RESPONSI 2 KONSEP PEMROGRAMAN

Nama : Muhammad Haidar Amru

NIM : L0125025

Program Studi : Informatika

Kelas : A

A. Judul Program

Judul dari program adalah “FATISDA Medical Center v2.0” yang inspirasinya saya ambil dari UNS Medical Center.

B. Deskripsi Program

Program yang saya buat kali ini mengambil tema pelayanan kesehatan. Sebenarnya ini adalah program yang kurang lebih sama seperti di Responsi 1 saya. Namun, saya mengubah struktur program hampir seutuhnya dan menambahkan beberapa fitur baru. Program ini sekarang memiliki file handling, sehingga dapat menyimpan data-data pasien yang sudah ada. Fitur baru lainnya yang saya tambahkan adalah animasi di main menu yang membuat program lebih menarik untuk dilihat. Selain itu, ada juga fitur untuk mengecek statistik umur pasien dan juga rekap kondisi pasien.

C. Daftar Fitur

1. Analisis Source Code

1.1 Header

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <windows.h>
```

a. #include <stdio.h>

Sebagai standar input output, wajib ada agar fungsi seperti printf, scanf, fopen, dll. bisa digunakan.

b. #include <string.h>

Sebagai string handling, digunakan untuk manipulasi teks seperti strcpy.

c. #include <stdlib.h>

Standard library, digunakan untuk manajemen memori (malloc) dan kontrol sistem.

d. #include <windows.h>

Windows API, library khusus untuk OS Windows. Digunakan di program ini untuk mengakses fungsi Sleep().

1.2 Konstanta

```
#define DATA_PASIEN "data pasien.txt"
```

Mendefinisikan nama file sebagai sebuah konstanta. Digunakan jika suatu saat kita mengganti nama file, kita cukup mengganti di baris define saja.

1.3 Fungsi Pembantu

```c
void delay(int milliseconds) {
    Sleep(milliseconds);
}

// Fungsi untuk mencetak teks per karakter (animasi)
void typeAnimation(char *teks, int kecepatan) {

    while (*teks != '\0') {
        printf("%c", *teks);
        fflush(stdout);
        delay(kecepatan);
        teks++;
    }
}

//Fungsi clear screen
void clearScreen() {
        system("cls");
}
```

a. void delay

Menyimpan fungsi sleep() bawaan dari Windows. Fungsi ini menjeda program selama beberapa milidetik, memberikan efek transisi yang halus.

b. void typeAnimation

Fungsi untuk memberikan efek animasi mengetik huruf per huruf. Menggunakan pointer *teks untuk melihat string dari karakter demi karakter. Fflush(stdout) digunakan agar fungsi memaksa huruf segera muncul di layar tanpa menunggu baris baru.

c. void clearScreen

Menyimpan perintah system(“cls”) yang berfungsi agar membersihkan layar terminal agar tampilan menu selalu rapi.

1.4 Struktur Data Pasien

```
struct dataPasien {
    int id;
    char nama[50];
    char jk; //(jenis kelamin L/P)
    int umur;
    float suhu;
    float berat;
    float tinggi;

    struct dataPasien *next;
};
```

Struktur data ini membungkus berbagai tipe data, int untuk ID pasien, char untuk nama dari pasien dan jenis kelaminnya, float untuk menyimpan suhu, berat, dan tinggi pasien dalam bentuk desimal.

Pointer *next adalah inti dari linked list. Variabel ini akan menghubungkan data satu pasien ke pasien berikutnya. Konsep ini disebut sebagai Self-Referential Structure. Pointer next sendiri nantinya akan digunakan untuk menyimpan alamat/memori dari data pasien berikutnya.

1.5 Variabel Global

```
struct dataPasien *head = NULL;
```

Berfungsi untuk membuat node pertama dalam rangkaian linked list bernama head. Di awal head akan selalu bernilai NULL saat program baru di mulai. Variabel ini diletakkan di luar fungsi agar bisa diakses oleh semua fungsi lainnya.

1.6 File Handling

a. Simpan Data Pasien

```
void simpanDataPasien(){
    FILE *fp = fopen(DATA_PASIEN, "w");
    if (fp == NULL) {
        printf("Gagal membuat file :(\n");
        return;
    }
    struct dataPasien *current = head;

    while (current != NULL){
        fprintf(fp, "%d#%s#%c#%d#%.1f#%.1f#%.1f\n", current->id, current->nama, current->jk, current->umur,
                current->suhu, current->tinggi, current->berat);
        current = current->next;
    }
    fclose (fp);
}
```

Di awal akan dideklarasikan pointer fp yang akan membuka konstanta DATA_PASIEN (file yang dituju adalah "data pasien.txt") dan menggunakan mode write atau menulis file. Kemudian akan dicek jika ada error dalam fopen. Ketika fp bernilai NULL, maka program akan dipaksa return supaya tidak terjadi crash program.

Kemudian akan dibuat pointer current yang berisikan data dari head. Pointer ini digunakan agar kita tidak harus menggeser headnya. Karena jika head digeser, program akan lupa di mana letak data pertama berada.

Untuk menyimpan file dalam txt akan dilakukan looping selama isi dari current tidak kosong (mengecek semua node), akan dilakukan fprintf (mencetak ke file) ke fp dengan format id#nama#jenis kelamin#umur#suhu#tinggi#berat\newline sampai current berada di node kosong. Jika sudah di node kosong maka file (fp) akan ditutup.

b. Muat Data Pasien

