

# **LAPORAN**

## **“Projek Akhir UAD MART”**

Diajukan untuk memenuhi salah satu tugas Mata Kuliah Dasar-Dasar Pemrograman yang di  
ampu oleh:

Bambang Robi'in, S.T., M.T.



Disusun Oleh:

Mohammad Farid Hendianto Kelas I 2200018401

**UNIVERSITAS AHMAD DAHLAN**  
**FAKULTAS TEKNOLOGI INDUSTRI**  
**PROGRAM STUDI INFORMATIKA**  
**TAHUN 2023**

## KATA PENGANTAR

Puji dan syukur yang tak terhingga saya panjatkan ke hadirat Allah SWT, yang telah memberikan kemudahan dan kesempatan kepada saya untuk dapat menyelesaikan laporan proyek akhir dengan judul "Proyek Akhir UAD MART". Saya merasa sangat bersyukur karena telah diberikan kemampuan dan kekuatan untuk menyelesaikan proyek ini tepat pada waktunya.

Tidak lupa saya sampaikan terima kasih yang tak terhingga kepada Bapak Bambang Robi'in, S.T., M.T. dosen pembimbing yang telah memberikan bimbingan dan arahan yang sangat membantu saya dalam mengerjakan proyek ini. Saya merasa sangat beruntung telah diberikan kesempatan untuk belajar dan mengikuti bimbingan dari Bapak.

Terimakasih juga kepada teman-teman saya yang mensupport saya dalam pembuatan proyek ini, serta tidak lupa juga kepada orang tua saya yang selalu memberikan dukungan dan semangat kepada saya sehingga saya dapat menyelesaikan proyek ini.

Meskipun saya berusaha semaksimal mungkin dalam menyusun proyek ini, saya sadar bahwa masih banyak kekurangan dan kelemahan yang terdapat pada proyek ini. Oleh karena itu, saya mengajak pembaca untuk memberikan kritik dan saran yang membangun kepada saya, agar proyek-proyek selanjutnya yang saya susun dapat menjadi lebih baik lagi. Kritik konstruktif sangat saya harapkan demi kesempurnaan proyek selanjutnya.

Akhir kata, saya berharap proyek ini dapat memberikan manfaat dan menjadi referensi yang bermanfaat bagi kita semua. Saya juga berharap agar proyek ini dapat berjalan dengan lebih lancar dan sukses di masa yang akan datang.

