

Nama : Muhammad Nabil Afrizal Rahmadani
NRP : 5025231014

Final Project Dasar Pemrograman Sistem Database Menu Toko

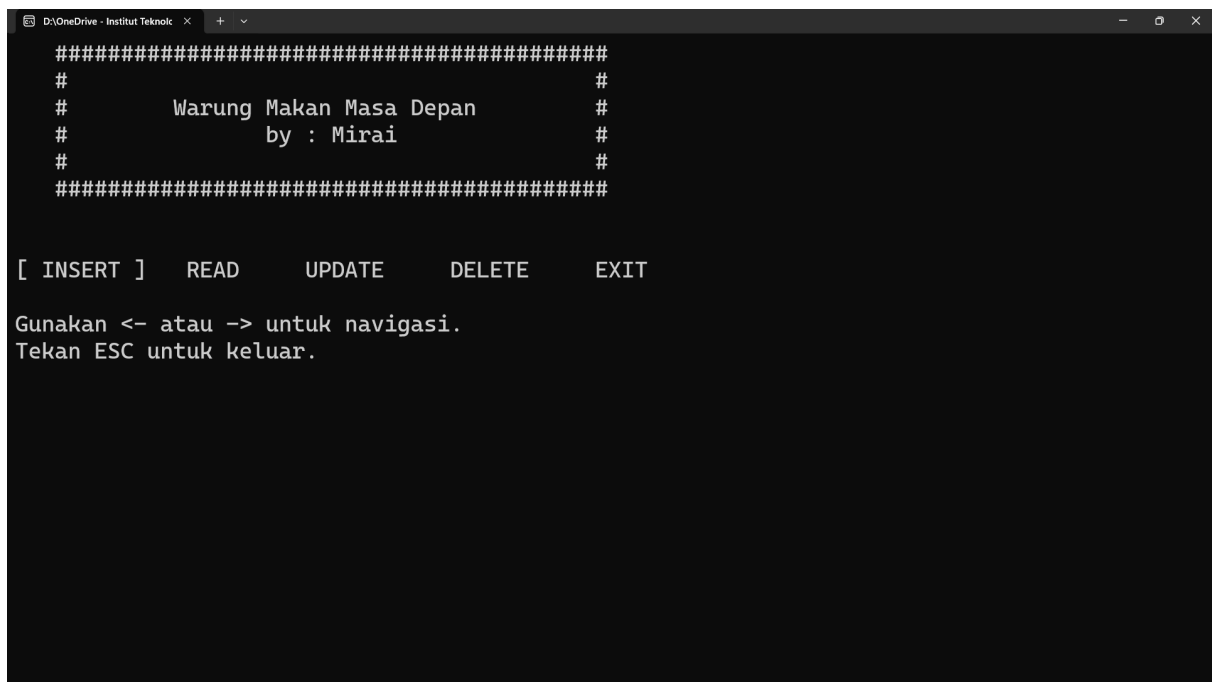
- Cara menggunakan program

Untuk menjalankan program pertama kali, program wajib di compile terlebih dahulu, untuk meng-compile program, eksekusi command line berikut pada folder project.

```
gcc main.c -o 5025231014_finalProject.exe
```

Setelah di compile, anda akan mendapati file baru sesuai dengan nama **5025231014_finalProject.exe**

Saat program dibuka, akan ada tampilan seperti berikut:



```
D:\OneDrive - Institut Teknok x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#              by : Mirai            #
#                                     #
#####

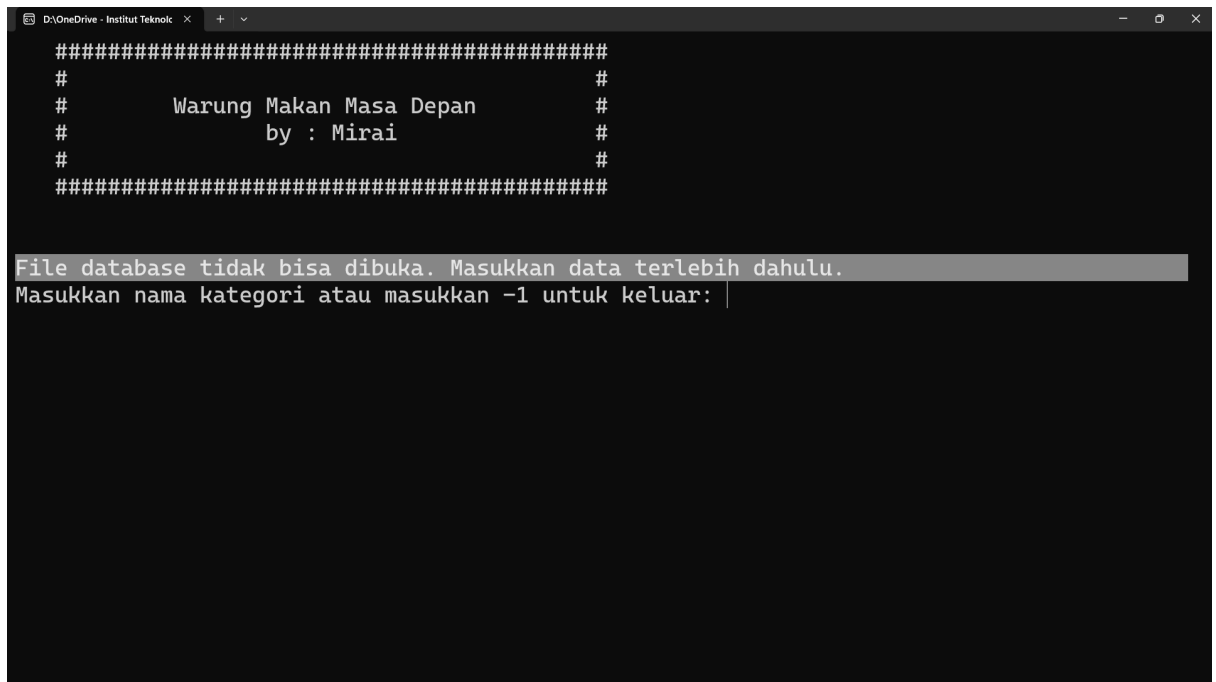
[ INSERT ]  READ      UPDATE      DELETE      EXIT

Gunakan <- atau -> untuk navigasi.
Tekan ESC untuk keluar.
```

