Praktikum

Struktur Data

Projek Akhir

Dosen Pembimbing:

Randi Proska Sandra, M.Sc

Disusun oleh :

Nama : Rafli Arianto

NIM : 23343051

UNIVERSITAS NEGERI PADANG

FAKULTAS TEKNIK

INFORMATIKA

2024

Nama Program

Program untuk Berlangganan Gym

Latar Belakang Pembuatan Program

Saat ini, gaya hidup sehat dan kebugaran fisik menjadi semakin penting bagi banyak orang. Gym atau pusat kebugaran telah menjadi pilihan populer untuk membantu setiap orang yang ingin mencapai tujuan kesehatan dan kebugaran. Seiring dengan meningkatnya minat terhadap kebugaran, manajemen keanggotaan gym menjadi lebih kompleks. Karenanya, sebuah sistem manajemen yang efisien dan terorganisir sangat diperlukan untuk mengelola data keanggotaan, memantau durasi berlangganan, serta memfasilitasi administrasi yang efektif.

Dengan adanya program manajemen langganan gym, informasi penting mengenai anggota dapat disimpan dan diakses dengan mudah, mengurangi risiko kesalahan dan meningkatkan efisiensi operasional. Program ini memungkinkan staf gym untuk dengan mudah menambahkan, mengedit, mencari, dan menghapus data keanggotaan. Fitur seperti pencarian berdasarkan ID atau nama, serta pengurutan data, mempermudah pengelolaan informasi anggota. Dengan akses cepat dan mudah ke data anggota, staf gym dapat memberikan layanan yang lebih baik dan responsif.

Program ini membantu memastikan bahwa data keanggotaan disimpan dengan aman dan terstruktur, mengurangi risiko kehilangan atau kerusakan data yang sering terjadi pada pencatatan manual. Data yang tersimpan secara digital memungkinkan gym untuk melakukan analisis lebih lanjut mengenai keanggotaan, seperti tren berlangganan, tingkat retensi anggota, dan efektivitas promosi. Ini membantu dalam pengambilan keputusan yang lebih baik dan strategi bisnis yang lebih terfokus. Dengan mengotomatiskan tugas-tugas administrasi, program ini membantu menghemat waktu staf, selain itu, penghematan waktu ini berguna secara tidak langsung pada penghematan biaya operasional.

Unsur atau Konsep Bahasa Pemrograman yang digunakan

Looping:

Penggunaan loop untuk menghasilkan 3 huruf kapital secara acak sebagai id konsumen yang baru ditambahkan.

```
for (int i = 0; i < 3; i++) {
        id[i] = 'A' + rand() % 26;
    }
```

Penggunaan loop untuk terus menghasilkan output jika belum mencapai jumlah konsumen yang terdata.

```
for (int i = 0; i < jumlahKonsumen; i++) {

    }
```

Penggunaan loop untuk pembatalan langganan atau penghapusan data konsumen dari indeks konsumen

```
for (int i = indeksKonsumen; i < jumlahKonsumen - 1; i++) {

        konsumen[i] = konsumen[i + 1];

    }
```

Penggunaan loop untuk memlih menu yang terdapat pada program yang nanti terdapat switch case di dalamnya

```
do {

        printf("1. Tambah Konsumen\n");

        printf("2. Hitung Total Harga Berlangganan\n");

        printf("3. Tampilkan Data Semua Konsumen\n");

        printf("4. Tampilkan Sisa Durasi Berlangganan\n");

        printf("5. Batalkan Langganan\n");

        printf("6. Cetak Data Konsumen Baru\n");

        printf("0. Keluar\n");

        printf("Pilih menu: ");

        scanf("%d", &pilihan);

    } while (pilihan != 0);
```

Decision Making:

Decision making digunakan untuk menghasilkan output sebuah pernyataan jika kondisi jumlah konsumen sudah sama dengan atau akan melebihi maksimal konsumen

```
if (jumlahKonsumen >= MAKSIMAL_KONSUMEN) {

        printf("Konsumen sudah mencapai batas maksimal.\n");

        return;

    }
```