```
void muatDataPasien() {
    FILE *fp = fopen(DATA_PASIEN, "r");

    if (fp == NULL){
        return;
    }

    int tmpID;
    char tmpNama[50];
    char tmpJK;
    int tmpUmur;
    float tmpSuhu;
    float tmpBerat;
    float tmpTinggi;
```

Di awal akan dideklarasikan pointer fp yang akan membuka konstanta DATA_PASIEN dengan mode read (membaca). Lalu akan dilakukan pengecekan jika fp tidak ada isinya, program akan direturn supaya tidak crash.

Lalu kita perlu menyiapkan variabel temporarry (tmpID, tmpNama, dll.) untuk wadah sementara. Kita tidak boleh langsung membuat node sebelum kita yakin bahwa datanya bisa terbaca dengan benar. Kita harus membaca dulu variabel biasanya baru dimasukkan ke node.

```
while (fscanf(fp, "%d#%[^#]#%c#%d#%f#%f#%f",
    &tmpID, tmpNama, &tmpJK, &tmpUmur, &tmpSuhu, &tmpTinggi, &tmpBerat) != EOF){
        struct dataPasien *dataBaru = (struct dataPasien *)malloc(sizeof(struct dataPasien));

        dataBaru->id = tmpID;
        strcpy(dataBaru->nama, tmpNama);
        dataBaru->jk = tmpJK;
        dataBaru->umur = tmpUmur;
        dataBaru->suhu = tmpSuhu;
        dataBaru->tinggi = tmpTinggi;
        dataBaru->berat = tmpBerat;

        dataBaru->next = NULL;

        if (head == NULL){
        head = dataBaru;
        }
        else{
        struct dataPasien *temp = head;
        while (temp->next != NULL)
        {
        temp = temp->next;
        }
        temp->next = dataBaru;
        }
    }
    fclose(fp);
    printf("Data berhasil dimuat dari data pasien.txt\n");
    delay(1000);
    clearScreen();
}
```

Kemudian akan dilakukan looping untuk membaca file menggunakan fscanf. Di mana program akan membaca fp dengan seperti format tadi (id#nama#jk dll). Program hanya akan mengambil variabel (mengabaikan #) dan memasukkan nilai dari variabel itu ke variabel temporary yang kita buat tadi. Fscanf ini akan dilakukan selama fp belum menyentuh EOF (End of File) atau ujung dari isi txt.

Kemudian kita akan mengalokasikan memori untuk menyimpan data pasien baru (dataBaru) di struktur dataPasien yang sudah ada. Jika sudah, maka id, nama, jk, dll. dari data baru akan diisi oleh nilai dari variabel temporary tadi. Kemudian akan dicek apabila head nya masih kosong, maka data baru tersebut akan menjadi head dari nodenya. Jika tidak maka akan dilakukan pengecekan menggunakan pointer temp yang berisi head. Selama node sebelah dari temp (head) bukan NULL, node akan bergeser sampai node sebelahnya adalah NULL. Jika node sebelahnya adalah NULL, maka node sebelah tersebut akan diisi dataBaru. Terakhir program akan menutup fp dengan fclose.

1.7 Operasi Linked List

a. Fitur tambah data pasien

```
void tambahPasien() {
    struct dataPasien *pasienBaru = (struct dataPasien *)malloc(sizeof(struct dataPasien));

    if (pasienBaru == NULL) {
        printf("Memori Penuh!\n");
        return;
    }
```

Pertama-tama akan mengalokasikan memori untuk pasienBaru agar bisa menyimpan nilai-nilai dari pasien baru di struktur data pasien. Sebelum menambahkan data, akan dicek apakah memori di komputer masih ada dengan if (pasienBaru == NULL). Jika memori penuh maka program akan langsung berhenti

```
printf("=== Tambah Data Pasien ===\n");
printf("Masukkan ID Pasien: ");
scanf ("%d", &pasienBaru->id);
printf("Masukkan Nama Pasien: ");
scanf (" %[^\n]", pasienBaru->nama);
do
{
    printf("Masukkan Jenis Kelamin (L/P): ");
    scanf(" %c", &pasienBaru->jk);
} while (pasienBaru->jk != 'L' && pasienBaru->jk != 'P' && pasienBaru->jk != 'l' && pasienBaru->jk != 'p');

printf("Masukkan Umur Pasien (tahun): ");
scanf ("%d", &pasienBaru->umur);
printf("Masukkan Suhu Tubuh Pasien (Celsius): ");
scanf ("%f", &pasienBaru->suhu);
printf("Masukkan Berat Badan Pasien (kg): ");
scanf ("%f", &pasienBaru->berat);
printf("Masukkan Tinggi Badan Pasien (cm): ");
scanf ("%f", &pasienBaru->tinggi);
```

Kemudian program akan meminta input dari user yang akan dimasukkan ke ID, nama, dst. di node pasienBaru. Di bagian jenis kelamin, user akan dipaksa memasukkan jenis kelamin L/P (bisa dalam kapital atau tidak). Jika user menginputkan yang lain maka akan diulang hingga user menginputkan yang benar.