Anda dapat menggunakan *Arrow Key* kiri dan kanan untuk navigasi start menu. Terdapat empat fitur yang disediakan disini yaitu **INSERT DATA, READ DATA, UPDATE DATA, DELETE DATA.**

1. INSERT DATA

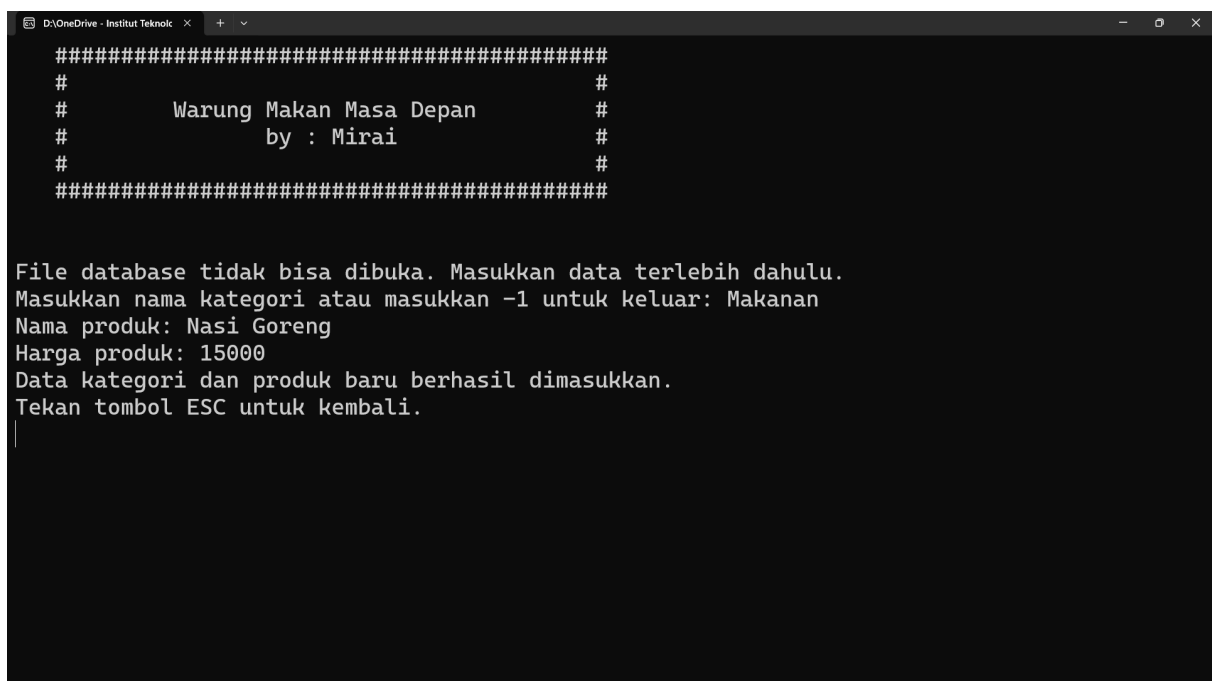
Saat anda pertama kali menggunakan program ini, akan muncul *warning* **File database tidak bisa dibuka. Masukkan data terlebih dahulu**, seperti gambar dibawah:



```
D:\OneDrive - Institut Teknolc x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#           by : Mirai              #
#                                     #
#####

File database tidak bisa dibuka. Masukkan data terlebih dahulu.
Masukkan nama kategori atau masukkan -1 untuk keluar: |
```

Disini saya akan mencoba untuk menambahkan kategori dan produk seperti gambar dibawah:



```
D:\OneDrive - Institut Teknolc x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#           by : Mirai              #
#                                     #
#####

File database tidak bisa dibuka. Masukkan data terlebih dahulu.
Masukkan nama kategori atau masukkan -1 untuk keluar: Makanan
Nama produk: Nasi Goreng
Harga produk: 15000
Data kategori dan produk baru berhasil dimasukkan.
Tekan tombol ESC untuk kembali.
|
```

Maka data akan berhasil dimasukkan ke database.