Decision making digunakan untuk menghasilkan total harga berlangganan konsumen jika id yang diinputkan sama dengan id yang terdata

```
if (strcmp(konsumen[i].id, id) == 0) {
            totalHarga = konsumen[i].durasi * 200000;
            break;
        }
```

Decision making digunakan untuk menghasilkan durasi tersisa dari waktu berlangganan konsumen jika id yang yang diinputkan sama dengan id yang terdata

```
if (strcmp(konsumen[i].id, id) == 0) {
            durasiTersisa = konsumen[i].durasi;
            break;
        }
```

Decision making digunakan untuk menghasilkan indeks konsumen pada bagian pembatalan berlangganan jika id yang diinputkan sama dengan id yang terdata

```
if (strcmp(konsumen[i].id, id) == 0) {
            indeksKonsumen = i;
            break;
        }
```

Decision making digunakan untuk menghasilkan pernyataan jika konsumen yang dimaksud tidak ada pada indeks yang terdata

```
if (indeksKonsumen == -1) {
        printf("Tidak ada konsumen yang dimaksud.\n\n");
        return;
    }
```

Decision making digunakan untuk memilih menu dan menampilkan output sesuai dengan pilihan yang diinputkan

```
switch (pilihan) {
            case 1:
                tambahKonsumen();
                break;
            case 2:
                hitungTotalHarga();
                break;
            case 3:
```

```
            konsumenYangBerlangganan();
            break;
        case 4:
            durasiTersisa();
            break;
        case 5:
            pembatalanBerlangganan();
            break;
        case 6:
         cetakDataKonsumenBaru();
         break;
        case 0:
            printf("Keluar dari program.\n");
            break;
        default:
            printf("Pilih menu yang lain.\n");
            break;
    }
```

Array:

Deklarasi array untuk id konsumen, nama konsumen dan telepon konsumen

```
char id[4];
char nama[50];
char telepon[15];
```

Menghasilkan ouput berupa id, nama, telepon dan durasi berlangganan konsumen berdasarkan sesuai urutan array

```
printf("ID: %s\n", konsumen[i].id);
printf("Nama: %s\n", konsumen[i].nama);
printf("Telepon: %s\n", konsumen[i].telepon);
printf("Durasi: %d bulan\n", konsumen[i].durasi);
```

Fungsi:

Fungsi untuk judul program

```
void judul(){ }
```

Fungsi untuk membuat id konsumen secara acak

```
void pembuatanID(char *id) { }
```

Fungsi untuk input data konsumen baru

```
void tambahKonsumen() { }
```

Fungsi untuk hitung total harga berlangganan konsumen berdasarkan waktu berlangganan

```
void hitungTotalHarga() { }
```

Fungsi untuk menampilkan data konsumen yang berlangganan

```
void konsumenYangBerlangganan() { }
```

Fungsi untuk menampilkan durasi waktu berlangganan konsumen yang tersisa

```
void durasiTersisa() { }
```

Fungsi untuk melakukan pembatalan berlangganan konsumen

```
void pembatalanBerlangganan() { }
```

Fungsi untuk mencetak data konsumen pada file .txt

```
void cetakDataKonsumenBaru() { }
```

Struct:

Deklarasi struct untuk data konsumen berupa id, nama, telepon dan durasi berlangganan dengan nama struct Konsumen

```
typedef struct {
    char id[4];
    char nama[50];
    char telepon[15];
    int durasi;
} Konsumen;
```

Input data konsumen berupa nama, telepon dan durasi berlangganan dengan menggunakan struct Konsumen

```
scanf("%s", konsumenBaru.nama);
```

```
scanf("%s", konsumenBaru.telepon);
```

```
scanf("%d", &konsumenBaru.durasi);
```

Output id konsumen dari struct Konsumen

```
printf("ID konsumen adalah: %s\n",konsumenBaru.id);
```

File Handling:

Membuka file dengan nama KonsumenBaru.txt

```
FILE *file = fopen("KonsumenBaru.txt", "w");
```

Mencetak data konsumen berupa id, nama, telepon dan durasi berlangganan pada file KonsumenBaru.txt

```
fprintf(file, "ID: %s\n", konsumen[i].id);
```

```
fprintf(file, "Nama: %s\n", konsumen[i].nama);
```

```
fprintf(file, "Telepon: %s\n", konsumen[i].telepon);
```

```
fprintf(file, "Durasi: %d bulan\n", konsumen[i].durasi);
```

```
fprintf(file, "--------------------\n");
```