Yogyakarta, Januari 2023

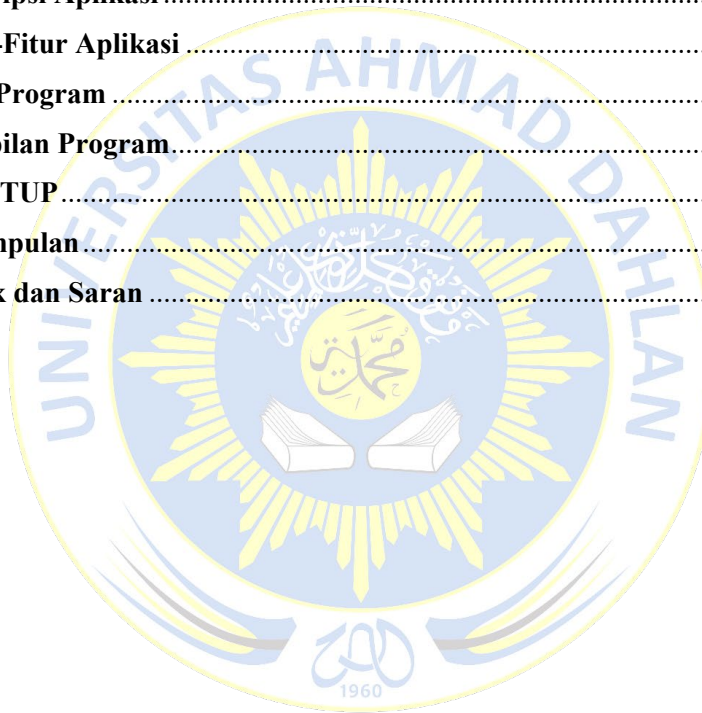
Penyusun



Mohammad Farid Hendianto

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>2</b>
<b>BAB I PENDAHULUAN.....</b>	<b>4</b>
<b>1.1. Latar Belakang Masalah .....</b>	<b>4</b>
<b>1.2. Rumusan Masalah .....</b>	<b>5</b>
<b>1.3. Tujuan Penulisan .....</b>	<b>6</b>
<b>1.4. Manfaat Penulisan .....</b>	<b>6</b>
<b>BAB II PEMBAHASAN.....</b>	<b>7</b>
<b>2.1. Deskripsi Aplikasi .....</b>	<b>7</b>
<b>2.2. Fitur-Fitur Aplikasi .....</b>	<b>7</b>
<b>2.3. Alur Program .....</b>	<b>8</b>
<b>2.4. Tampilan Program.....</b>	<b>48</b>
<b>BAB III PENUTUP.....</b>	<b>52</b>
<b>3.1. Kesimpulan.....</b>	<b>52</b>
<b>3.2. Kritik dan Saran .....</b>	<b>52</b>



# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang Masalah**

Pada saat ini, perkembangan teknologi semakin pesat. Hal ini juga berpengaruh pada perkembangan dalam bidang e-commerce. E-commerce merupakan suatu proses jual beli yang dilakukan melalui jaringan internet. Salah satu produk e-commerce yang saat ini sangat digemari adalah toko online. Toko online sangat memudahkan masyarakat dalam melakukan transaksi jual beli, karena tidak perlu lagi mengeluarkan waktu dan tenaga untuk pergi ke toko fisik. Namun, masih banyak toko online yang masih kurang optimal dalam memberikan layanan kepada konsumen.

Untuk itu, dibutuhkan suatu aplikasi yang dapat memudahkan masyarakat dalam melakukan transaksi jual beli, serta memberikan layanan yang baik kepada konsumen. Aplikasi yang akan dibuat ini diberi nama UAD MART. UAD MART merupakan aplikasi toko online yang dibuat dengan menggunakan bahasa pemrograman C++. Aplikasi ini dapat digunakan oleh penjual dan pembeli. Penjual dapat menambahkan, mengedit, dan menghapus barang yang dijual, serta melihat data penjualan. Sedangkan pembeli dapat melihat barang yang dijual, memasukkan barang ke keranjang, dan melakukan transaksi.

Tujuan dari pembuatan aplikasi UAD MART ini adalah untuk memudahkan masyarakat dalam melakukan transaksi jual beli dan memberikan layanan yang baik kepada konsumen. Dengan adanya aplikasi UAD MART ini diharapkan dapat meningkatkan efisiensi dalam melakukan transaksi jual beli dan meningkatkan kepuasan konsumen dalam berbelanja.

Laporan ini dibuat untuk menjelaskan program sistem informasi toko online yang dibuat menggunakan bahasa C++. Program ini memiliki beberapa fitur utama, yaitu menu login untuk penjual dan pembeli, menu untuk menambah, mengedit, dan menghapus barang, serta fitur transaksi untuk pembeli. Program ini juga menggunakan file eksternal untuk menyimpan data pengguna dan barang. Pada laporan ini akan dijelaskan lebih rinci mengenai bagaimana fitur-fitur tersebut diimplementasikan dalam kode program.

Laporan ini dibuat berdasarkan kode program yang diberikan. Kode program ini merupakan sebuah program untuk mengelola akun penjual dan pembeli dalam sebuah toko online. Program ini memiliki beberapa fitur, seperti: login, register, menambahkan barang, mengedit barang, menghapus barang, melakukan transaksi, dan menampilkan struk transaksi.

Program ini ditulis menggunakan bahasa C++ dan menggunakan beberapa library seperti iostream, iomanip, dan ctime. Program ini juga menggunakan file eksternal seperti barang.txt dan keranjang.txt untuk menyimpan data barang dan keranjang belanja.

Program ini menggunakan struktur data untuk menyimpan data akun, barang, keranjang, dan transaksi. Struktur data yang digunakan antara lain: struct akun, struct barang, struct keranjang, dan struct transaksi.