Bagaimana ketika anda ingin memasukkan produk baru ke kategori yang sudah ada? Cukup ketik nama kategori nya dan masukkan nama dan harga produk baru tersebut.

```
D:\OneDrive - Institut Teknol... x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#              by : Mirai            #
#                                     #
#####

===== Makanan =====
Nama      Harga
Nasi Goreng      15000
Masukkan nama kategori atau masukkan -1 untuk keluar: makanan
Kategori makanan ditemukan!
Masukkan nama produk: Nasi Pecel
Harga produk: 10000
Data produk berhasil dimasukkan.
Tekan tombol ESC untuk kembali...
```

Ada perbedaan “**Success Message**” yang ditampilkan. Jika anda memasukkan produk ke dalam kategori yang sudah ada, maka success message yang ditampilkan adalah “**Data produk berhasil dimasukkan.**” Sedangkan jika anda membuat kategori baru, maka success message yang tampil adalah “**Data kategori dan produk berhasil dimasukkan**”.

2. READ DATA

Fungsi ini membaca data menu dari database.

```
D:\OneDrive - Institut Teknol... x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#              by : Mirai            #
#                                     #
#####

===== Makanan =====
Nama      Harga
Nasi Goreng      15000
Nasi Pecel      10000
===== Minuman =====
Nama      Harga
Es Teh      3000

Tekan ESC untuk kembali...
```

3. UPDATE DATA

Saat anda memilih menu ini, anda akan diharuskan untuk memilih kategori terlebih dahulu.

```
D:\OneDrive - Institut Teknol x + v
#####
#
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#
#####

===== Makanan =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar:
```

Disini saya akan menggunakan kategori makanan untuk demonstrasi.

```
D:\OneDrive - Institut Teknol x + v
#####
#
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#
#####

===== Makanan =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: Makanan
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: |
```

Terdapat 3 pilihan yaitu:

1. Update Nama Kategori

Saat anda memilih pilihan ini, user akan diminta untuk memasukkan nama baru, disini saya akan menggunakan nama Foods untuk demonstrasi.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Makanan =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: Makanan
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: 1
Nama kategori sekarang -> Makanan
Masukkan nama kategori baru: Foods
Data telah ter-update!
Tekan tombol ESC untuk kembali..|
```

Saat anda memilih fungsi READ pada start menu, anda akan mendapati bahwa kategori makanan telah berubah mejadi Foods.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Foods =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Tekan ESC untuk kembali...
```

2. Update Harga Produk

Saat anda memilih fungsi ini, user akan diminta untuk memberikan kategori yang terdapat produk yang ingin diubah, untuk demonstrasi, saya akan mengubah harga Nasi Goreng.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Foods =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: foods
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: 2
Masukkan nama produk yang ingin diubah harga nya: Nasi Goreng
```

Disini user diminta untuk memberikan nama produk yang ingin diubah harganya. Saya akan mencoba untuk memberikan harga baru 20000

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Foods =====
Nama      Harga
Nasi Goreng    15000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: foods
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: 2
Masukkan nama produk yang ingin diubah harga nya: Nasi Goreng
Masukkan harga baru: 20000
Data telah ter-update!
Tekan tombol ESC untuk kembali...|
```

Data telah terupdate dengan sukses.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#             by : Mirai             #
#                                     #
#####

===== Foods =====
Nama      Harga
Nasi Goreng    20000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000

Tekan ESC untuk kembali...
```

Dan disini, harga Nasi Goreng sudah diganti menjadi 20000

3. Update Nama Produk

Di fungsi ini, user dapat mengubah nama produk yang sudah ada.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#             by : Mirai             #
#                                     #
#####

===== Foods =====
Nama      Harga
Nasi Goreng    20000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: foods
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: 3
Masukkan nama produk saat ini: |
```

Untuk demonstrasi, saya akan mencoba untuk mengubah nama Nasi Goreng menjadi “Fried Rice”.

```
D:\OneDrive - Institut Teknol x + v
#####
#
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#
#####

===== Foods =====
Nama      Harga
Nasi Goreng    20000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Masukkan nama kategori atau -1 untuk keluar: foods
1. Update Nama Kategori
2. Update Harga Produk
3. Update Nama Produk
Masukkan pilihan: 3
Masukkan nama produk saat ini: nasi goreng
Nama lama -> Nasi Goreng
Masukkan nama baru: Fried Rice
Data telah ter-update!
Tekan tombol ESC untuk kembali...
```

```
D:\OneDrive - Institut Teknol x + v
#####
#
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#
#####

===== Foods =====
Nama      Harga
Fried Rice    20000
Nasi Pecel     10000
===== Minuman =====
Nama      Harga
Es Teh     3000
Tekan ESC untuk kembali...
```

Saat anda baca kembali menggunakan fungsi READ, terlihat bahwa nama sudah berubah.

4. DELETE DATA

Pada fungsi ini, user dapat menghapus data 1 produk atau seluruh kategori beserta produknya.

Disini saya akan menggunakan kategori Foods sebagai demonstrasi.


```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Foods =====
Nama      Harga
Fried Rice 20000
Nasi Pecel 10000
===== Minuman =====
Nama      Harga
Es Teh    3000
Masukkan nama kategori atau -1 untuk keluar: Foods
1. Hapus isi seluruh kategori
2. Hapus produk dalam kategori
Masukkan pilihan:
```

User akan diberikan 2 pilihan, yaitu menghapus isi seluruh kategori atau menghapus suatu produk dalam kategori.