Menutup file KonsumenBaru.txt

```
fclose(file);
```

Penjelasan Aplikasi

Menu:

1. Menu pertama berfungsi untuk menginputkan data konsumen baru berupa nama, nomor telepon dan durasi berlangganan yang diinginkan oleh konsumen serta menampilkan id khusus untuk konsumen yang dihasilkan berupa 3 huruf kapital secara acak.

2. Menu kedua berguna untuk menampilkan data semua konsumen yang sedang berlangganan.

3. Menu ketiga berguna untuk menampilkan urutan konsumen berdasarkan nama dari A ke Z

4. Menu keempat berguna untuk menampilkan urutan konsumen berdasarkan nama dari Z ke A

5. Menu kelima memiliki fungsi untuk mencari konsumen berdasarkan ID

6. Menu kelima memiliki fungsi untuk mencari konsumen berdasarkan nama

7. Menu ketujuh berfungsi untuk mengedit data konsumen berdasarkan ID, seperti mengubah nama, nomor telepon dan durasi berlangganan

8. Menu kedelapan berfungsi untuk membatalkan berlangganan konsumen dan menghapusnya dari data yang tersimpan berdasarkan ID

9. Menu terkakhir berfungsi untuk keluar dari program

Fitur:

1. Menghasilkan id untuk konsumen berupa 3 huruf kapital secara acak

2. Pembatalan berlangganan konsumen dan penghapusan data konsumen dari data tersimpan

3. Mengedit data konsumen

4. Mengurutkan data konsumen berdasarkan nama dari A ke Z maupun sebaliknya

5. Mencari data konsumen berdasarkan nama dan juga ID

6. Menampilkan data konsumen dengan urutan data terlama ada di paling atas

Penjelasan Baris Kode Program:

Membuat struct untuk data konsumen dengan nama Konsumen yang berisi nama, id, nomor telepon dan juga durasi lama berlangganan serta membuat penamaan untuk double link list

```
typedef struct Konsumen {
    char id[PANJANG_ID];
    char nama[PANJANG_NAMA];
    char telepon[NO_TELEPON];
    int durasi;
    struct Konsumen *sebelum;
    struct Konsumen *sesudah;
} Konsumen;

Konsumen *depan = NULL, *belakang = NULL;
```

Membuat id untuk konsumen dengan rand yaitu 3 huruf kapital secara acak

```
void buatID(char *id) {
    static const char alfabet[] = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    for (int i = 0; i < PANJANG_ID - 1; i++) {
        id[i] = alfabet[rand() % 26];
    }
```

```
    id[PANJANG_ID - 1] = '\0';
}
```

Membuat bagian untuk menambahkan data konsumen dengan input nama, nomor telepon dan durasi berlangganan yang nantinya akan muncul kode secara acak sebagai id konsumen serta membuatnya menjadi double link list

```
void tambahKonsumen() {
    Konsumen *konsumenBaru = (Konsumen *)malloc(sizeof(Konsumen));
    buatID(konsumenBaru->id);
    printf("Ketikan Nama: ");
    scanf("%s", konsumenBaru->nama);
    printf("Ketikan Nomor Telepon: ");
    scanf("%s", konsumenBaru->telepon);
    printf("Ketikan durasi berlanganan (bulan): ");
    scanf("%d", &konsumenBaru->durasi);

    konsumenBaru->sebelum = belakang;
    konsumenBaru->sesudah = NULL;
    if (belakang) {
        belakang->sesudah = konsumenBaru;
    } else {
        depan = konsumenBaru;
    }
    belakang = konsumenBaru;

    printf("ID Konsumen: %s\n", konsumenBaru->id);
}
```

Membuat bagian untuk menampilkan data konsumen dengan urutan data terlama ada di paling atas

```
void tampilkanKonsumen() {
    Konsumen *sesuai = depan;
```

```
    while (sesuai) {
        printf("ID: %s, Nama: %s, Telepon: %s, Durasi: %d bulan\n",
               sesuai->id, sesuai->nama, sesuai->telepon, sesuai->durasi);
        printf("--------------------\n");
        sesuai = sesuai->sesudah;
    }
}
```

