = 20 \text{ k}\Omega$, $C = 183007 \text{ pF}$



Dengan menggunakan KCL pada loop Kiri

$V_{in} = I_C R + V_C$, karena V_{out} adalah kabel terbuka, $I_C = I_R$

$I_C = I_R$

$$V_{in} = R \cdot C \frac{dV_C}{dt} + V_C$$

menganalisis pada saat $t = \infty$ dan saat $t < 0$

pada saat keadaan tunak, $V_C(\infty) = V_{in} = 5 \text{ V}$ (open circuit)

saat $t = 0$, $V_C(0) = 0 \text{ V}$ (short circuit)

Hambatan ekuivalen, $R_{th} = R = 20 \text{ k}\Omega$

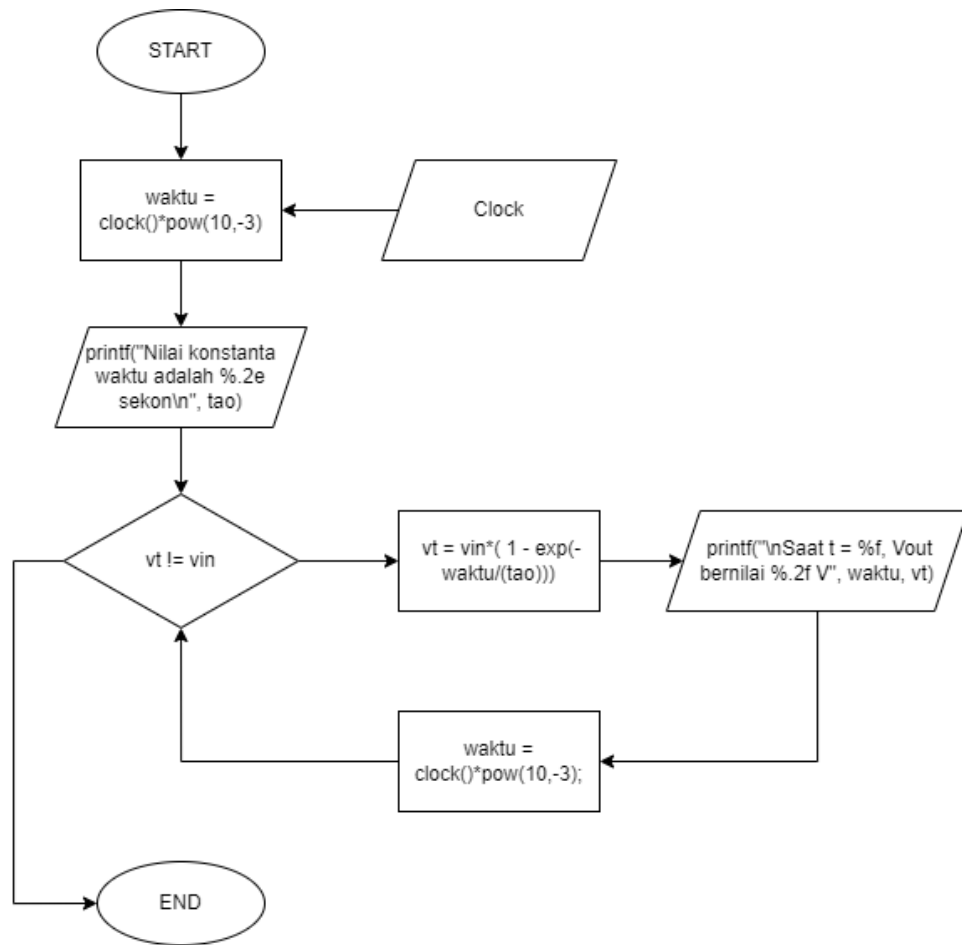
maka konstanta waktu, $\tau = RC = 183007 \cdot 10^{-12} \cdot 20 \cdot 10^3 = 3,66 \cdot 10^{-3}$

Jadi, persamaan rangkaian orde satu, $\rightarrow x(t) = x(\infty) + (x(0) - x(\infty))e^{-t/\tau}$