1. Hapus isi seluruh kategori

Pada fungsi ini, user akan menghapus kategori beserta isi menu nya, saya akan menggunakan kategori Minuman untuk demonstrasi.

```
D:\OneDrive - Institut Teknol x + v
#####
#                                     #
#      Warung Makan Masa Depan      #
#      by : Mirai                    #
#                                     #
#####

===== Foods =====
Nama      Harga
Fried Rice 20000
Nasi Pecel 10000
===== Minuman =====
Nama      Harga
Es Teh    3000
Masukkan nama kategori atau -1 untuk keluar: minuman
1. Hapus isi seluruh kategori
2. Hapus produk dalam kategori
Masukkan pilihan: 1
Data telah terhapus!
Tekan tombol ESC untuk kembali...|
```

```
D:\OneDrive - Institut Teknol x + v
#####
#
#          Warung Makan Masa Depan          #
#              by : Mirai                    #
#
#####

===== Foods =====
Nama          Harga
Fried Rice    20000
Nasi Pecel    10000

Tekan ESC untuk kembali...
```

Saat anda menggunakan fungsi READ, dilihat data Minuman sudah tidak ada lagi.

2. Hapus produk dalam kategori

Pada fungsi ini, user bisa menghapus salah satu produk dari suatu kategori, saya akan menggunakan kategori Foods dan Nasi Pecel untuk demonstrasi.

```
D:\OneDrive - Institut Teknol x + v
#####
#
#          Warung Makan Masa Depan          #
#              by : Mirai                    #
#
#####

===== Foods =====
Nama          Harga
Fried Rice    20000
Nasi Pecel    10000
Masukkan nama kategori atau -1 untuk keluar: foods
1. Hapus isi seluruh kategori
2. Hapus produk dalam kategori
Masukkan pilihan: 2
Kategori terpilih -> Foods
Masukkan nama produk yang ingin dihapus: nasi pecel
Data telah terhapus!
Tekan tombol ESC untuk kembali...
```

```
D:\OneDrive - Institut Teknologi  X  +  v  -  □  X

#####
#                                     #
#           Warung Makan Masa Depan           #
#           by : Mirai                       #
#                                     #
#####

===== Foods =====
Nama           Harga
Fried Rice     20000
Tekan ESC untuk kembali...|
```

Data telah terhapus dengan sukses.

- Penjelasan Source Code

1.

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

#define MAX_SIZE 100
#define ANSI_COLOR_RED "\x1b[31m"
#define ANSI_COLOR_BOLD "\x1b[1m"
#define ANSI_COLOR_RESET "\x1b[0m"
```

Berikut adalah library dan beberapa define yang saya gunakan.

2.

```
typedef struct
{
    char namaProduk[MAX_SIZE];
    int hargaProduk;
} produk;

typedef struct
{
    char namaKategori[MAX_SIZE];
    produk products[MAX_SIZE];
    int indexProduk;
} kategori;

typedef struct
{
    int indexKategori;
    kategori category[MAX_SIZE];
} dataContainer;
```

Berikut adalah struct yang saya gunakan untuk menyimpan database:

1. Struct produk berfungsi untuk menyimpan nama produk dan harga produk.
2. Struct kategori berfungsi untuk menyimpan nama kategori, array dari produk yang berada dalam kategori tersebut, dan menyimpan panjang dari index produk itu sendiri dalam variabel indexProduk.
3. Struct dataContainer berfungsi untuk menyimpan array dari struct kategori dan panjang dari struct kategori yaitu indexKategori.

3.

```
void clearScreen()
{
    system("cls");
}

void banner()
{
    printf(" #####\n");
    printf(" #                               #\n");
    printf(" #           Warung Makan Masa Depan           #\n");
    printf(" #                   by : Mirai                   #\n");
    printf(" #                               #\n");
    printf(" #####\n");
    printf("\n\n");
}

enum ArrowKeys
{
    UP_ARROW = 72,
    DOWN_ARROW = 80,
    LEFT_ARROW = 75,
    RIGHT_ARROW = 77,
    ENTER_KEY = 13,
    ESC_KEY = 27
};

void handleArrowKeyStartMenu(int arrowKey, int *pointer);
void insertData(dataContainer *container);
void displayData(dataContainer *container, int selectedIndex);
void updateData(dataContainer *container);
void deleteData(dataContainer *container);
void clearInputBuffer();
```

Disini hanya beberapa prototipe fungsi dan shortcut untuk memanggil beberapa *function key*

4.

```
void readFromFile(dataContainer *container)
{
    FILE *database = fopen("data.dat", "r");
    if (database == NULL)
    {
        printf("File database tidak bisa dibuka. Masukkan data terlebih dahulu.\n");
        return;
    }

    fread(&container->indexKategori, sizeof(int), 1, database);
    for (int i = 0; i < container->indexKategori; i++)
    {
        fread(container->category[i].namaKategori, sizeof(char), MAX_SIZE, database);
        fread(&container->category[i].indexProduk, sizeof(int), 1, database);
        for (int j = 0; j < container->category[i].indexProduk; j++)
        {
            fread(container->category[i].products[j].namaProduk, sizeof(char), MAX_SIZE, database);
            fread(&container->category[i].products[j].hargaProduk, sizeof(int), 1, database);
        }
    }
    fclose(database);
}
```

Fungsi `readFromFile` akan pertama kali membuka file “data.dat” dan membaca `indexKategori`, yaitu panjang dari data kategori yang akan dibaca. Lalu program akan loop dari 0 hingga panjang `indexKategori` dan memasukkan data tersebut ke dalam `dataContainer`, terdapat 2 loop yang digunakan, yang pertama untuk memasukkan kategori nya, lalu yang kedua untuk memasukkan produk-produk yang berada dalam kategori tersebut sesuai index produk masing-masing.

5.