```
    //utk pasien pertama
    pasienBaru->next = NULL;

    //utk pasien pertama
    if (head == NULL)
    {
        head = pasienBaru;
    }
    //utk pasien selanjutnya
    else
    {
        //mencari letak pasien terakhir
        struct dataPasien *temp = head;
        while (temp->next != NULL)
        {
        temp = temp->next;
        }
        temp->next = pasienBaru;
    }
    printf("\nData pasien berhasil ditambahkan!\n");
    delay(1000);
}
```

Jika sudah, maka akan node selanjutnya dari pasien baru akan dibuat kosong karena bisa saja ada memori sampah di sebelah node pasienBaru. Lalu akan dicek apakah node head masih kosong, jika iya maka data pasienBaru akan menjadi headnya. Namun jika tidak, maka akan dilakukan pengecekan dengan pointer temp yang berisi node head. Di mana, selama node selanjutnya dari temp bukan berisi NULL, maka akan bergeser terus sampai node selanjutnya adalah NULL. Jika node selanjutnya adalah NULL maka pasienbaru akan ditaruh di node selanjutnya (node terakhir).

b. Fitur cek data pasien

```
void cekDataPasien() {
    if (head == NULL) {
        printf("\nBelum ada data pasien!\n");
        delay(2000);
        return;
    }

    struct dataPasien *current = head;
    int count = 0;
```

Fitur ini akan menampilkan semua data pasien nantinya. Pertama-tama akan dicek apabila head masih NULL berarti data pasien masih kosong. Lalu akan dibuat variabel sementara current yang berisi node head dan juga variabel count untuk menghitung jumlah pasien yang sudah terdafatar.

```
    printf("=== Lihat Semua Data Pasien ===\n");
    printf("-------------------------------------------------------------------------------------\n");
    printf("| %-3s | | %-15s | | %-4s | | %-4s | | %-6s  | | %-6s | | %-6s |\n",
           "ID", "Nama", "JK", "Umur", "Suhu", "Tinggi", "Berat");
    while (current != NULL) {
        printf("| %-3d | | %-15s | | %-4c | | %-4d | | %-6.1fC | | %-6.1f | | %-6.1f |\n",
               current->id, current->nama, current->jk, current->umur,
               current->suhu, current->tinggi, current->berat);

        //pindah ke pasien berikutnya
        current = current->next;
        count++;
```

Bagian ini akan mencetak seluruh data dari pasien mulai dari ID, nama, hingga berat badan. Bagian atas (sebelum while) akan menjadi header dari data ini. Kemudian di looping, selama current bukan NULL (ada isinya) maka akan dicetak id, nama, sampai berat dari pasien yang ada di node current. Setelah itu node akan bergeser ke node berikutnya lalu menambahkan variabel count sebesar 1.

```
    //itung jml total pasien
    printf("-------------------------------------------------------------------------------------\n");
    typeAnimation("Total pasien terdaftar: ", 50);
    printf("%d\n", count);
    delay(2000);
}
```

Jika sudah semua pasien tercetak (current == NULL) maka akan dicetak lagi jumlah dari pasien yang terdaftar saat ini.

1.8 Fungsi-fungsi untuk fitur program

a. Fitur Rekursif Mencari Data Pasien By ID

```
void cariPasienByID() {
    if (head == NULL){
        printf("\nBelum ada data pasien!\n");
        delay(1000);
        return;
    }

    int cariID;

    struct dataPasien *current = head;

    printf("Masukkan ID pasien yang ingin dicari : ");
    scanf("%d", &cariID);
    clearScreen();
    cariRekursif(head, cariID);

}
```

Fitur ini akan berfungsi untuk mencari data pasien melalui ID-nya. Seperti biasa, jika head berisi NULL maka berarti belum ada data pasien yang terdaftar. Lalu disiapkan variabel cariID sebagai id dari pasien yang ingin kita cari. Kemudian kita akan siapkan variabel sementara current yang berisi head. Lalu user akan diminta memasukkan ID pasien yang ingin dicari datanya. Kemudian program akan menjalankan fungsi cariRekursif dengan pointer current nantinya diisi node head dan variabel cariID akan dimasukkan ke idCari.