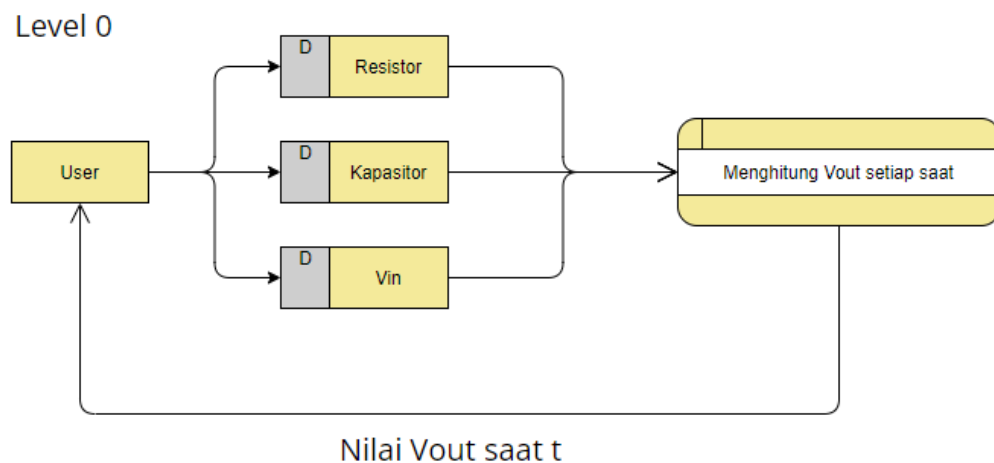
$$V(t) = 5 + (0 - 5)e^{-t/3,66 \cdot 10^{-3}}$$

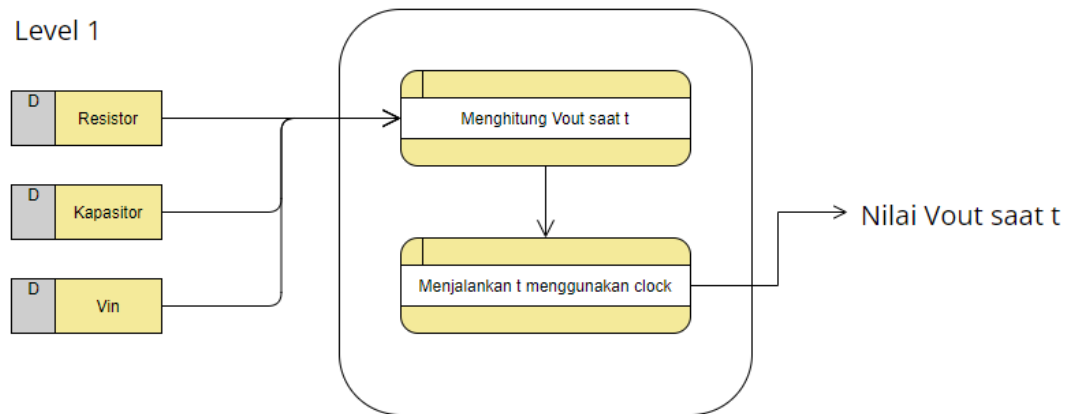
$$V(t) = 5(1 - e^{-0,2732 t \cdot 10^3})$$

- b. Berdasarkan dari persamaan (a), buat diagram alir (flowchart) program yang dapat digunakan menghitung Vout tiap saat sampai dengan kondisi tunak



- c. Berdasarkan dari persamaan (a), buat DFD dari yang dapat digunakan menghitung Vout tiap saat sampai dengan kondisi tunak





d. Buat program tersebut dalam bahasa C

Source Code:

```

#include <stdio.h>

#include <stdlib.h>

#include <time.h>

#include <math.h>


#define r 20*pow(10,3)

#define c 183007*pow(10,-12)

#define vin 5

#define tao r*c


int main() {

// REFERENSI:
https://www.tutorialspoint.com/c\_standard\_library/c\_function\_clock.htm

// REFERENSI:
https://www.ibm.com/docs/en/zos/2.3.0?topic=functions-clock-determine-processor-time

    double waktu;

    waktu = clock()*pow(10,-3); // Milisekon


    printf("Nilai konstanta waktu adalah %.2e sekon\n", tao);

