

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **TUGAS AKHIR**

**“DAFTAR MENU WARUNG KOPI”**

Dosen Pengampu :

Randi Proska Sandra, M.Sc



Disusun Oleh :

Nama : Ridho Hamdani Putra

NIM : 23343052

Kode kelas : 202323430156 (INF1.62.2014)

**PROGRAM STUDI INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS NEGERI PADANG**

**2024**

## A. LATAR BELAKANG PROGRAM

Saya membuat program ini karena orang tua saya membuka warung kopi di rumah. Warung kopi ini menyediakan berbagai jenis minuman dan makanan ringan. Namun, pengelolaan menu di warung kopi ini masih dilakukan secara manual, yang seringkali menyebabkan kesalahan dalam mencatat pesanan, kesulitan dalam memperbarui harga, dan masalah dalam mengatur menu yang tersedia. Oleh karena itu, saya merasa perlu untuk membuat suatu sistem yang dapat membantu mempermudah manajemen menu di warung kopi tersebut.

Program ini dirancang untuk mengatasi berbagai masalah dalam pengelolaan menu di warung kopi. Dengan menggunakan program ini, saya dapat dengan mudah menambah, mengedit, menghapus, mencari, mengurutkan, dan menampilkan menu dengan lebih efisien. Hal ini tidak hanya akan mengurangi kesalahan dalam pencatatan, tetapi juga akan mempercepat proses pelayanan kepada pelanggan. Selain itu, program ini dapat membantu dalam mengatur harga jual sehingga lebih mudah untuk melakukan penyesuaian harga.

## B. PENJELASAN PROGRAM SECARA UMUM

Program ini dibuat untuk mengelola daftar menu di warung kopi. Program ini menyediakan beberapa fitur, seperti:

- **Menambah menu baru:** Pengguna dapat menambahkan menu baru dengan memasukkan nama dan harga menu.
- **Mengedit menu:** Pengguna dapat mengedit nama dan harga menu yang sudah ada.
- **Menghapus menu:** Pengguna dapat menghapus menu yang sudah ada.
- **Mencari menu:** Pengguna dapat mencari menu berdasarkan nama.
- **Mengurutkan menu:** Pengguna dapat mengurutkan menu berdasarkan harga (dari terendah ke tertinggi).
- **Menampilkan semua menu:** Pengguna dapat melihat daftar semua menu beserta nama dan harganya.

Program ini menggunakan struktur data linked list untuk menyimpan daftar menu. Setiap node dalam linked list berisi informasi tentang satu menu, termasuk nama dan harga.

### C. PENJALASAN BARIS PRORGAM

```
7  #include <stdio.h>
8  #include <stdlib.h>
9  #include <string.h>
```

Baris ini merupakan penggunaan beberapa header file dengan memanggil library yang dibutuhkan untuk fungsi standar dalam bahasa C.

```
11 // Definisi struktur Menu untuk linked list
12 typedef struct Menu {
13     char nama[50];
14     double harga;
15     struct Menu* berikutnya;
16 } Menu;
```

Baris kode ini mendefinisikan struktur data **Menu** yang digunakan untuk menyimpan informasi tentang satu menu. Struktur ini memiliki tiga anggota:

- **nama**: Sebuah string yang berisi nama menu.
- **harga**: Sebuah double yang berisi harga menu.
- **berikutnya**: Sebuah pointer ke node berikutnya dalam linked list.

```
18 // Fungsi untuk membuat Menu baru
19 Menu* buatMenu(char* nama, double harga) {
20     Menu* MenuBaru = (Menu*)malloc(sizeof(Menu));
21     strcpy(MenuBaru->nama, nama);
22     MenuBaru->harga = harga;
23     MenuBaru->berikutnya = NULL;
24     return MenuBaru;
25 }
```

Fungsi ini membuat menu baru untuk linked list. Fungsi ini menerima dua parameter: **nama** dan **harga**. Fungsi ini mengalokasikan memori untuk menu baru, menyalin nama dan harga ke menu baru, mengatur pointer selanjutnya ke NULL, dan kemudian mengembalikan pointer ke menu baru.

```
27 // Fungsi untuk menambah menu
28 void tambahMenu(Menu** awal, char* nama, double harga) {
29     Menu* MenuBaru = buatMenu(nama, harga);
30     MenuBaru->berikutnya = *awal;
31     *awal = MenuBaru;
32 }
```