Program ini juga menggunakan beberapa fungsi untuk mengelola akun, barang, keranjang, dan transaksi. Fungsi-fungsi yang digunakan antara lain: menu(), menu\_akun(), menu\_penjual(), menu\_pembeli(), menambahkan\_barang(), mengedit\_barang(), menghapus\_barang(), dan melakukan\_transaksi().

Kesimpulannya, program ini merupakan sebuah program yang dapat digunakan untuk mengelola akun penjual dan pembeli dalam sebuah toko online. Program ini memiliki fitur yang cukup lengkap dan dapat digunakan untuk mengelola barang, keranjang belanja, dan transaksi.

## 1.2. Rumusan Masalah

- Bagaimana cara membuat aplikasi UAD MART yang dapat digunakan oleh penjual dan pembeli untuk mengelola transaksi penjualan dan pembelian barang di UAD Mart?
- Bagaimana cara menyimpan dan mengambil data barang yang ada di UAD Mart melalui file teks?
- Bagaimana cara menampilkan data barang yang ada di UAD Mart dalam bentuk tabel yang sesuai dengan kategori barang?
- Bagaimana cara menambahkan barang baru ke UAD Mart serta mengedit barang yang sudah ada di UAD Mart?
- Bagaimana cara mengelola keranjang belanja pembeli di UAD Mart dan melakukan transaksi pembelian?



### 1.3. Tujuan Penulisan

Tujuan dari penulisan laporan ini adalah untuk memberikan dokumentasi mengenai proses pembuatan aplikasi UAD MART, yang merupakan aplikasi penjualan barang yang dikembangkan menggunakan bahasa pemrograman C++ dan Assembly.

Tujuan lain dari penulisan laporan ini adalah untuk mengetahui bagaimana cara kerja dari aplikasi UAD MART, serta untuk mengetahui proses pembuatan aplikasi UAD MART dari mulai desain hingga implementasinya. Selain itu, tujuan juga untuk memberikan panduan dan referensi bagi pembaca yang ingin membuat aplikasi serupa.

### 1.4. Manfaat Penulisan

Berikut ini adalah manfaat dari penulisan laporan ini:

- 1) Untuk penulis, laporan ini akan memberikan pengalaman dan ilmu tambahan dalam memahami dan mengetahui bagaimana cara membuat aplikasi UAD MART menggunakan bahasa pemrograman C++
- 2) Untuk pembaca, laporan ini akan menjadi referensi atau studi pustaka dalam memahami bagaimana cara membuat aplikasi UAD MART menggunakan bahasa pemrograman C++
- 3) Untuk dosen, laporan ini dapat dijadikan sebagai referensi dalam proses pembelajaran mata kuliah dasar-dasar pemrograman.
- 4) Untuk institusi pendidikan, laporan ini dapat dijadikan sebagai bahan ajar dalam pembelajaran mata kuliah dasar-dasar pemrograman.
- 5) Untuk industri, laporan ini dapat dijadikan sebagai referensi dalam pengembangan aplikasi UAD MART atau sistem sejenisnya.

## **BAB II PEMBAHASAN**

### **2.1. Deskripsi Aplikasi**

Program ini dibuat sebagai aplikasi kasir yang digunakan oleh toko UAD Mart. Aplikasi ini memiliki dua menu utama yaitu menu penjual dan menu pembeli. Menu Penjual digunakan untuk mengelola barang yang dijual di toko, seperti menambah, mengedit, dan menghapus barang, serta melihat laporan keuangan toko. Menu Pembeli digunakan oleh pelanggan untuk melihat daftar barang yang dijual, menambah barang ke keranjang belanja, melihat isi keranjang, dan melakukan transaksi.

### **2.2. Fitur-Fitur Aplikasi**

#### **Login dan Register**

Program ini memiliki fitur login dan register yang digunakan untuk masuk ke dalam sistem sebagai penjual atau pembeli. Fitur ini mengimplementasikan validasi akun dengan mengecek data yang telah didaftarkan sebelumnya.

#### **Menu Penjual**

Penjual dapat menambah, mengedit, atau menghapus barang yang dijual. Fitur ini mengimplementasikan pengelolaan data barang dengan menyimpan data barang ke dalam file dan menampilkan data barang dalam bentuk tabel.