Membuat bagian untuk membatalkan berlangganan konsumen berdasarkan id yang diinput dan nantinya data konsumen akan dihapus dari data tersimpan

```
void batalkanBerlangganan(char *id) {
    Konsumen *sesuai = depan;
    while (sesuai) {
        if (strcmp(sesuai->id, id) == 0) {
            if (sesuai->sebelum) {
                sesuai->sebelum->sesudah = sesuai->sesudah;
            } else {
                depan = sesuai->sesudah;
            }
            if (sesuai->sesudah) {
                sesuai->sesudah->sebelum = sesuai->sebelum;
            } else {
                belakang = sesuai->sebelum;
            }
            free(sesuai);
            printf("Berlangganan dengan ID %s batal.\n", id);
            return;
        }
        sesuai = sesuai->sesudah;
    }
    printf("Konsumen dengan ID %s tidak ada.\n", id);
}
```

Membuat bagian yang berfungsi untuk mengurutkan data konsumen berdasarkan nama konsumen dari A ke Z atau sebaliknya dengan merge sort

```
void urutanKonsumendenganMerge(Konsumen **depanDepan, int meningkat);

void bagiDaftar(Konsumen *asal, Konsumen **bagianDepan, Konsumen
**bagianBelakang);

Konsumen* urutandenganMerge(Konsumen *a, Konsumen *b, int meningkat);


void urutanKonsumendengannama(int meningkat) {

    urutanKonsumendenganMerge(&depan, meningkat);

}


void urutanKonsumendenganMerge(Konsumen **depanDepan, int meningkat) {

    Konsumen *depan = *depanDepan;

    Konsumen *a;

    Konsumen *b;


    if ((depan == NULL) || (depan->sesudah == NULL)) {

        return;

    }


    bagiDaftar(depan, &a, &b);


    urutanKonsumendenganMerge(&a, meningkat);

    urutanKonsumendenganMerge(&b, meningkat);


    *depanDepan = urutandenganMerge(a, b, meningkat);

}


Konsumen* urutandenganMerge(Konsumen *a, Konsumen *b, int meningkat) {

    Konsumen *hasil = NULL;


    if (a == NULL)

        return b;
```

```
    else if (b == NULL)
        return a;

    if (meningkat ? (strcmp(a->nama, b->nama) <= 0) : (strcmp(a->nama, b->nama) >= 0)) {
        hasil = a;
        hasil->sesudah = urutandenganMerge(a->sesudah, b, meningkat);
        hasil->sesudah->sebelum = hasil;
        hasil->sebelum = NULL;
    } else {
        hasil = b;
        hasil->sesudah = urutandenganMerge(a, b->sesudah, meningkat);
        hasil->sesudah->sebelum = hasil;
        hasil->sebelum = NULL;
    }
    return hasil;
}

void bagiDaftar(Konsumen *asal, Konsumen **bagianDepan, Konsumen **bagianBelakang) {
    Konsumen *cepat;
    Konsumen *lama;
    if (asal == NULL || asal->sesudah == NULL) {
        *bagianDepan = asal;
        *bagianBelakang = NULL;
    } else {
        lama = asal;
        cepat = asal->sesudah;

        while (cepat != NULL) {
            cepat = cepat->sesudah;
            if (cepat != NULL) {
                lama = lama->sesudah;
```

```
                cepat = cepat->sesudah;
            }
        }

        *bagianDepan = asal;
        *bagianBelakang = lama->sesudah;
        lama->sesudah = NULL;
    }
}
```

Membuat bagian untuk mencari data konsumen dengan id konsumen

```
Konsumen* cariKonsumendenganID(char *id) {
    Konsumen *sesuai = depan;
    while (sesuai) {
        if (strcmp(sesuai->id, id) == 0) {
            return sesuai;
        }
        sesuai = sesuai->sesudah;
    }
    return NULL;
}
```