```

```

double vt;

while (vt != vin){

    vt = vin*( 1 - exp(-waktu/(tao)));

    printf("\nSaat t = %f s, Vout bernilai %.2f V", waktu, vt);

    // Supaya waktunya bergerak harus dipanggil terus menerus

    waktu = clock()*pow(10,-3);

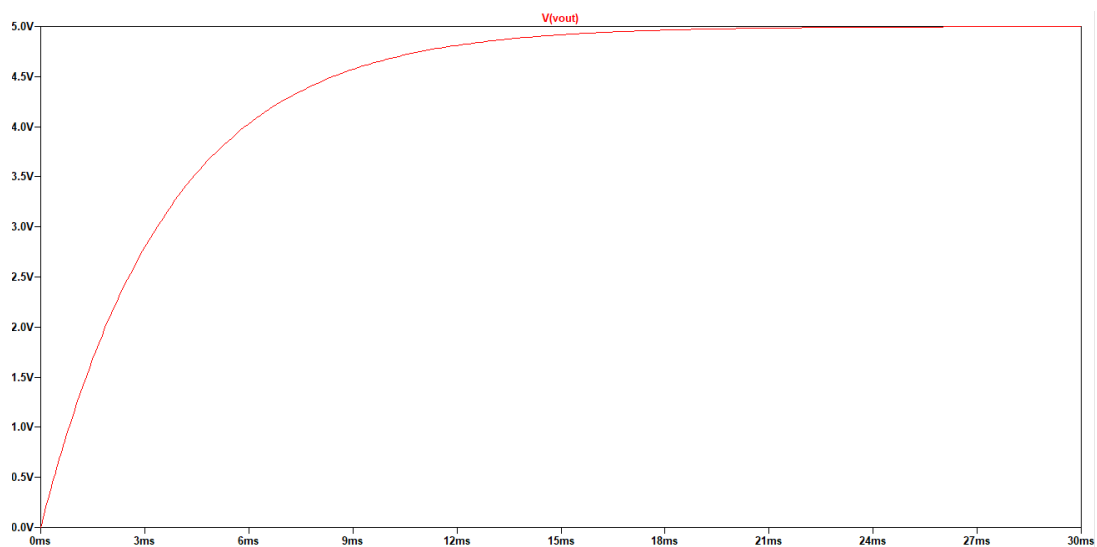
}

return 0;

}

```

e. Buat grafik Vout terhadap waktu (t)



f. Berapa konstanta waktu yang dihasilkan dari persamaan (a)? Apakah ada perbedaan dengan konstanta waktu yang dihasilkan oleh program? Jelaskan.

Tidak ada, karena rumus konstanta waktu tetap $R \cdot C$, dengan nilai yang sudah ditentukan dari soal. Konstanta waktu tidak bergantung pada waktu dan bersifat konstan jika variabel resistor dan kapasitor terus konstan.

Konstanta waktu yang didapat bernilai 3.66×10^{-3} .

4. Sebuah kecamatan di Indonesia sedang melakukan migrasi sistem pencatatan data kependudukan; yaitu dari sistem pencatatan data secara manual (tulisan) menjadi sistem pencatatan data berbasis komputer. Secara umum, data penduduk yang akan direkam pada sistem pencatatan data kependudukan memiliki parameter data sebagai berikut:

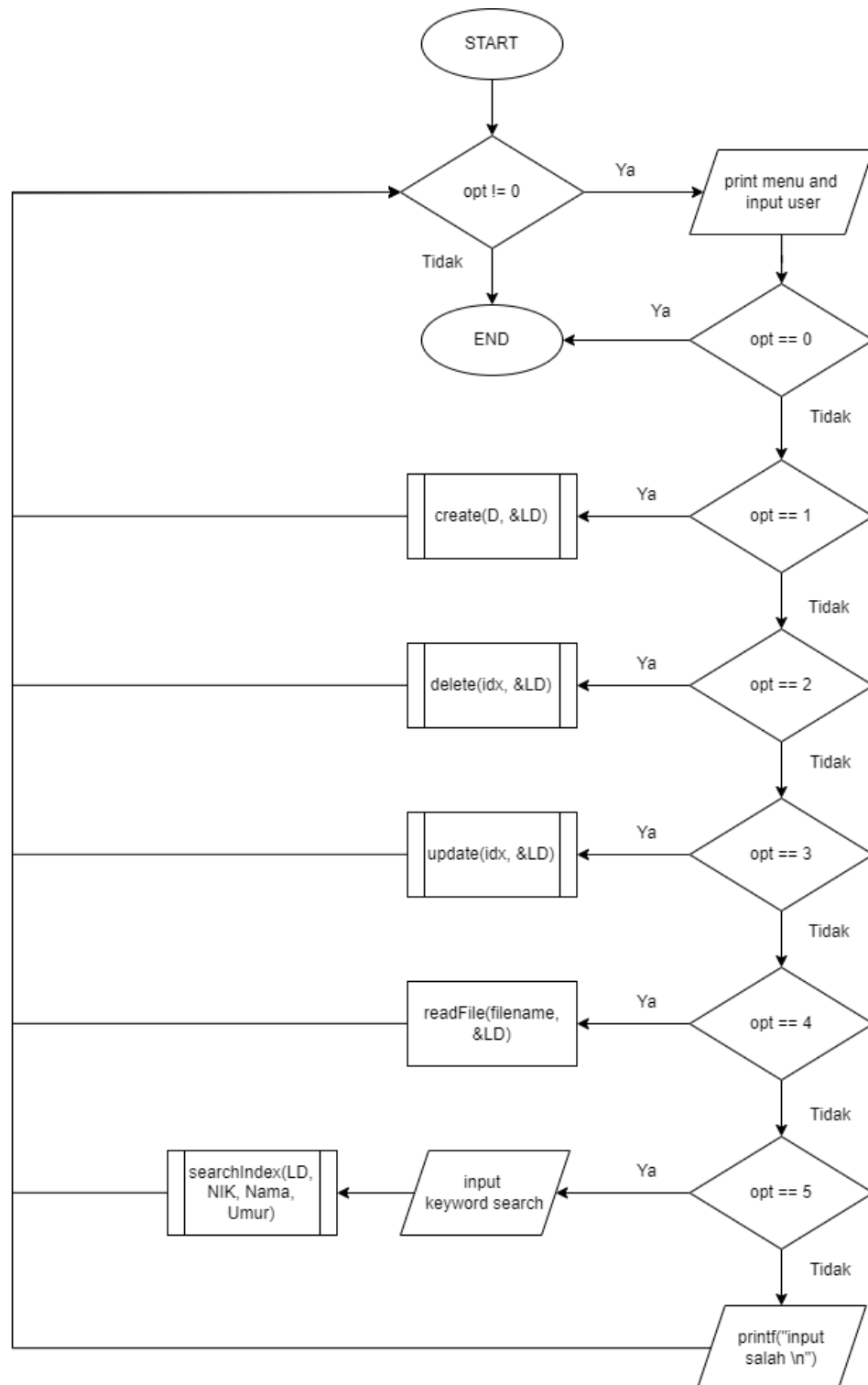
- | | |
|-----------------------------------|----------------------|
| a. Nomor Induk Kependudukan (NIK) | f. Jenis Kelamin |
| b. Nama Lengkap | g. Golongan Darah |
| c. Tempat Lahir | h. Status Perkawinan |
| d. Tanggal Lahir | i. Pekerjaan |
| e. Umur | |

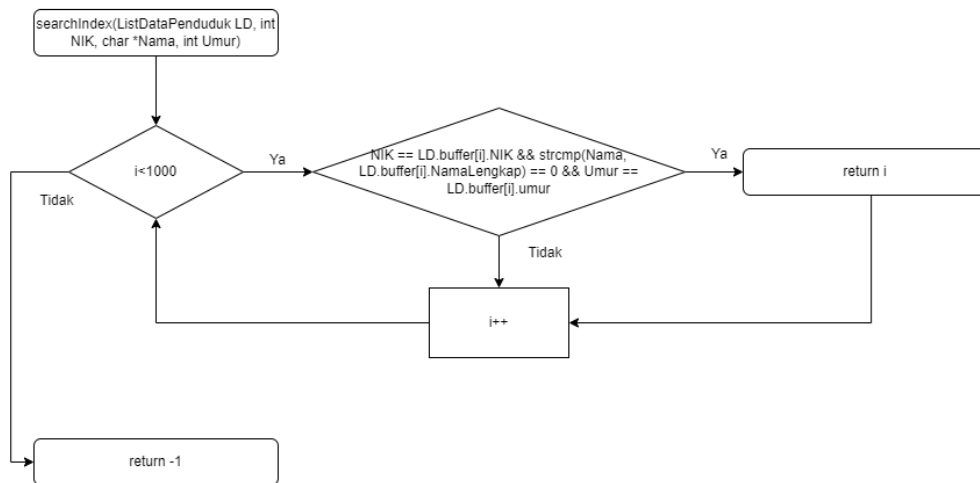
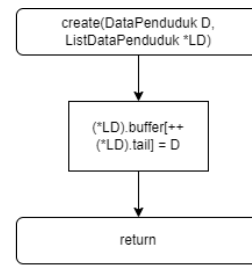
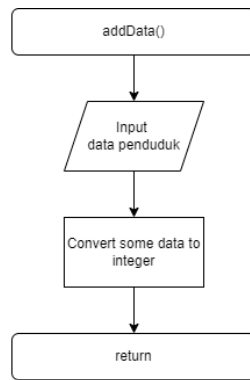
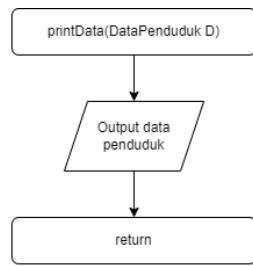
Sebagaimana sistem pencatatan pada umumnya, sistem pencatatan data berbasis komputer ini memiliki fitur utama sebagai berikut:

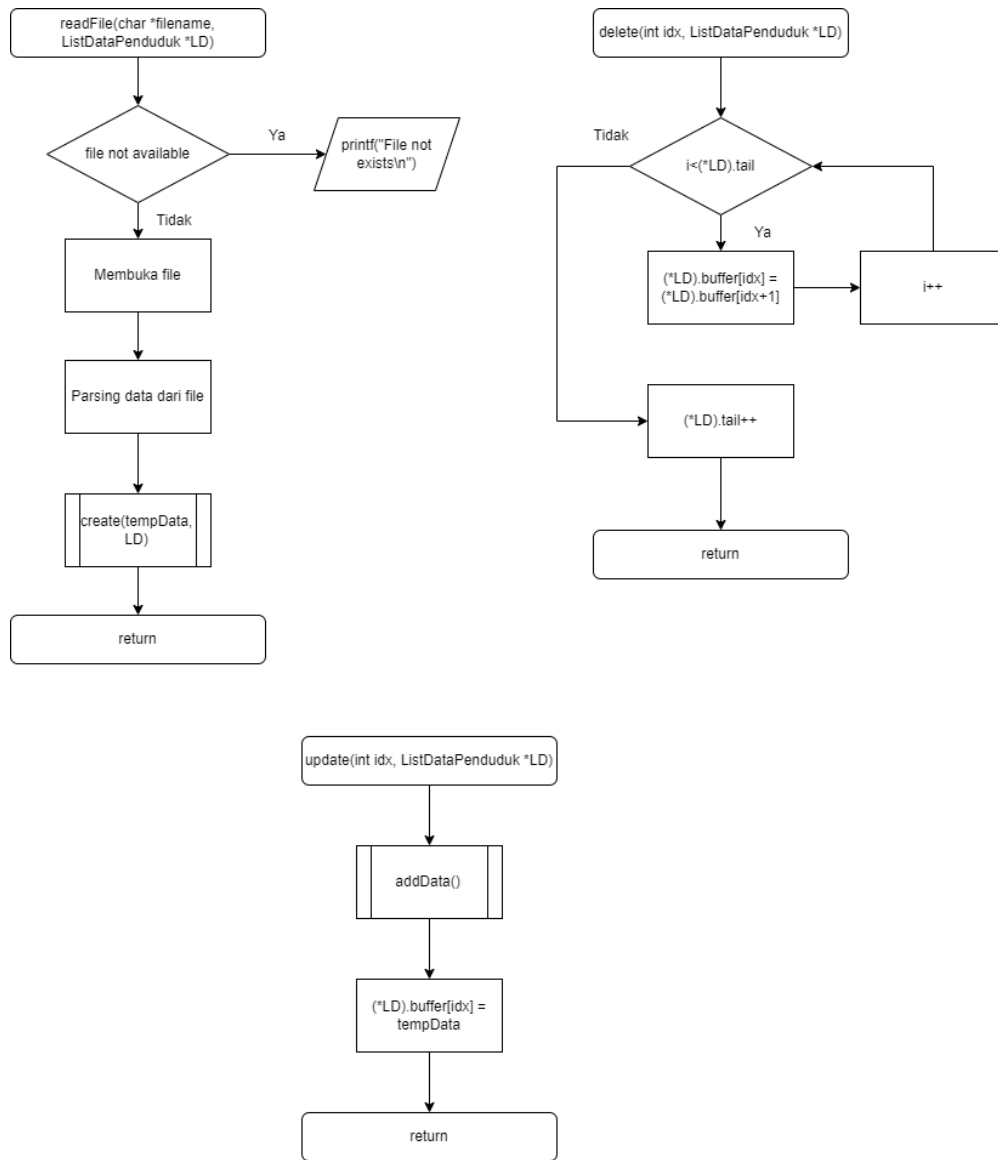
- a. memiliki fasilitas membuat (create), mengubah (update), menghapus (delete) data
- b. memiliki fasilitas import data dari file external
- c. memiliki fasilitas pencarian (searching) pada parameter NIK, Nama dan Umur