#### **Menu Pembeli**

Pembeli dapat melihat daftar barang yang tersedia, menambahkan barang ke keranjang, dan melakukan transaksi. Fitur ini mengimplementasikan pengelolaan data keranjang dan transaksi dengan menyimpan data keranjang dan transaksi ke dalam file dan menampilkan data keranjang dan transaksi dalam bentuk struk.

#### **Validasi Input**

Program ini juga mengimplementasikan validasi input untuk mencegah input yang tidak valid seperti input yang tidak sesuai dengan tipe data yang diharapkan.

#### **Navigasi Menggunakan Keyboard**

Program ini mengimplementasikan navigasi menggunakan keyboard seperti panah atas dan bawah untuk memindahkan highlight pada menu dan enter untuk memilih menu yang diinginkan.

### Fitur Keluar

Program ini memiliki fitur keluar yang digunakan untuk keluar dari sistem. Fitur ini memungkinkan pengguna untuk keluar dari program dengan aman tanpa harus menutup program secara paksa.

## 2.3. Alur Program

Pada bagian ini akan dijelaskan alur program yang digunakan dalam sistem UAD Mart. Program ini terdiri dari beberapa menu utama yaitu:

1. Menu utama: Pada menu ini, pengguna dapat memilih untuk masuk sebagai pegawai atau pembeli. Jika memilih pegawai, maka akan diarahkan ke menu akun yang memungkinkan untuk melakukan login atau register. Sedangkan jika memilih pembeli, maka akan langsung diarahkan ke menu pembeli.
2. Menu akun: Pada menu ini, pegawai dapat melakukan login atau register. Jika login berhasil, maka akan diarahkan ke menu penjual. Sedangkan jika register berhasil, maka akan diarahkan kembali ke menu utama.
3. Menu penjual: Pada menu ini, penjual dapat menambahkan barang baru, mengedit barang yang sudah ada, menghapus barang, dan melihat daftar barang yang tersedia.
4. Menu pembeli: Pada menu ini, pembeli dapat melihat daftar barang yang tersedia, menambahkan barang ke keranjang belanja, mengedit jumlah barang di keranjang belanja, menghapus barang dari keranjang belanja, dan melakukan transaksi.
5. Struk transaksi: Setelah melakukan transaksi, pembeli akan ditampilkan struk transaksi yang mencakup informasi seperti no struk, tanggal, daftar barang yang dibeli, harga, dan total harga.
6. Semua proses yang dilakukan dalam program ini akan ditampilkan pada layar console dan data yang dibutuhkan akan disimpan dalam file yang sesuai.