```
void saveToFile(dataContainer *container)
{
    FILE *database = fopen("data.dat", "w");
    if (database == NULL)
    {
        printf("Error opening file for writing.\n");
        return;
    }
    fwrite(&container->indexKategori, sizeof(int), 1, database);
    for (int i = 0; i < container->indexKategori; i++)
    {
        fwrite(container->category[i].namaKategori, sizeof(char),
MAX_SIZE, database);
        fwrite(&container->category[i].indexProduk, sizeof(int),
1, database);
        for (int j = 0; j < container->category[i].indexProduk;
j++)
        {
            fwrite(container->category[i].products[j].namaProduk,
sizeof(char), MAX_SIZE, database);

fwrite(&container->category[i].products[j].hargaProduk,
sizeof(int), 1, database);
        }
    }
    fclose(database);
}
```

Fungsi ini berfungsi untuk menyimpan data yang sudah disimpan pada data container ke dalam file. Fungsi ini akan meng-overwrite data lama dengan data baru ("w" parameter) yang sudah dimodifikasi. Cara kerja nya kurang lebih sama dengan readFromFile, tetapi fungsi ini berguna untuk membaca dari dataContainer dan menyimpan nya ke file. Pertama program akan membaca panjang index kategori lalu di for-loop sepanjang indexKategori, dalam for-loop index kategori tersebut ada for-loop lagi yang akan memasukkan produk sesuai dengan kategori nya masing-masing.

6.

```
void printStartMenu(int selectedIndex)
{
    clearScreen();
    banner();
    char menuOptions[][10] = {"INSERT", "READ", "UPDATE",
"DELETE", "EXIT"};
    int numOptions = sizeof(menuOptions) /
sizeof(menuOptions[0]);
    for (int i = 0; i < numOptions; i++)
    {
        if (i == selectedIndex)
        {
            if (i == 4)
                printf(ANSI_COLOR_BOLD ANSI_COLOR_RED "["
ANSI_COLOR_RESET " " ANSI_COLOR_BOLD ANSI_COLOR_RED "%s"
ANSI_COLOR_RESET " " ANSI_COLOR_BOLD ANSI_COLOR_RED "]"
ANSI_COLOR_RESET " ", menuOptions[i]);
            else
                printf("[ %s ] ", menuOptions[i]);
        }
        else
            printf(" %s ", menuOptions[i]);
    }
    printf("\n\nGunakan <- atau -> untuk navigasi.\nTekan ESC
untuk keluar.\n");
}

void navigateStartMenu(dataContainer *myContainer)
{
    int selectedIndex = 0;
    printStartMenu(selectedIndex);

    int ch, pointer = 0;
    while ((ch = _getch()) != ESC_KEY)
    {
        if (ch == 0 || ch == 224)
        {
            int arrowKeyCode = _getch();
            handleArrowKeyStartMenu(arrowKeyCode, &pointer);
            if (arrowKeyCode == LEFT_ARROW || arrowKeyCode ==
RIGHT_ARROW)
            {

```



```

        selectedIndex = pointer;
        printStartMenu(selectedIndex);
    }
}
else if (ch == ENTER_KEY)
{
    switch (selectedIndex)
    {
        case 0:
            insertData(myContainer);
            break;
        case 1:
            displayData(myContainer, selectedIndex);
            break;
        case 2:
            updateData(myContainer);
            break;
        case 3:
            deleteData(myContainer);
            break;
        case 4:
            printf("Exiting...");
            exit(0);
        default:
            break;
    }
}
}

void handleArrowKeyStartMenu(int arrowKey, int *pointer)
{
    switch (arrowKey)
    {
        case LEFT_ARROW:
            (*pointer)--;
            if (*pointer < 0)
            {
                *pointer = 0;
            }
            break;
        case RIGHT_ARROW:
            (*pointer)++;

```

```

        if (*pointer > 4)
        {
            *pointer = 4;
        }
        break;
default:
    break;
}
}

```

Fungsi ini berfungsi untuk navigasi start menu menggunakan *arrow key*. Fungsi ini menggunakan “_getch” dari library <conio.h> untuk menerima input dari user.

7. Breakdown fungsi INSERT DATA

```

void insertData(dataContainer *myContainer)
{
    readFromFile(myContainer);
    displayData(myContainer, 0);
    char queryKategori[MAX_SIZE];
    printf("Masukkan nama kategori atau masukkan -1 untuk keluar:");
    scanf("%99[^\n]", queryKategori);
    clearInputBuffer();
    if (strcmp(queryKategori, "-1") == 0)
    {
        clearScreen();
        printStartMenu(0);
        return;
    }
}

```

Pertama, program akan menampilkan daftar menu, lalu user akan diminta untuk memasukkan nama kategori.

```

int saveIndexKategori = -1;
for (int i = 0; i < myContainer->indexKategori; i++)
{
    if (strcasecmp(queryKategori,
myContainer->category[i].namaKategori) == 0)
    {
        saveIndexKategori = i;
        break;
    }
}

```

Lalu program akan mencari nama kategori tersebut menggunakan `strcasecmp`, jika ketemu, maka `saveIndexKategori` akan di update dengan index yang benar.