Berdasarkan hal tersebut,

- a. Buatlah flowchart aplikasi sistem pencatatan data kependudukan yang dapat diimplementasikan di kecamatan tersebut. Jelaskan.

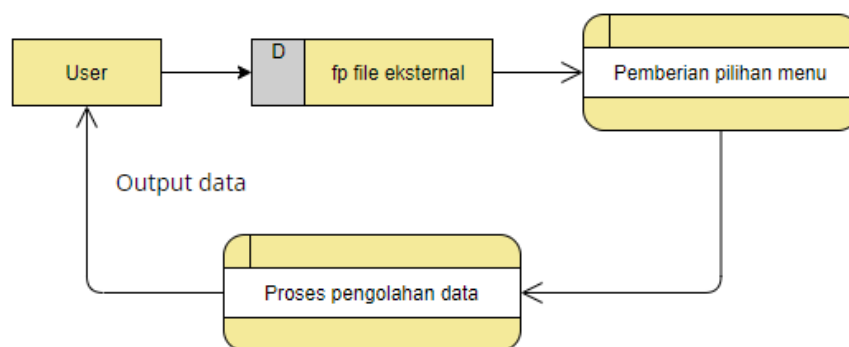






b. Buatlah DFD dari aplikasi tersebut. Jelaskan.

Level 0



c. Buatlah aplikasi tersebut menggunakan bahasa C

Source code:

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct DataPenduduk {

    int NIK;

    char NamaLengkap[50];

    char TempatLahir[50];

    int TanggalLahir;

    int umur;

    char JenisKelamin[50];

    char GolonganDarah[50];

    char StatusPerkawinan[50];

    char Pekerjaan[50];

} DataPenduduk;

typedef struct ListDataPenduduk {

    DataPenduduk buffer[1000];

    int tail;

} ListDataPenduduk;

void printData(DataPenduduk D) {

    printf("\n---Data Penduduk---\n");

    printf("NIK: %d\n", D.NIK);

    printf("Nama Lengkap: %s\n", D.NamaLengkap);

    printf("tempat lahir: %s\n", D.TempatLahir);

    printf("Tanggal Lahir (DDMMYYYY): %d\n", D.TanggalLahir);

    printf("Umur: %d\n", D.umur);
```

```

printf("Jenis Kelamin: %s\n", D.JenisKelamin);

printf("Golongan Darah: %s\n", D.GolonganDarah);

printf("Status Perkawinan: %s\n", D.StatusPerkawinan);

printf("Pekerjaan: %s\n", D.Pekerjaan);

}

```

```

DataPenduduk addData() {

    DataPenduduk tempData;

    char tempNIK[50];

    char tempNama[50];

    char tempTempatLahir[50];

    char tempTanggalLahir[50];

    char tempUmur[50];

    char tempjenisKelamin[50];

    char tempGolonganDarah[50];

    char tempStatusPerkawinan[50];

    char tempPekerjaan[50];


    printf("NIK: ");

    scanf(" %[^\\n]s", &tempNIK);

    // printf("%d", atoi(tempNIK));

    printf("Nama Lengkap: ");

    scanf(" %[^\\n]s", &tempNama);

    printf("Tempat lahir: ");

    scanf(" %[^\\n]s", &tempTempatLahir);

    printf("Tanggal Lahir (DD/MM/YY): ");

    scanf(" %[^\\n]s", &tempTanggalLahir);

    printf("Umur: ");

```



```

scanf(" %[^\\n]s", &tempUmur);

printf("Jenis Kelamin: ");

scanf(" %[^\\n]s", &tempjenisKelamin);

printf("Golongan Darah: ");

scanf(" %[^\\n]s", &tempGolonganDarah);

printf("Status Perkawinan: ");

scanf(" %[^\\n]s", &tempStatusPerkawinan);

printf("Pekerjaan: ");

scanf(" %[^\\n]s", &tempPekerjaan);


tempData.NIK = atoi(tempNIK);