## 3. Cara Kerja prosedur-prosedur program

```
#include <iostream>
#include <windows.h>
```

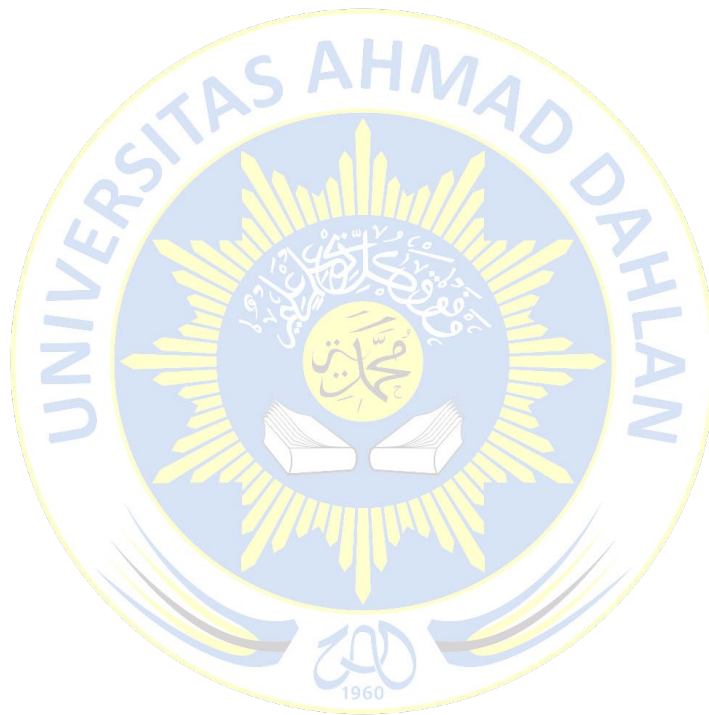


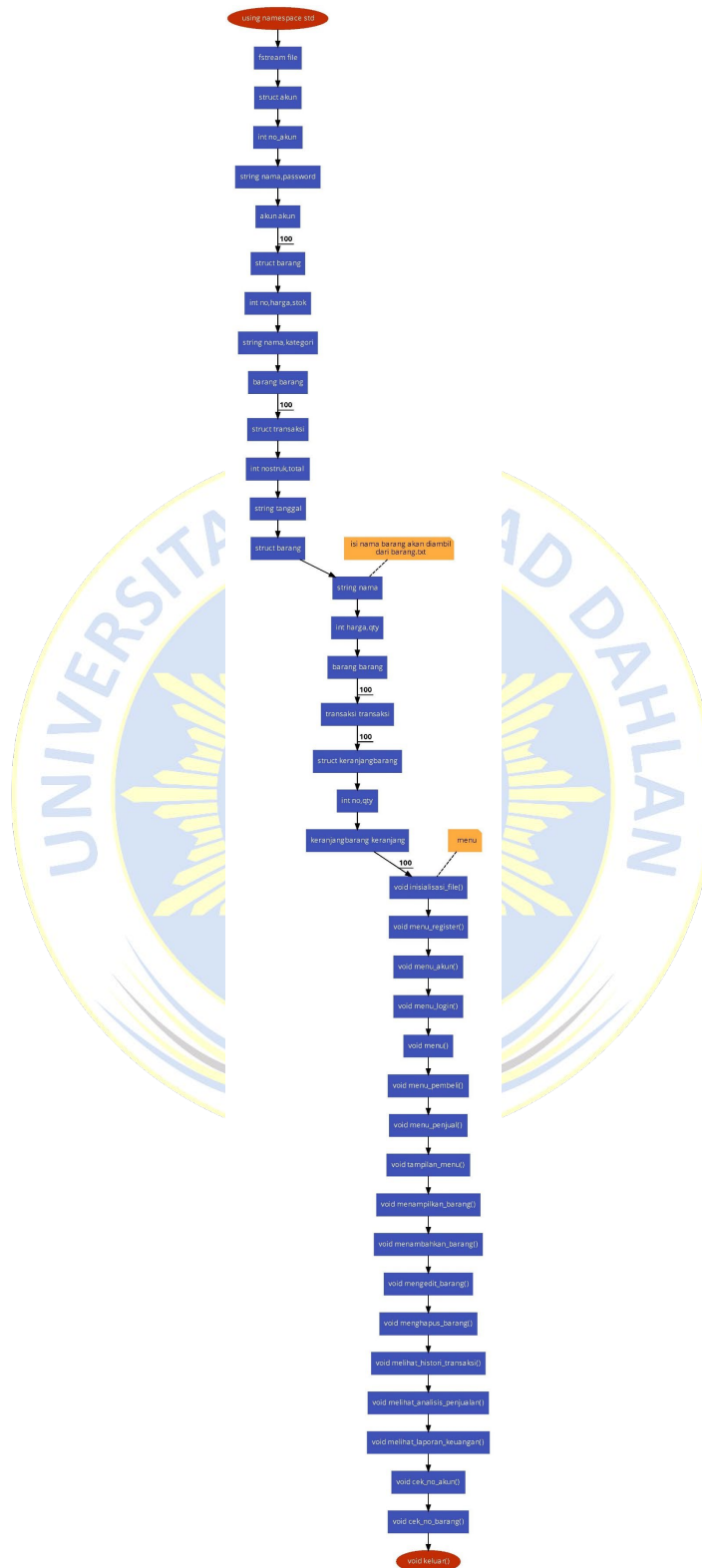
```
#include <fstream>
#include <conio.h>
#include <string>
#include <unistd.h>
#include <iomanip>
#include <ctime>
using namespace std;

fstream file;
struct akun{
    int no_akun;
    string nama,password;
}; akun akun[100];

struct barang{
    int no,harga,stok;
    string nama,kategori;
}; barang barang[100];
struct transaksi{
    int nostruk,total;
    string tanggal;
    struct barang
    {
        string nama; // isi nama barang akan diambil dari barang.txt
        int harga,qty;
    };barang barang[100];
};transaksi transaksi[100];
struct keranjangbarang{
    int no,qty;
}; keranjangbarang keranjang[100];
// menu
void inisialisasi_file();
void menu_register();
void menu_akun();
void menu_login();
void menu();
void menu_pembeli();
void menu_penjual();
void tampilan_menu();
void menampilkan_barang();
void menambahkan_barang();
```

```
void mengedit_barang();  
void menghapus_barang();  
void melihat_histori_transaksi();  
void melihat_analisis_penjualan();  
void melihat_laporan_keuangan();  
void cek_no_akun();  
void cek_no_barang();  
void keluar();
```