```
void cariRekursif(struct dataPasien *current, int idCari) {
    if(current == NULL){
        printf("Pasien dengan id %d tidak ditemukan!", idCari);
        return;
    }
    if (current->id == idCari){
        printf("Pasien ditemukan!\n");
        printf("-------------------------------------------------------------------------------\n");
        printf("| %-3s | | %-15s | | %-4s | | %-4s | | %-6s  | | %-6s | | %-6s |\n",
            "ID", "Nama", "JK", "Umur", "Suhu", "Tinggi", "Berat");

        printf("| %-3d | | %-15s | | %-4c | | %-4d | | %-6.1fC | | %-6.1f | | %-6.1f |\n",
                current->id, current->nama, current->jk, current->umur,
                current->suhu, current->tinggi, current->berat);

        return;
    }
    cariRekursif(current->next, idCari);
}
```

Fungsi ini akan menggunakan metode rekursif, di mana akan mengambil node current dan variabel idCari. di awal akan dicek apakah current == NULL, ini akan menjadi base case pertama. Di mana jika sampai ujung node id tidak ketemu maka program akan direturn. Di base case kedua, jika id dari current cocok dengan idCari. Maka akan dicetak data pasien tersebut dengan format yang sama seperti pada fungsi cekDataPasien. Jika base case kedua ini tidak terpenuhi maka fungsi cariRekursif akan dijalankan lagi dengan node di currentnya digeser ke node selanjutnya. Rekursi akan terjadi sampai current berisi NULL atau id current cocok dengan idCari.

b. Fitur Cek Kondisi Pasien

```
void cekKondisi() {
    if (head == NULL){
        printf("\nBelum ada data pasien!\n");
        delay(1000);
        return;
    }

    int cariID;
    struct dataPasien *current = head;
```

Fitur ini nantinya akan mengecek kondisi dari pasien berdasarkan suhunya. Sama seperti fitur sebelumnya, di awal akan dicek apakah head berisi NULL dan akan disiapkan variabel cariID serta pointer current yang berisi head.

```
    printf("Masukkan ID pasien yang ingin dicek : ");
    scanf("%d", &cariID);
    clearScreen();
    while (current != NULL){
    if (current->id == cariID){
            printf("=== Kondisi Pasien Saat Ini ===\n");
            printf ("\nNama : %s\n", current->nama);
            printf ("Suhu %s : %.1fC\n", current->nama, current->suhu);

            if (current->suhu < 36.0)
                printf("Keterangan kondisi %s : Suhu rendah, mengalami hipotermia ringan\n", current->nama);
            else if (current->suhu <= 37.5)
                printf("Keterangan kondisi %s : Suhu normal jaga kesehatan!\n", current->nama);
            else if (current->suhu <= 38.5)
                printf("Keterangan kondisi %s : Demam ringan, minum paracetamol!\n", current->nama);
            else if (current->suhu <= 40.0)
                printf("Keterangan kondisi %s : Demam tinggi, konsultasikan ke dokter segera!\n", current->nama);
            else
                printf("Keterangan kondisi : Suhu terlalu tinggi, %s butuh penanganan segera!\n", current->nama);

                return;
        }
        current = current->next;
    }
    printf("Pasien dengan id %d tidak ditemukan!", cariID);
    return;
}
```

Kemudian user akan diminta menginputkan id yang akan dicari, lalu loop akan dilakukan selama current tidak berisi NULL. Kemudian ada logika di mana jika id dari current cocok dengan cariID, maka akan dicetak kondisi pasien berdasarkan suhunya. Jika kurang dari 36 derajat maka suhu rendah, kurang dari sama dengan 37.5 derajat maka suhu normal, jika kurang dari sama dengan 38.5 derajat maka demam ringan, jika kurang dari sama dengan 40 derajat maka demam tinggi, dan jika diatas itu maka kondisi suhu sudah memasuki status darurat. Namun jika id current tidak cocok dengan cariID maka node akan digeser ke next sampai id current cocok atau current == NULL. Jika current berisi NULL maka berarti pasien dengan id tersebut tidak ditemukan.

c. Fitur Cek Berat Badan Ideal Pasien

```
void cekBBI() {

    float bbi = 0;

    if (head == NULL){
        printf("\nBelum ada data pasien!\n");
        delay(1000);
        return;
    }

    int cariID;
    struct dataPasien *current = head;
```

Fitur ini nantinya akan menghitung BBI dari pasien sesuai ID nya. Proses awalnya sama seperti fitur cek kondisi pasien, hanya saja kita menyiapkan variabel bbi (float untuk desimal) untuk menyimpan hasil hitungan bbi.

```
while (current != NULL){
if (current->id == cariID){
    if (current->jk == 'l' || current->jk == 'L'){
    bbi = (current->tinggi - 100) * 0.9;
        }
    else if (current->jk == 'p' || current->jk == 'P'){
    bbi = (current->tinggi - 100) * 0.85;
        }
    else {
            printf("Error: Jenis kelamin pasien tidak valid!\n");
            return;
        }
```

Sama seperti fitur sebelumnya akan dicek dulu mana id pasien yang cocok dengan id yang diinput (logikanya sama seperti fitur sebelumnya). Jika id yang cocok ditemukan, maka program akan menghitung dulu berapa BBI dari pasien tersebut. Jika jenis kelamin pasien laki-laki (L/l) maka rumus BBI nya adalah (tinggi-100)*0.9, jika perempuan (P/p) maka rumusnya adalah (tinggi-100)*0.85, dan jika input jenis kelamin salah maka akan diprint pesan error dan direturn.