Membuat bagian untuk mencari data konsumen dengan nama konsumen

```
Konsumen* cariKonsumendengannama(char *nama) {
    Konsumen *sesuai = depan;
    while (sesuai) {
        if (strcmp(sesuai->nama, nama) == 0) {
            return sesuai;
        }
        sesuai = sesuai->sesudah;
    }
    return NULL;
```

```
}
```

Membuat bagian untuk mengedit data konsumen berdasarkan id yang dipilih

```
void editdataKonsumen(char *id) {
    Konsumen *konsumen = cariKonsumendenganID(id);
    if (!konsumen) {
        printf("Konsumen dengan ID %s tidak ada.\n", id);
        return;
    }

    printf("Edit data konsumen dengan ID %s\n", id);
    printf("Ketikan nama baru: ");
    scanf("%s", konsumen->nama);
    printf("Ketikan nomor telepon baru: ");
    scanf("%s", konsumen->telepon);
    printf("Ketikan durasi berlangganan baru (bulan): ");
    scanf("%d", &konsumen->durasi);
}
```

Fungsi untuk judul program

```
void judul(){
      printf("Program Manajemen Langganan Gym\n\n");
}
```

Bagian utama dari program yang berfungsi untuk menjalankan semua bagian yang sudah dibuat dengan pilihan menu menu untuk memanggil fungsi perbagian sesuai kode yang sudah dibuat dengan tujuan berbeda dari setiap menu

```
int main() {
    int pilihan;
    char id[PANJANG_ID];
    char nama[PANJANG_NAMA];
```

```
judul();
do {
    printf("1. Tambah konsumen\n");
    printf("2. Tampilkan semua data konsumen\n");
    printf("3. Urutan konsumen berdasarkan nama (A-Z)\n");
    printf("4. Urutan konsumen berdasarkan nama (Z-A)\n");
    printf("5. Cari data konsumen dengan ID\n");
    printf("6. Cari data konsumen dengan nama\n");
    printf("7. Edit data konsumen berdasarkan ID\n");
    printf("8. Batalkan berlangganan berdasarkan ID\n");
    printf("0. Keluar\n");
    printf("Pilih menu: ");
    scanf("%d", &pilihan);

    switch (pilihan) {
        case 1:
            tambahKonsumen();
            break;
        case 2:
            tampilkanKonsumen();
            break;
        case 3:
            urutanKonsumendengannama(1);
            tampilkanKonsumen();
            break;
        case 4:
            urutanKonsumendengannama(0);
            tampilkanKonsumen();
            break;
        case 5: {
            printf("Ketikan ID konsumen: ");
            scanf("%s", id);
```

```
                Konsumen *konsumenberdasarID = cariKonsumendenganID(id);
                if (konsumenberdasarID) {
                    printf("ID: %s, Nama: %s, Telepon: %s, Durasi: %d
bulan\n",
                           konsumenberdasarID->id, konsumenberdasarID->nama,
konsumenberdasarID->telepon, konsumenberdasarID->durasi);
                    printf("--------------------\n");
                } else {
                    printf("Konsumen dengan ID %s tidak ada.\n", id);
                }
                break;
            }
            case 6: {
                printf("Ketikan nama konsumen: ");
                scanf("%s", nama);
                Konsumen *konsumenberdasarnama =
cariKonsumendengannama(nama);
                if (konsumenberdasarnama) {
                    printf("ID: %s, Nama: %s, Telepon: %s, Durasi: %d
bulan\n",
                           konsumenberdasarnama->id, konsumenberdasarnama-
>nama, konsumenberdasarnama->telepon, konsumenberdasarnama->durasi);
                    printf("--------------------\n");
                } else {
                    printf("Konsumen dengan nama %s tidak ada.\n", nama);
                }
                break;
            }
            case 7:
                printf("Ketikan ID konsumen: ");
                scanf("%s", id);
                editdataKonsumen(id);
                break;
            case 8:
```

```
                printf("Ketikan ID konsumen: ");
                scanf("%s", id);
                batalkanBerlangganan(id);
                break;
            case 0:
                printf("Keluar dari program.\n");
                break;
            default:
                printf("Pilih menu yang lain.\n");
        }
    } while (pilihan != 0);

    return 0;
}
```