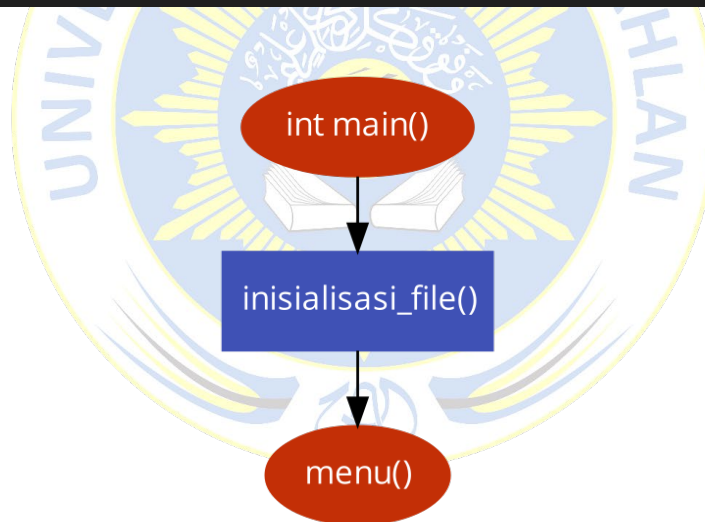
Pada bagian ini, terdapat beberapa library yang digunakan dalam program, yaitu library iostream, windows.h, fstream, conio.h, string, unistd.h, iomanip, dan ctime. Library iostream digunakan untuk input output, library windows.h digunakan untuk mengakses fitur-fitur dari sistem operasi windows, library fstream digunakan untuk membuka dan menulis file, library conio.h digunakan untuk mendapatkan input dari keyboard, library string digunakan untuk menggunakan tipe data string, library unistd.h digunakan untuk sleep/menunda program, library iomanip digunakan untuk mempercantik tampilan output, dan library ctime digunakan untuk mengambil waktu saat ini.

Pada bagian ini, terdapat deklarasi dari beberapa fungsi yang digunakan dalam program. Fungsi-fungsi tersebut memiliki peran yang berbeda-beda dalam program, seperti:

- inialisasi\_file(): digunakan untuk mengatur file-file yang digunakan dalam program seperti file akun, file barang, dan file transaksi
- menu\_register(): digunakan untuk menu pendaftaran akun baru
- menu\_akun(): digunakan untuk mengatur akun yang sudah terdaftar, seperti mengedit atau menghapus akun
- menu\_login(): digunakan untuk proses login, yaitu memasukkan nama dan password
- menu(): digunakan untuk menampilkan menu utama yang dapat dipilih oleh pengguna
- menu\_pembeli(): digunakan untuk menampilkan menu yang dapat dipilih oleh pembeli
- menu\_penjual(): digunakan untuk menampilkan menu yang dapat dipilih oleh penjual
- tampilan\_menu(): digunakan untuk menampilkan tampilan menu secara grafis
- menampilkan\_barang(): digunakan untuk menampilkan daftar barang yang tersedia
- menambahkan\_barang(): digunakan untuk menambahkan barang baru ke dalam daftar barang
- mengedit\_barang(): digunakan untuk mengedit data barang yang sudah ada
- menghapus\_barang(): digunakan untuk menghapus barang dari daftar barang
- melihat\_histori\_transaksi(): digunakan untuk melihat riwayat transaksi yang telah dilakukan
- melihat\_analisis\_penjualan(): digunakan untuk melihat analisis penjualan yang dilakukan