```
        printf ("=== BBI Pasien === \n");
        printf("Nama   : %s\n", current->nama);
        printf("Tinggi : %.1f cm\n", current->tinggi);
        printf("Berat  : %.1f kg\n", current->berat);
        printf("BBI    : %.1f kg\n", bbi);

        if (current->berat > bbi + 5){
            printf("Berat badan %s di atas ideal.\n", current->nama);
            }
        else if (current->berat < bbi - 5){
            printf("Berat badan %s di bawah ideal.\n",  current->nama);
            }
        else {
            printf ("Berat badan %s sudah ideal.",  current->nama);
            }
        return;
        }
        current = current->next;
    }
    printf("Pasien dengan id %d tidak ditemukan!", cariID);
    return;
}
```

Setelah BBI dihitung maka akan dicetak data dari pasien mulai dari nama tinggi berat dan BBI. Kemudian untuk hasil BBI akan dicek dulu apabila berat pasien lebih dari BBI maka berat badannya di atas ideal, apabila berat pasien kurang dari BBI maka berat badannya di bawah ideal, jika tidak keduanya maka berat pasien sudah ideal. Sama seperti fitur sebelumnya, jika id current tidak cocok dengan cariID maka node akan digeser hingga current == NULL atau cariID cocok dengan current id.

d. Fitur Cek Statistik Umur Pasien

juga

```c
void statistikUmur() {
    if (head == NULL) {
        printf("\nBelum ada data pasien!\n");
        return;
    }

    int kumpulanUmur[100];
    int jumlahData = 0;
    float totalUmur = 0;

    struct dataPasien *current = head;

    while (current != NULL && jumlahData < 100) {
        kumpulanUmur[jumlahData] = current->umur;
        current = current->next;
        jumlahData++;
    }

    printf("=== Statistik Umur Pasien (Dari Array) ===\n");
    printf("Data Umur: ");
    for (int i = 0; i < jumlahData; i++) {
        printf("%d ", kumpulanUmur[i]);
        totalUmur += kumpulanUmur[i];
    }

    printf("\nRata-rata Umur: %.2f tahun\n", totalUmur / jumlahData);
}
```

Fitur ini akan berfungsi untuk mendata rata-rata umur pasien yang terdaftar. Seperti biasa, di awal akan dicek apakah node sudah berisi atau kosong. Kemudian akan disiapkan array yang menampung maksimal 100 umur dari pasien. Kemudian disiapkan juga variabel jumlahdata untuk menghitung berapa banyak data yang tersalin ke array.

Kemudian akan dijalankan loop di mana, selama current tidak sama dengan NULL dan jumlahData kurang dari 100 berarti masih ada data pasien dan array belum penuh. Kemudian indeks dari array kumpulanUmur akan diisi umur dari current. Kemudian akan bergeser ke node berikutnya dan jumlahData akan ditambah 1 sampai current berisi NULL.

Kemudian semua umur tersebut akan dicetak melalui looping for. Variabel totalUmur akan menjumlahkan semua umur dari array ke dalam satu wadah. TotalUmur tersebut akan dibagi dengan jumlahdata pasien yang terdaftar, menghasilkan rata-rata umur pasien yang terdaftar.

e. Fitur Rekap Kondisi Pasien

```
void rekapKondisi (){
    if (head == NULL) {
        printf("\nBelum ada data pasien!\n");
        return;
    }

    int suhu[3] = {0, 0, 0};

    struct dataPasien *current = head;

    while (current != NULL){
        if (current->suhu <= 37.5){
            suhu[0]++;

        }
        else if (current->suhu <= 39.0){
            suhu[1]++;
        }
        else {
            suhu[2]++;
        }
        current = current->next;
    }
    printf("=== Rekap Kondisi Pasien ===\n");
    printf("1. Kondisi Normal/Aman  : %d pasien\n", suhu[0]);
    printf("2. Kondisi Demam        : %d pasien\n", suhu[1]);
    printf("3. Kondisi DARURAT      : %d pasien\n", suhu[2]);
}
```

Fitur ini akan merekap jumlah kondisi pasien berdasarkan suhunya. Sama seperti fitur lainnya, jika head berisi NULL maka berarti belum ada data pasien yang terdaftar. Kemudian untuk menampung data pasiennya, kita menyiapkan array suhu dengan 3 elemen. Lalu kita siapkan pointer current yang berisi head.

Untuk mendata, kita lakukan looping selama current tidak sama dengan NULL maka akan mengelompokkan berbagai kondisi pasien. Jika suhu

pasien kurang dari sama dengan 37.5 maka indeks suhu ke 0 akan ditambah 1, Jika suhu pasien kurang dari sama dengan 39.0 maka indeks suhu ke 1 akan ditambah 1, selain itu indeks suhu ke 2 akan ditambah 1. Kemudian node akan bergeser ke next sampai NULL. Lalu hasil akan menampilkan jumlah masing-masing dari kondisi pasien berdasarkan suhunya.

1.9 Program Utama

a. Menu Awal

```
int main (){
    typeAnimation("Loading data pasien...", 5);
    delay(1000);
    clearScreen();
    muatDataPasien();
    int pilMenu;
    char pilUlang;