```

        if (saveIndexKategori != -1) // kalo ketemu masukkan ke
        kategori tersebut
        {
            kategori *currentKategori =
&myContainer->category[saveIndexKategori];
            printf("Kategori %s ditemukan!\n", queryKategori);
            char queryProduk[MAX_SIZE];
            printf("Masukkan nama produk: ");
            scanf("%99[^\n]", queryProduk);
            clearInputBuffer();
            for (int i = 0; i < currentKategori->indexProduk; i++)
            {
                if (strcasecmp(queryProduk,
currentKategori->products[i].namaProduk) == 0) // cari produk,
jika ketemu arahkan user ke update
                {
                    printf("Produk sudah ada, untuk update gunakan
fitur UPDATE\n");
                    printf("Tekan tombol ESC untuk kembali...");
                    int ch;
                    while ((ch = _getch()) != ESC_KEY)
                    {
                        /* code */
                    }
                    printStartMenu(0);
                    return;
                }
            }
        }
    }

```

Lanjut kebawahnya, disini adalah snippet kode untuk pengecekan agar tidak ada item duplikat.

```

// kalo nyampe sini, produk tersebut belum ada, masukkan ke
database
    if (currentKategori->indexProduk < MAX_SIZE)
    {
        // kategori *currentProduct = currentKategori; //
        Pointer to the current category

strcpy(currentKategori->products[currentKategori->indexProduk].na
maProduk, queryProduk);

        printf("Harga produk: ");
        while (scanf("%d",
&currentKategori->products[currentKategori->indexProduk].hargaPro
duk) != 1)
        {
            printf("Masukkan angka yang valid!\n");
            clearInputBuffer();
            printf("Harga produk: ");
        }
        clearInputBuffer();

        // Increment dalam current kateogri
        currentKategori->indexProduk++;

        // Save to file
        saveToFile(myContainer);
        printf("Data produk berhasil dimasukkan.\n");
        printf("Tekan tombol ESC untuk kembali...");
    }
    else
    {
        printf("Kategori penuh! Tidak dapat menambahkan
produk lebih lanjut.\n");
        printf("Tekan tombol ESC untuk kembali...");
    }
}

```

Masih di dalam if saveIndexKategori != -1, jika database sudah penuh, program akan berhenti menerima input, tetapi jika masih ada tempat untuk menaruh produk, produk akan dimasukkan ke dalam database, nama produk akan dimasukkan ke dalam database menggunakan strcpy dan user akan diminta untuk memasukkan harga produk tersebut.

8.

```
else
{
    // kategori nya ga ketemu, bikin baru
    if (myContainer->indexKategori < MAX_SIZE)
    {
        kategori *newKategori =
&myContainer->category[myContainer->indexKategori];

        // Set nama kategori nya
        strcpy(newKategori->namaKategori, queryKategori);

        // masukkan detail produk
        printf("Nama produk: ");
        scanf("%99[^\n]",
newKategori->products[newKategori->indexProduk].namaProduk);
        clearInputBuffer();

        printf("Harga produk: ");
        while (scanf("%d",
&newKategori->products[newKategori->indexProduk].hargaProduk) !=
1)
        {
            printf("Masukkan angka yang valid!\n");
            clearInputBuffer();
            printf("Harga produk: ");
        }
        clearInputBuffer();

        // naikin index produk di newKategori
        newKategori->indexProduk++;

        // naikin index kategori
        myContainer->indexKategori++;

        // simpen ke file
        saveToFile(myContainer);

        printf("Data kategori dan produk baru berhasil
dimasukkan.\n");
        printf("Tekan tombol ESC untuk kembali.\n");
    }
    else
```

```
        {  
            printf("Database penuh! Tidak dapat menambahkan  
kategori baru.\n");  
            printf("Tekan tombol ESC untuk kembali.\n");  
        }  
    }  
}
```

Jika `saveIndexKategori == -1`, maka akan masuk ke dalam `else`, program akan meng-set pointer ke array paling akhir di kategori, dan menambahkan kategori dan produk baru ke dalam array tersebut. Lalu `indexKategori` akan diupdate value nya menggunakan `myContainer->indexKategori++`;

9. Breakdown fungsi DISPLAY DATA

```
void displayData(dataContainer *container, int selectedIndex)
{
    int ch;
    clearScreen();
    banner();
    readFromFile(container);
    for (int i = 0; i < container->indexKategori; i++)
    {
        printf("%s  %s  %s\n",  "=====",
container->category[i].namaKategori, "=====");
        printf("%-25s%-10s\n", "Nama", "Harga");
        for (int j = 0; j < container->category[i].indexProduk;
j++)
        {
            printf("%-25s%-10d\n",
container->category[i].products[j].namaProduk,
container->category[i].products[j].hargaProduk);
        }
    }

    if (selectedIndex == 1)
    {
        printf("\nTekan ESC untuk kembali...");
    }

    while (selectedIndex == 1 && (ch = _getch()) != ESC_KEY)
    {
        // Do nothing
    }
    if (selectedIndex == 1)
    {
        printStartMenu(1);
    }
}
```

Pada fungsi display data, data dari file akan di read menggunakan fungsi readFromFile dan disimpan ke program menggunakan struct dataContainer. Lalu program akan print kategori, dan produk sesuai panjang array nya masing-masing.

10. Breakdown fungsi UPDATE DATA

```
readFromFile(myContainer);  
char queryKategori[MAX_SIZE];  
displayData(myContainer, 3);  
printf("Masukkan nama kategori atau -1 untuk keluar: ");  
scanf("%99[^\n]", queryKategori);  
clearInputBuffer();  
if (strcmp(queryKategori, "-1") == 0)  
{  
    clearScreen();  
    printStartMenu(2);  
    return;  
}
```

Pertama user akan diminta untuk memasukkan nama kategori atau produk yang ingin diubah, atau jika ingin membatalkan, memasukkan -1.