// printf("%d", atoi(tempNIK));

strcpy(tempData>NamaLengkap, tempNama);

strcpy(tempData.TempatLahir, tempTempatLahir);

tempData.TanggalLahir = atoi(tempTanggalLahir);

strcpy(tempData.JenisKelamin, tempjenisKelamin);

strcpy(tempData.GolonganDarah, tempGolonganDarah);

strcpy(tempData.StatusPerkawinan, tempStatusPerkawinan);

strcpy(tempData.Pekerjaan, tempPekerjaan);

tempData.umur = atoi(tempUmur);


// printf("----");

// printData(tempData);

// printf("Nama Lengkap: %s\\n", tempData>NamaLengkap);

// printf("tempat lahir: %s\\n", tempData.TempatLahir);

// printf("Tanggal Lahir (DD/MM/YY): %s\\n", tempData.TanggalLahir);

// printf("Umur: %d\\n", tempData.umur);

// printf("Jenis Kelamin: %s\\n", tempData.JenisKelamin);

// printf("Golongan Darah: %s\\n", tempData.GolonganDarah);

// printf("Status Perkawinan: %s\\n", tempData.StatusPerkawinan);

```

```

        // printf("Pekerjaan: %s\n", tempData.Pekerjaan);

        return tempData;

    }

void create(DataPenduduk D, ListDataPenduduk *LD) {
    (*LD).buffer[++(*LD).tail] = D;
}

int searchIndex(ListDataPenduduk LD, int NIK, char *Nama, int Umur) {
    for (int i=0; i<1000; i++) {
        if (NIK == LD.buffer[i].NIK && strcmp(Nama,
LD.buffer[i].NamaLengkap) == 0 && Umur == LD.buffer[i].umur) {
            return i;
        }
    }
    return -1;
}

void readFile(char *filename, ListDataPenduduk *LD) {
    FILE* fp = fopen(filename, "r");

    if (!fp) {
        printf("File not exists\n");
    }
    else {
        {
            DataPenduduk tempData;

```

```

char buffer[1024];

int row = 0;

int column = 0;

while (fgets(buffer,

            1024, fp)) {

    column = 0;

    row++;

    if (row == 1)

        continue;

    // Memisahkan data dari file eksternal

    char* value = strtok(buffer, ";");

    while (value) {

        // Column 1

        if (column == 0) {

            }

            if (column == 1) {

                tempData.NIK = atoi(value);

            }

            // Column 2

            if (column == 2) {

                strcpy(tempData>NamaLengkap, value);

            }

```

```

// Column 3

if (column == 3) {
    strcpy(tempData.TempatLahir, value);
}

if (column == 4) {
    tempData.TanggalLahir = atoi(value);
}

if (column == 5) {
    tempData.umur = atoi(value);
}

if (column == 6) {
    strcpy(tempData.JenisKelamin, value);
}

if (column == 7) {
    strcpy(tempData.GolonganDarah, value);
}

if (column == 8) {
    strcpy(tempData.StatusPerkawinan, value);
}

if (column == 9) {
    strcpy(tempData.Pekerjaan, value);
}

value = strtok(NULL, ";");
column++;
}

create(tempData, LD);

printf("\n");
}

```

```

        // Close file

        fclose(fp);

    }

}

void delete(int idx, ListDataPenduduk *LD) {

    for (int i=idx; i<(*LD).tail; i++) {

        (*LD).buffer[idx] = (*LD).buffer[idx+1];

    }

    (*LD).tail++;

}

void update(int idx, ListDataPenduduk *LD) {

    DataPenduduk tempData = addData();

    (*LD).buffer[idx] = tempData;

}


int main() {

    DataPenduduk D;

    ListDataPenduduk LD;

    LD.tail = -1;

    int opt = -1;

    while (opt != 0) {

        printf("\nSelamat datang di Data Kependudukan\n");

        printf("Input fitur:\n");

        printf("0. Keluar \n");
    }
}

```

```

printf("1. Tambahkan data(create)\n");

printf("2. Hapus data(delete)\n");

printf("3. Update data(update)\n");

printf("4. Import data(csv)\n");

printf("5. Pencarian data(search)\n");

scanf(" %d", &opt);

if (opt == 0) {

    printf("Keluar\n");