Fungsi ini menambahkan menu baru ke awal linked list. Fungsi ini menerima tiga parameter: **awal**, **nama**, dan **harga**. Parameter **awal** adalah pointer ke pointer ke menu pertama dalam linked list. Fungsi ini membuat menu baru menggunakan fungsi **buatMenu**, kemudian mengatur pointer next menu baru ke menu pertama dalam linked list, dan terakhir mengatur pointer awal ke menu baru.

```

34 // Fungsi untuk mengedit menu
35 void editMenu(Menu* awal, char* namaLama, char* namaBaru, double hargaBaru) {
36     Menu* sementara = awal;
37     while (sementara != NULL) {
38         if (strcmp(sementara->nama, namaLama) == 0) {
39             strcpy(sementara->nama, namaBaru);
40             sementara->harga = hargaBaru;
41             return;
42         }
43         sementara = sementara->berikutnya;
44     }
45 }

```

Fungsi ini mengedit menu yang sudah ada. Fungsi ini menerima empat parameter: **awal**, **namaLama**, **namaBaru**, dan **hargaBaru**. Parameter **awal** adalah pointer ke menu pertama dalam linked list. Fungsi ini mencari menu dengan nama yang sesuai dengan **namaLama**. Jika menu ditemukan, fungsi ini menyalin **namaBaru** dan **hargaBaru** ke menu tersebut dan kemudian return.

```

47 // Fungsi untuk menghapus menu
48 int hapusMenu(Menu** awal, char* nama) {
49     Menu* sementara = *awal;
50     Menu* prev = NULL;
51     if (sementara != NULL && strcmp(sementara->nama, nama) == 0) {
52         *awal = sementara->berikutnya;
53         free(sementara);
54         return 1;
55     }
56     while (sementara != NULL && strcmp(sementara->nama, nama) != 0) {
57         prev = sementara;
58         sementara = sementara->berikutnya;
59     }
60     if (sementara == NULL) return 0;
61     prev->berikutnya = sementara->berikutnya;
62     free(sementara);
63     return 1;
64 }

```

Fungsi ini menghapus menu dari linked list. Fungsi ini menerima dua parameter: **awal** dan **nama**. Parameter **awal** adalah pointer ke pointer ke menu pertama dalam linked list. Fungsi ini mencari menu dengan nama yang sesuai dengan **nama**. Jika menu ditemukan, fungsi ini menghapus menu tersebut dari linked list dan membebaskan memori yang digunakan oleh menu. Fungsi ini kemudian mengembalikan 1 jika menu berhasil dihapus, dan 0 jika menu tidak ditemukan.

```

66 // Fungsi untuk menampilkan semua menu
67 void tampilkanMenu(Menu* awal) {
68     Menu* sementara = awal;
69     printf("+-----+\n");
70     printf("| No | %-40s | %-16s |\n", "Nama", "Harga");
71     printf("+-----+\n");
72     int nomorUrut = 1;
73     while (sementara != NULL) {
74         printf("| %-2d | %-40s | Rp. %-12.2f |\n", nomorUrut, sementara->nama, sementara->harga);
75         sementara = sementara->berikutnya;
76         nomorUrut++;
77     }
78     printf("+-----+\n");
79 }

```

Fungsi ini menampilkan daftar semua menu beserta nama dan harganya. Fungsi ini menerima satu parameter: **awal**. Parameter **awal** adalah pointer ke menu pertama dalam linked list. Fungsi ini iterates melalui linked list dan mencetak informasi tentang setiap menu.