    // --- FATISDA ---
    typeAnimation("                    _____   _ _____ ___ ____  ____    _      \n", 5);
    typeAnimation("                   |  ___| / \\|_   _|_ _/ ___||  _ \\  / \\   \n", 5);
    typeAnimation("                   | |_   / _ \\ | |  | |\\___ \\| | | |/ _ \\  \n", 5);
    typeAnimation("                   |  _| / ___ \\| |  | | ___) | |_| / ___ \\ \n", 5);
    typeAnimation("                   | |  / /   \\ \\ |  | ||____/|____/_/   \\_\\\n", 5);
    typeAnimation("                   |_| /_/     \\_\\_| |___|                  \n", 5);
    typeAnimation("\n", 5);

    // --- MEDICAL ---
    typeAnimation("  __  __ _____ ____ ___ ____    _    _          ____ _____ _   _ _____ _____ ____        \n", 5);
    typeAnimation(" |  \\/  | ___|  _ \\_ _/ ___|  / \\  | |        / ___| ____| \\ | |_   _| ____|  _ \\       \n", 5);
    typeAnimation(" | |\\/| |  _| | | | | |     / _ \\ | |       | |   |  _| |  \\| | | | |  _| | |_) |      \n", 5);
    typeAnimation(" | |  | | |___| |_| | |___ / ___ \\| |___    | |___| |___| |\\  | | | | |___|  _ <       \n", 5);
    typeAnimation(" |_|  |_|_____|____/___\\____/_/   \\_\\_____|   \\____|_____|_| \\_| |_| |_____|_| \\_\\      \n", 5);
    typeAnimation("\n", 5);

    typeAnimation("                          BY Muhammad Haidar Amru                    \n", 20);
    typeAnimation("                        Informatika A, NIM L0125025                  \n", 20);
    typeAnimation("                       Version 2.0.  -  December 2025                \n", 20);
    printf("\n");

    typeAnimation("Press Enter to continue...", 50);
    getchar();
    clearScreen();
```

Sebelum masuk ke menu utama, di awal akan ditampilkan tulisan loading data pasien lalu setelah jeda 1 detik (fungsi delay), akan dipanggil fungsi muatDataPasien untuk membaca file txt dari komputer. Disiapkan juga variabel pilihan pada menu (pilMenu) dan pilihan untuk kembali ke menu (pilUlang). Lalu akan ditampilkan teks ASCII bertuliskan FATISDA MEDICAL CENTR dengan animasi menggunakan typeAnimation. Lalu ditampilkan juga nama saya, prodi kelas, NIM, dan versi serta tanggal update program ini dengan typeAnimation. Lalu user perlu menekan enter untuk lanjut ke menu utama (getchar untuk mendeteksi apakah user menekan enter). Lalu layar terminal akan dibersihkan dengan fungsi clearScreen.

b. Menu Utama

```
do
{
    do {
    printf ("============================\n");
    printf ("     SILAHKAN PILIH MENU    \n");
    printf ("============================\n");
    printf("\n1. Tambah Data Pasien");
    printf("\n2. Lihat Semua Data Pasien");
    printf("\n3. Cari Data Pasien Berdasarkan ID");
    printf("\n4. Cek Kondisi Pasien");
    printf("\n5. Hitung Berat Badan Ideal");
    printf("\n6. Statistik Umur");
    printf("\n7. Rekap Kondisi Pasien");
    printf("\n8. Keluar Program\n");

    printf("\n---------------------------");
    printf("\nPilih menu (1-8) : ");
    int pilihan = scanf("%d", &pilMenu);

    if (pilihan != 1) {
        while(getchar() != '\n');
        pilMenu = 0;
    }

    if (pilMenu < 1 || pilMenu > 8){
        printf ("\nPilihan tidak VALID, silahkan coba lagi\n");
        delay (1000);
        clearScreen();
        }

    } while (pilMenu < 1 || pilMenu > 8);
```

Menu ini berisi pilihan fitur-fitur dari program ini mulai dari tambah data, hingga rekap kondisi. User akan diminta untuk memasukkan menu yang ingin diakses (disimpan di pilMenu), jika user memasukkan selain angka maka akan terdeteksi dan input tersebut “diubah” menjadi 0 dan disimpan di pilMenu. Dan juga selama user memasukkan angka selain 1-8 maka menu akan mengulang sampai user memasukkan angka yang tepat.

c. Pemanggilan Fungsi

```
    switch (pilMenu) {
case 1:
    clearScreen();
    tambahPasien();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();

    break;
case 2 :
    clearScreen();
    cekDataPasien();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
case 3 :
    clearScreen();
    cariPasienByID();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
case 4 :
    clearScreen();
    cekKondisi();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
```