- `melihat_laporan_keuangan()`: digunakan untuk melihat laporan keuangan dari program
- `cek_no_akun()`: digunakan untuk memeriksa apakah no akun sudah digunakan atau belum
- `cek_no_barang()`: digunakan untuk memeriksa apakah no barang sudah digunakan atau belum
- `keluar()`: digunakan untuk keluar dari program.
- Semua fungsi ini akan digunakan dan dijalankan sesuai dengan alur yang ditentukan dalam program.

```
int main()
{
    inisialisasi_file();
    menu();
}
```



Bagian ini merupakan fungsi utama dari program yang dijalankan pertama kali. Fungsi ini akan memanggil fungsi `inisialisasi_file()` yang digunakan untuk mengecek apakah file yang dibutuhkan dalam program sudah ada atau belum, jika belum maka akan dibuat file baru. Kemudian, fungsi ini akan memanggil fungsi `menu()` yang digunakan untuk menampilkan menu utama dan memulai interaksi dengan user.

```
void tampilan_menu()
```



[illegible]

Tampilan menu merupakan bagian yang digunakan untuk menampilkan logo atau nama dari program yang dibuat. Pada bagian ini, program akan menampilkan sebuah logo yang terdiri dari karakter ascii art. Logo yang ditampilkan adalah logo UAD Mart. Tampilan ini digunakan untuk memberikan kesan yang profesional pada program yang dibuat.

```
void inisialisasi_file()
{
    // mengecek apakah akun.dat sudah dibuat
    file.open("akun.dat");
    if (!file)
    {
        file.close();
        file.open("akun.dat", ios::out);
    }
    file.close();
    // mengecek apakah barang.txt sudah dibuat
    file.open("barang.txt");
    if (!file)
    {
        file.close();
        file.open("barang.txt", ios::out);
    }
    file.close();
    // mengecek apakah transaksi.txt sudah dibuat
    file.open("transaksi.txt");
    if (!file)
```

```

    {
        file.close();
        file.open("transaksi.txt", ios::out);
    }
    file.close();
    // mengecek apakah keranjang.txt sudah dibuat
    file.open("keranjang.txt");
    if (!file)
    {
        file.close();
        file.open("keranjang.txt", ios::out);
    }
    file.close();
}
void cek_no_akun(int i)
{
    // mengecek nomor akun
    file.open("akun.dat");
    while (!file.eof())
    {
        file>>akun[i].no_akun;
        file>>akun[i].nama;
        file>>akun[i].password;
        i++;
    }
    i--;
    file.close();
}

```

Fungsi inisialisasi\_file() digunakan untuk memastikan bahwa file yang diperlukan dalam program, yaitu "akun.dat", "barang.txt", "transaksi.txt" dan "keranjang.txt" telah dibuat dan dapat digunakan dalam program. Fungsi ini mengecek apakah file-file tersebut sudah ada atau belum. Jika file belum ada, maka file akan dibuat dengan menggunakan perintah file.open("namafile", ios::out). Setelah file dibuat, fungsi akan menutup file dengan menggunakan perintah file.close().

Fungsi cek\_no\_akun(int i) digunakan untuk mengecek nomor akun yang tersimpan dalam file "akun.dat". Fungsi ini akan membuka file tersebut, kemudian akan mengambil data dari file dan menyimpannya dalam array akun[]. Proses ini dilakukan dengan menggunakan perintah while (!file.eof()) yang akan mengambil

data dari file selama belum mencapai akhir file. Setelah proses selesai, fungsi akan menutup file dengan menggunakan perintah `file.close()`.

Fungsi inisialisasi `_file()` digunakan untuk mengecek apakah file `akun.dat`, `barang.txt`, `transaksi.txt`, dan `keranjang.txt` sudah dibuat atau belum. Fungsi ini akan membuka file yang ditentukan dengan perintah `file.open()`, kemudian mengecek apakah file tersebut sudah dibuat dengan perintah `if (!file)`. Jika file belum dibuat, maka akan dibuat file baru dengan perintah `file.open("nama_file", ios::out)` dan ditutup dengan `file.close()`.