Fungsi ini digunakan untuk menampilkan judul program

```
131 // Fungsi utama
132 int main() {
133     Menu* awal = NULL;
134     int pilihan;
135     char nama[50], namaLama[50];
136     double harga;
137     int hasilScan;
138
139     system("cls");
140     judul();
141
142     do {
143         printf("\nMenu:\n");
144         printf("1. Tambah Menu\n");
145         printf("2. Edit Menu\n");
146         printf("3. Hapus Menu\n");
147         printf("4. Cari Menu\n");
148         printf("5. Urutkan Menu berdasarkan Harga\n");
149         printf("6. Tampilkan Semua Menu\n");
150         printf("7. Keluar\n");
151         printf("\nPilih (1-7): ");
152         hasilScan = scanf("%d", &pilihan);
153         while (getchar() != '\n'); // Clear input buffer
154
155         if (hasilScan != 1 || pilihan < 1 || pilihan > 7) {
156             printf("Pilihan tidak valid!\n");
157             continue; // Meminta input ulang jika input tidak valid
158         }
159
160         switch (pilihan) {
161             case 1:
162                 printf("Masukkan Nama: ");
163                 fgets(nama, sizeof(nama), stdin);
164                 nama[strcspn(nama, "\n")] = 0; // Hapus newline character
165
166                 do {
167                     printf("Masukkan Harga: ");
168                     hasilScan = scanf("%lf", &harga);
169                     while (getchar() != '\n'); // Clear input buffer
170                     if (hasilScan != 1) {
171                         printf("Input tidak valid. Masukkan angka.\n");
172                     }
173                 } while (hasilScan != 1);
174
175                 tambahMenu(&awal, nama, harga);
176                 break;
177             case 2:
178                 printf("Masukkan Nama Menu yang Ingin Diedit: ");
179                 fgets(namaLama, sizeof(namaLama), stdin);
```

```

179 fgets(namaLama, sizeof(namaLama), stdin);
180 namaLama[strcspn(namaLama, "\n")] = 0; // Hapus newline character
181 printf("Masukkan Nama Baru: ");
182 fgets(nama, sizeof(nama), stdin);
183 nama[strcspn(nama, "\n")] = 0; // Hapus newline character
184
185 do {
186     printf("Masukkan Harga Baru: ");
187     hasilScan = scanf("%lf", &harga);
188     while (getchar() != '\n'); // Clear input buffer
189     if (hasilScan != 1) {
190         printf("Input tidak valid. Masukkan angka.\n");
191     }
192 } while (hasilScan != 1);
193
194 editMenu(awal, namaLama, nama, harga);
195 break;
196
197 case 3:
198     printf("Masukkan Nama Menu yang Ingin Dihapus: ");
199     fgets(nama, sizeof(nama), stdin);
200     nama[strcspn(nama, "\n")] = 0; // Hapus newline character
201     int berhasil = hapusMenu(awal, nama);
202     if (berhasil) {
203         printf("Menu berhasil dihapus.\n");
204     } else {
205         printf("Menu tidak ditemukan.\n");
206     }
207     break;
208
209 case 4:
210     printf("Masukkan Nama yang Dicari: ");
211     fgets(nama, sizeof(nama), stdin);
212     nama[strcspn(nama, "\n")] = 0; // Hapus newline character
213     Menu* hasil = cariMenu(awal, nama);
214     if (hasil != NULL) {
215         printf("Menu Ditemukan \nNama: %s, Harga: Rp. %.2f\n", hasil->nama, hasil->harga);
216     } else {
217         printf("Menu Tidak Ditemukan.\n");
218     }
219     break;
220
221 case 5:
222     urutkanMenu(awal);
223     printf("Menu berhasil diurutkan berdasarkan harga.\n");
224     break;
225
226 case 6:
227     tampilkanMenu(awal);
228     break;
229
230 case 7:
231     exit(0);
232
233 default:
234     printf("Pilihan tidak valid!\n");
235 }
236
237 } while (1);
238
239 return 0;
240 }

```

Fungsi main adalah fungsi utama dalam program ini. Fungsi ini memiliki beberapa tugas, yaitu:

- **Menginisialisasi linked list:** Fungsi ini menginisialisasi pointer **awal** ke NULL, yang berarti linked list kosong.
- **Menampilkan menu:** Fungsi ini menampilkan menu utama program kepada pengguna.
- **Memproses pilihan pengguna:** Fungsi ini memproses pilihan yang dimasukkan oleh pengguna. Ada 7 pilihan yang tersedia:
  1. **Tambah Menu:** Memanggil fungsi **tambahMenu** untuk menambahkan menu baru.
  2. **Edit Menu:** Memanggil fungsi **editMenu** untuk mengedit menu yang sudah ada.
  3. **Hapus Menu:** Memanggil fungsi **hapusMenu** untuk menghapus menu yang ada.
  4. **Cari Menu:** Memanggil fungsi **cariMenu** untuk mencari menu berdasarkan nama.
  5. **Urutkan Menu berdasarkan Harga:** Memanggil fungsi **urutkanMenu** untuk mengurutkan menu berdasarkan harga.
  6. **Tampilkan semua menu:** Memanggil fungsi **tampilkanMenu** untuk menampilkan semua menu yang telah diinputkan.
  7. **Keluar:** Keluar dari program.

## **D. KESIMPULAN**

Dalam laporan ini, telah dibahas sebuah program manajemen daftar menu untuk warung kopi menggunakan bahasa pemrograman C. Program ini memanfaatkan struktur data linked list untuk menyimpan dan mengelola menu-menu yang tersedia. Pengguna dapat melakukan berbagai operasi, seperti menambah, mengedit, menghapus, mencari, mengurutkan, dan menampilkan menu. Dengan menggunakan loop dan fungsi-fungsi yang sesuai, program ini memberikan pengalaman interaktif kepada pengguna untuk mengelola daftar menu secara efisien. Selain itu, program juga dilengkapi dengan fitur validasi input untuk memastikan kelengkapan dan keakuratan data yang dimasukkan pengguna. Dengan demikian, program ini dapat menjadi alat yang berguna dalam membantu pengelolaan menu di warung kopi.