```
case 5 :
    clearScreen();
    cekBBI();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
case 6 :
    clearScreen();
    statistikUmur();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
case 7 :
    clearScreen();
    rekapKondisi();
    printf("\nKembali ke menu utama? (y/n) : ");
    scanf(" %c", &pilUlang);
    clearScreen();
    break;
case 8 :
    clearScreen();
    simpanDataPasien();
    typeAnimation("Data Pasien berhasil disimpan!\n", 10);
    typeAnimation ("\nTerima kasih telah menggunakan layanan FATISDA Medical Center v2.0!\n", 10);
    delay(1000);
    return 0;
default:
    break;
```

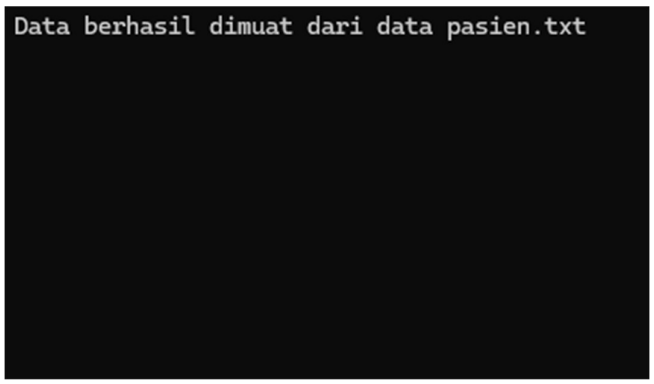
```
    }
        } while (pilUlang == 'y' || pilUlang == 'Y');

    return 0;
    }
```

Switch case digunakan untuk mendeteksi pilihan fitur oleh user berdasarkan pilMenu. Setiap case akan berawalan dengan clearScreen lalu pemanggilan fungsi. Setelah fungsi selesai dipanggil, maka user akan ditanya apakah ingin kembali ke menu (disimpan di pilUlang), jika jawabannya ya (Y/y) maka program syarat while akan terpenuhi dan program akan kembali ke menu awal. Program akan berakhir jika user memilih menu nomor 8, di mana program akan memanggil fungsi simpanDataPasien lalu menutup program dengan return 0.

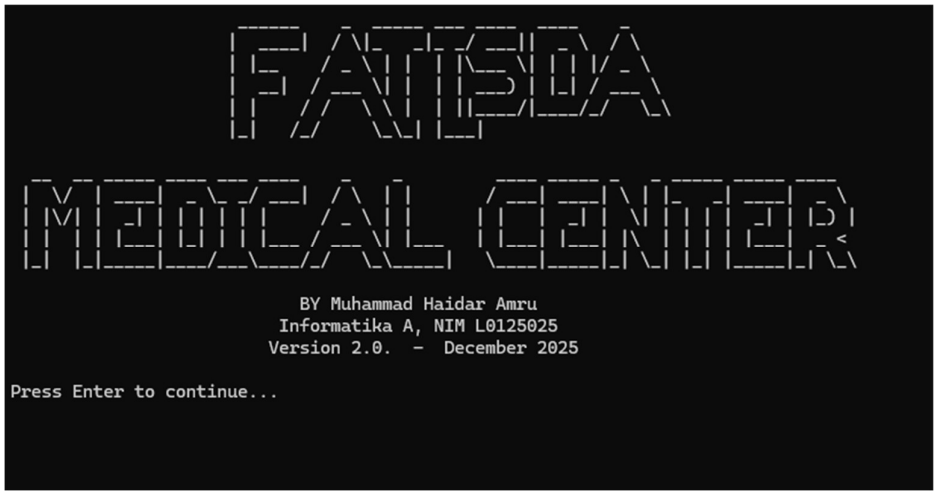
2. Jalan Kerja Program

2.1 Loading Screen

2.2 Fungsi Muat Data Pasien Dijalankan

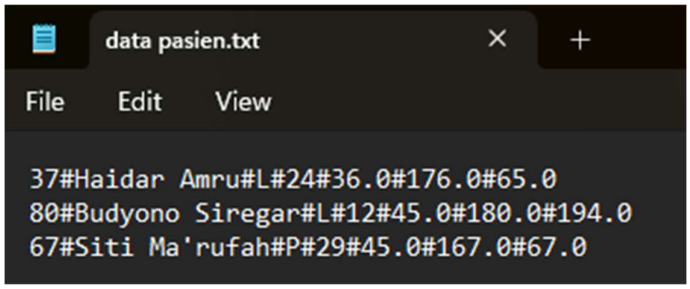


2.3 Tampilan Awal Program (ada animasi menggunakan fungsi typeAnimation)



2.4 Menu Awal

2.5 Isi File txt Sebelum Penambahan Data



2.6 Fitur Tambah Data Pasien (user memberi input satu persatu)

```
=== Tambah Data Pasien ===
Masukkan ID Pasien: 25
Masukkan Nama Pasien: Mamat Junaedi
Masukkan Jenis Kelamin (L/P): L
Masukkan Umur Pasien (tahun): 20
Masukkan Suhu Tubuh Pasien (Celsius): 42.3
Masukkan Berat Badan Pasien (kg): 56
Masukkan Tinggi Badan Pasien (cm): 160

Data pasien berhasil ditambahkan!