```
// cek kalo kategori nya ada  
int saveIndexKategori = -1;  
for (int i = 0; i < myContainer->indexKategori; i++)  
{  
    if (strcasecmp(myContainer->category[i].namaKategori,  
queryKategori) == 0)  
    {  
        saveIndexKategori = i;  
        break;  
    }  
}
```

Program akan mengecek nama kategori dari user input menggunakan strcasecmp, jika ketemu, saveIndexKategori akan di update dengan index yang benar dan loop akan di break.


```

    if (saveIndexKategori == -1)
    {
        printf("Kategori tidak ditemukan!\n"); // gak ketemu
        langsung return
        printf("Tekan tombol ESC untuk kembali...");
        int ch;
        while ((ch = _getch()) != ESC_KEY)
        {
            /* code */
        }
        clearScreen();
        printStartMenu(2);
        return;
    }

```

Jika kategori tidak ditemukan, user akan di prompt untuk kembali ke start menu.

```

    printf("1. Update Nama Kategori\n");
    printf("2. Update Harga Produk\n");
    printf("3. Update Nama Produk\n");
    int input;
    printf("Masukkan pilihan: ");
    while (scanf("%d", &input) != 1 || (input != 1 && input != 2
    && input != 3))
    {
        printf("Input tidak valid!\n");
        printf("Masukkan pilihan: ");
        clearInputBuffer();
    }
    clearInputBuffer();

    kategori *currentKategori =
    &myContainer->category[saveIndexKategori];

```

Jika kategori ditemukan, user akan diberikan pilihan untuk melakukan 3 update, update nama kategori, update harga produk atau update nama produk. Setelah menerima input. Program akan set pointer dengan nama currentKategori yang menunjuk pada kategori yang sekarang ingin di update.

a. Breakdown Update Nama Kategori.

```
if (input == 1)
{
    printf("Nama kategori sekarang -> %s\n",
currentKategori->namaKategori);
    printf("Masukkan nama kategori baru: ");
    scanf("%99[^\n]", currentKategori->namaKategori);
    clearInputBuffer();
}
```

Program akan deklarasi ulang nama lama menjadi nama baru.

b. Breakdown Update Harga Produk.

```
else if (input == 2)
{
    // Update item price
    printf("Masukkan nama produk yang ingin diubah harga nya:
");

    char queryProduk[MAX_SIZE];
    scanf("%99[^\n]", queryProduk);
    clearInputBuffer();

    int saveProdukIndex = -1;
    for (int i = 0; i < currentKategori->indexProduk; i++)
    {
        if
(strcasecmp(currentKategori->products[i].namaProduk, queryProduk)
== 0)
        {
            saveProdukIndex = i;
            break;
        }
    }

    if (saveProdukIndex == -1)
    {
        printf("Produk tidak ditemukan pada kategori!\n");
        printf("Tekan tombol ESC untuk kembali...");
        int ch;
        while ((ch = _getch()) != ESC_KEY)
        {
            /* code */
        }
        printStartMenu(2);
    }
}
```

```

        return;
    }

    // Update the price
    printf("Masukkan harga baru: ");
    while (scanf("%d",
&currentKategori->products[saveProdukIndex].hargaProduk) != 1)
    {
        printf("Masukkan angka yang valid!\n");
        clearInputBuffer();
        printf("Masukkan harga baru: ");
    }
    clearInputBuffer();
}

```

Pertama user akan diminta untuk memasukkan nama produk yang ingin diubah harganya, lalu program akan mencari produk tersebut, jika tidak ditemukan, program akan langsung kembali ke start menu.

Tetapi jika produk ditemukan, program akan mendeklarasi ulang harga produk lama dengan produk baru.

c. Breakdown Update Nama Produk.

```
else if (input == 3)
{
    // Update item name
    printf("Masukkan nama produk saat ini: ");
    char queryProduk[MAX_SIZE];
    scanf("%99[^\n]", queryProduk);
    clearInputBuffer();
    int saveProdukIndex = -1;
    for (int i = 0; i < currentKategori->indexProduk; i++)
    {
        if
        (strcasecmp(currentKategori->products[i].namaProduk, queryProduk)
        == 0)
        {
            saveProdukIndex = i;
            break;
        }
    }

    if (saveProdukIndex == -1)
    {
        printf("Item %s tidak ditemukan pada kategori!\n",
        queryProduk);
        printf("Tekan tombol ESC untuk kembali...");
        int ch;
        while ((ch = _getch()) != ESC_KEY)
        {
            // loop nya jalan terus kalo bukan esc
        }
        printStartMenu(2);
        return;
    }
    else
    {
        // Update namanya
        printf("Nama lama -> %s\n",
        currentKategori->products[saveProdukIndex].namaProduk);
        printf("Masukkan nama baru: ");
        scanf("%99[^\n]",
        currentKategori->products[saveProdukIndex].namaProduk);
        clearInputBuffer();
    }
}
```

```
}
```

Pertama program akan meminta nama lama dari produk yang akan diganti, jika tidak ada yang cocok dengan nama dari input, program akan output tidak ada item yang bernama seperti input.

Tetapi jika ada, program akan meminta dari user nama baru nya, lalu program akan mendeklarasikan ulang nama nya menjadi nama baru tersebut.