}

else if (opt == 1) {

    printf("Tambahkan Data Baru\n");

    D = addData();

    create(D, &LD);

}

else if (opt == 2) {

    printf("Hapus Data\n");

    printf("Masukkan data\n");

    int idx;

    int NIK;

    char Nama[50];

    int Umur;

    //input;

    printf("NIK: ");

    scanf("%d", &NIK);

    printf("Nama Lengkap: ");

    scanf(" %[^\n]s", &Nama);

    printf("Umur: ");

    scanf("%d", &Umur);

```

```

        idx = searchIndex(LD, NIK, Nama, Umur);

        delete(idx, &LD);
    }

    else if (opt == 3) {
        printf("Update Data\n");
        printf("Masukkan data lama\n");

        int idx;

        int NIK;

        char Nama[50];

        int Umur;

        //input;

        printf("NIK: ");

        scanf("%d", &NIK);

        printf("Nama Lengkap: ");

        scanf(" %[^\\n]s", &Nama);

        printf("Umur: ");

        scanf("%d", &Umur);

        idx = searchIndex(LD, NIK, Nama, Umur);

        printf("\\nMasukkan Data Baru\\n");

        update(idx, &LD);
    }

    else if (opt == 4) {
        printf("Import Data\n");

        char filename[100];

        printf("Masukkan nama file: ");

        scanf(" %[^\\n]s", &filename);

        readFile(filename, &LD);
    }
}

```

```

else if (opt == 5) {

    int idx;

    int NIK;

    char Nama[50];

    int Umur;


    //input;

    printf("NIK: ");

    scanf("%d", &NIK);

    printf("Nama Lengkap: ");

    scanf(" %[^\\n]s", &Nama);

    printf("Umur: ");

    scanf("%d", &Umur);


    idx = searchIndex(LD, NIK, Nama, Umur);

    if (idx != -1) {

        printData(LD.buffer[idx]); }

    else {

        printf("Data tidak ditemukan\\n");

    }

}

else {

    printf("input salah \\n");

}

}

return 0;

}

```


Catatan:

1. umur dihitung dari parameter tanggal lahir dan tanggal ketika aplikasi diakses
2. diperbolehkan untuk menggunakan library yang dibuat oleh orang lain
3. gunakan contoh dataset yang terdapat pada file "DataPenduduk31122019.csv" yang diberikan untuk melakukan testing (contoh dataset kependudukan diambil terakhir 31 Desember 2019) dengan informasi tambahan sebagai berikut:
 - a. penanggalan memiliki format DDMMYYYY; dengan D adalah tanggal, M adalah bulan, Y adalah tahun
 - b. data jenis kelamin; L adalah laki-laki dan P adalah perempuan

REFERENSI:

- [1] <https://www.geeksforgeeks.org/difference-between-1s-complement-representation-and-2s-complement-representation-technique/#:~:text=The%20main%20difference%20between%201.if%20we%20add%201%20to>, diakses 19 Maret 2022, 18:28 WIB.
- [2] <https://www.javatpoint.com/c-program-to-convert-decimal-to-binary>, diakses 19 Maret 2022, 20:19 WIB.
- [3] <https://www.sanfoundry.com/c-program-sum-binary-numbers/>, diakses 19 Maret 2022, 20:53 WIB.
- [4] <https://www.geeksforgeeks.org/bitwise-operators-in-c-cpp/>, diakses 19 Maret 2022, 20:55 WIB.
- [5] <https://www.javatpoint.com/binary-to-decimal-number-in-c>, diakses 19 Maret 2022, 22:10 WIB.
- [6] <http://users.ece.utexas.edu/~valvano/embed/chap3/chap3.htm>, diakses 19 Maret 2022, 22:53 WIB.
- [7] <https://www.tutorialspoint.com/c-program-to-find-two-s-complement-for-a-given-number>, diakses 19 Maret 2022, 23:40 WIB.
- [8] <https://www.sanfoundry.com/c-program-convert-decimal-hex/>, diakses 20 Maret 2022, 02:15 WIB.
- [9] <https://www.ibm.com/docs/en/zos/2.3.0?topic=functions-clock-determine-processor-time>, diakses 20 Maret 2022, 10:16 WIB.
- [10] https://www.tutorialspoint.com/c_standard_library/c_function_clock.htm, diakses 20 Maret 2022, 10:17 WIB.