Kembali ke menu utama? (y/n) :
```

2.7 Kembali ke menu utama

```
============================
     SILAHKAN PILIH MENU
============================

1. Tambah Data Pasien
2. Lihat Semua Data Pasien
3. Cari Data Pasien Berdasarkan ID
4. Cek Kondisi Pasien
5. Hitung Berat Badan Ideal
6. Statistik Umur
7. Rekap Kondisi Pasien
8. Keluar Program

----------------------------
Pilih menu (1-8) :
```

2.8 Fitur Lihat Semua Data Pasien

```
=== Lihat Semua Data Pasien ===
------------------------------------------------------------------------------
| ID  | | Nama            | | JK   | | Umur | | Suhu    | | Tinggi | | Berat  |
| 37  | | Haidar Amru     | | L    | | 24   | | 36.0  C | | 176.0  | | 65.0   |
| 80  | | Budyono Siregar | | L    | | 12   | | 45.0  C | | 180.0  | | 194.0  |
| 67  | | Siti Ma'rufah   | | P    | | 29   | | 45.0  C | | 167.0  | | 67.0   |
| 25  | | Mamat Junaedi   | | L    | | 20   | | 42.3  C | | 160.0  | | 56.0   |
------------------------------------------------------------------------------
Total pasien terdaftar: 4

Kembali ke menu utama? (y/n) :
```

2.9 Fitur Cek Kondisi Pasien

```
=== Kondisi Pasien Saat Ini ===

Nama : Siti Ma'rufah
Suhu Siti Ma'rufah : 45.0C
Keterangan kondisi : Suhu terlalu tinggi, Siti Ma'rufah butuh penanganan segera!

Kembali ke menu utama? (y/n) :
```

```
=== Kondisi Pasien Saat Ini ===

Nama : Mamat Junaedi
Suhu Mamat Junaedi : 42.3C
Keterangan kondisi : Suhu terlalu tinggi, Mamat Junaedi butuh penanganan segera!

Kembali ke menu utama? (y/n) :
```

2.10 Fitur Cek BBI Pasien

```
=== BBI Pasien ===
Nama   : Haidar Amru
Tinggi : 176.0 cm
Berat  : 65.0 kg
BBI    : 68.4 kg
Berat badan Haidar Amru sudah ideal.
Kembali ke menu utama? (y/n) :
```

```
=== BBI Pasien ===
Nama   : Budyono Siregar
Tinggi : 180.0 cm
Berat  : 194.0 kg
BBI    : 72.0 kg
Berat badan Budyono Siregar di atas ideal.

Kembali ke menu utama? (y/n) :
```

2.11 Fitur Statistik Umur

```
=== Statistik Umur Pasien (Dari Array) ===
Data Umur: 24 12 29 20
Rata-rata Umur: 21.25 tahun

Kembali ke menu utama? (y/n) :
```

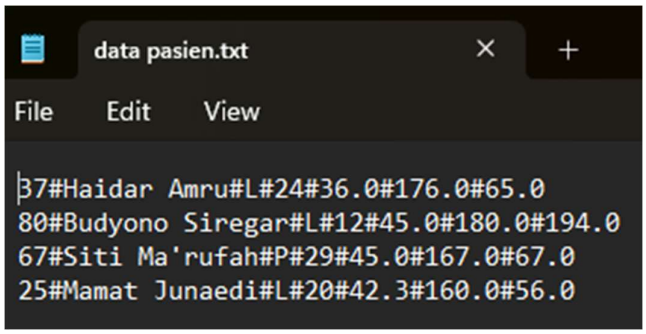
2.12 Fitur Rekap Kondisi Pasien

```
=== Rekap Kondisi Pasien ===

1. Kondisi Normal/Aman  : 1 pasien
2. Kondisi Demam        : 0 pasien
3. Kondisi DARURAT      : 3 pasien

Kembali ke menu utama? (y/n) :
```

2.13 Keluar Program dan Menyimpan Data Pasien

```
Data Pasien berhasil disimpan!

Terima kasih telah menggunakan layanan FATISDA Medical Center v2.0!
```

2.14 Isi File txt Setelah Penambahan Data



2.15 Error Input

a. Input Menu Salah

```
============================
     SILAHKAN PILIH MENU
============================

1. Tambah Data Pasien
2. Lihat Semua Data Pasien
3. Cari Data Pasien Berdasarkan ID
4. Cek Kondisi Pasien
5. Hitung Berat Badan Ideal
6. Statistik Umur
7. Rekap Kondisi Pasien
8. Keluar Program

----------------------------
Pilih menu (1-8) : a

Pilihan tidak VALID, silahkan coba lagi
```

```
============================
     SILAHKAN PILIH MENU
============================

1. Tambah Data Pasien
2. Lihat Semua Data Pasien
3. Cari Data Pasien Berdasarkan ID
4. Cek Kondisi Pasien
5. Hitung Berat Badan Ideal
6. Statistik Umur
7. Rekap Kondisi Pasien
8. Keluar Program

----------------------------
Pilih menu (1-8) : 99

Pilihan tidak VALID, silahkan coba lagi
```

b. Input Jenis Kelamin Salah

```
=== Tambah Data Pasien ===
Masukkan ID Pasien: 22
Masukkan Nama Pasien: Pak Vincent
Masukkan Jenis Kelamin (L/P): D
Masukkan Jenis Kelamin (L/P): U
Masukkan Jenis Kelamin (L/P): A
Masukkan Jenis Kelamin (L/P): L
Masukkan Umur Pasien (tahun):
```

c. ID Pasien Tidak Ditemukan

```
Masukkan ID pasien yang ingin dicari : 999
```

```
Pasien dengan id 999 tidak ditemukan!
Kembali ke menu utama? (y/n) :
```