```
saveToFile(myContainer);  
printf("Data telah ter-update!\n");  
printf("Tekan tombol ESC untuk kembali...");  
int ch;  
while ((ch = _getch()) != ESC_KEY)  
{  
    /* code */  
}  
printStartMenu(2);  
return;
```

Setelah operasi selesai dilakukan, maka program akan menampilkan success message.

11. Breakdown fungsi DELETE DATA

```
void deleteData(dataContainer *myContainer)  
{  
    readFromFile(myContainer);  
    displayData(myContainer, 3);  
    char queryKategori[MAX_SIZE];  
    printf("Masukkan nama kategori atau -1 untuk keluar: ");  
    scanf("%99[^\n]", queryKategori);  
    clearInputBuffer();  
    if (strcmp(queryKategori, "-1") == 0)  
    {  
        clearScreen();  
        printStartMenu(3);  
        return;  
    }  
}
```

Pertama user akan diminta untuk memasukkan nama kategori.

```

// Check if the category exists
int saveIndexKategori = -1;
for (int i = 0; i < myContainer->indexKategori; i++)
{
    if (strcasecmp(myContainer->category[i].namaKategori,
queryKategori) == 0)
    {
        saveIndexKategori = i;
        break;
    }
}

if (saveIndexKategori == -1)
{
    printf("Kategori %s tidak ditemukan.\n", queryKategori);
    printf("Tekan tombol ESC untuk kembali...\n");
    int ch;
    while ((ch = _getch()) != ESC_KEY)
    {
        /* code */
    }
    printStartMenu(3);
    return;
}

```

Jika kategori tidak ditemukan, program akan kembali ke start menu.

```

// kategori nya ketemu
printf("1. Hapus isi seluruh kategori\n");
printf("2. Hapus produk dalam kategori\n");
int input;
printf("Masukkan pilihan: ");
while (scanf("%d", &input) != 1 || (input != 1 && input !=
2))
{
    printf("Input tidak valid!\n");
    printf("Masukkan pilihan: ");
    clearInputBuffer();
}
clearInputBuffer();

```

Program akan meminta untuk memasukkan pilihan 1 atau 2. 1 untuk menghapus kategori beserta isinya. 2 untuk menghapus 1 produk dalam kategori.

- a. Hapus isi seluruh kategori.

```
switch (input)
{
case 1:
    // Delete entire category
    // geser setiap kategori naik 1 index
    for (int i = saveIndexKategori; i <
myContainer->indexKategori - 1; i++)
    {
        myContainer->category[i] = myContainer->category[i +
1];
    }
    myContainer->indexKategori--;
    break;
```

Pada kode ini, kategori setelah kategori yang ingin dihapus akan dinaikkan index nya, sehingga secara tidak langsung menghapus data yang diinginkan. Lalu counter indexKategori akan dikurangi 1.

- b. Hapus produk dalam kategori.

```
case 2:
    // hapus spesifik item dalam kategori
    printf("Kategori terpilih -> %s\n",
myContainer->category[saveIndexKategori].namaKategori);
    printf("Masukkan nama produk yang ingin dihapus: ");
    char queryProduk[MAX_SIZE];
    scanf("%99[^\n]", queryProduk);
    clearInputBuffer();
    int saveProdukIndex = -1;
    kategori *currentKategori =
&myContainer->category[saveIndexKategori];
    for (int i = 0; i < currentKategori->indexProduk; i++)
    {
        if
(strcasecmp(currentKategori->products[i].namaProduk, queryProduk)
== 0)
        {
            saveProdukIndex = i;
            break;
        }
    }

    if (saveProdukIndex == -1)
    {
```

```

        printf("Produk %s tidak ditemukan.\n", queryProduk);
        printf("Tekan tombol ESC untuk kembali...");
        int ch;
        while ((ch = _getch()) != ESC_KEY)
        {
            /* code */
        }
        printStartMenu(3);
        return;
    }

    // Delete the item
    // geser produk setelah index produk yang dihapus menjadi
+1 nya
    for (int i = saveProdukIndex; i <
currentKategori->indexProduk - 1; i++)
    {
        currentKategori->products[i] =
currentKategori->products[i + 1];
    }
    currentKategori->indexProduk--;
    break;

```

Pertama user akan diminta untuk input nama produk yang ingin dihapus, jika nama produk tidak cocok dengan apapun dalam kategori tersebut, produk akan mengeluarkan error message produk tidak ditemukan. Tetapi jika ditemukan, produk yang dihapus akan ditimpa dengan produk indeks berikutnya, dan seterusnya sehingga produk akan terhapus dari database.

```

    saveToFile(myContainer);

    printf("Data telah terhapus!\n");
    printf("Tekan tombol ESC untuk kembali...");
    int ch;
    while ((ch = _getch()) != ESC_KEY)
    {
        /* code */
    }
    printStartMenu(3);
    return;

```

Program akan menyimpan data yang sudah diubah ke database dan mengeluarkan success message.

12. Penjelasan Fungsi main()

```
int main()
{
    clearScreen();
    dataContainer currentContainer;
    currentContainer.indexKategori = 0;
    navigateStartMenu(&currentContainer);

    return 0;
}
```

Saat pertama kali dijalankan, dataContainer akan kosong. dataContainer akan diisi saat akan melakukan operasi INSERT READ UPDATE atau DELETE.

Sekian Terima Kasih.