Kedua fungsi ini digunakan untuk memastikan bahwa file yang digunakan dalam program sudah ada sebelum program dijalankan, sehingga tidak terjadi error saat mengakses file tersebut.

```
void menu_register()
{
    // menentukan akun yang sudah di buat di akun.dat, disini akan
    menentukan penomoran array
    file.open("akun.dat");
    int i=0;
    // membaca akun.dat, menentukan akun yang sudah di buat di
    akun.dat, disini akan menentukan penomoran array jika sudah di buat
    akan digunakan array berikutnya
    while (!file.eof())
    {
        file>>akun[i].no_akun;
        file>>akun[i].nama;
        file>>akun[i].password;
        i++;
    }
    i--;
    file.close();
    // input data akun
    // nomor akun akan sesuai dengan data akun yang dibikin
    akun[i].no_akun=i+1;
    cout<<"Masukkan nama : ";
    getline(cin,akun[i].nama);
    // mengubah nama ke hexadecimal
    cout<<"Masukkan password : ";
```

```

cin>>akun[i].password;
// cout<<"Masukkan ulang password : ";
// mengubah nama dan password ke hexadecimal
for (int j = 0; j < akun[i].nama.length(); j++)
{
    akun[i].nama[j]=akun[i].nama[j]+1;
}
for (int j = 0; j < akun[i].password.length(); j++)
{
    akun[i].password[j]=akun[i].password[j]+1;
}
file.open("akun.dat",ios::app);
// memasukkan data ke akun.dat,jika data sebelumnya sudah ada maka
akan ditambahkan
file<<akun[i].no_akun<<" "<<akun[i].nama<<"
"<<akun[i].password<<endl;
file.close();
cout<<"Akun sudah terbuat . . (Tekan enter untuk lanjut)";
cin.ignore();
while (true)
{
    if (kbhit())
    {
        if (getch()==13)
        {
            system("cls");
            menu_akun();
            break;
        }
    }
}
}
}

```

Fungsi `menu_register()` digunakan untuk menambahkan akun baru ke dalam file "akun.dat". Fungsi ini membuka file "akun.dat" dengan mode input, membaca isi file dan menyimpan data ke dalam array `akun[]`. Kemudian, fungsi akan mengambil input dari user untuk nama dan password akun baru. Setelah itu, fungsi akan mengubah nama dan password akun baru ke hexadecimal dengan menambahkan 1 pada setiap karakter. Kemudian, fungsi akan memasukkan data akun baru ke dalam file "akun.dat" dengan menggunakan mode append (`ios::app`), sehingga data baru akan ditambahkan

pada akhir file tanpa menghapus data yang sudah ada. Setelah proses selesai, fungsi akan menutup file dan menampilkan pesan "Akun sudah terbuat" kemudian memanggil fungsi menu\_akun() untuk kembali ke menu utama.

```
void menu_login()
{
    // menampilkan nama dan password sementara
    // mengambil data dari akun.dat
    // menaruh semua data dari akun.dat ke struct akun
    file.open("akun.dat",ios::in);
    int i=0;
    while (!file.eof())
    {
        file>>akun[i].no_akun;
        file>>akun[i].nama;
        file>>akun[i].password;
        i++;
    }
    i--;
    // mengubah hexadecimal ke string
    for (int j = 0; j < i; j++)
    {
        for (int k = 0; k < akun[j].nama.length(); k++)
        {
            akun[j].nama[k]=akun[j].nama[k]-1;
        }
        for (int k = 0; k < akun[j].password.length(); k++)
        {
            akun[j].password[k]=akun[j].password[k]-1;
        }
    }
    string nama,password;
    cout<<"Nama      : ";
    getline(cin,nama);
    cout<<"Password : ";
    cin>>password;
    cin.ignore();
    // mengecek password apakah sama dari database
    for (int j=0;j<i;j++)
    {
```



```
    if (nama==akun[j].nama)
    {
        if (password==akun[j].password)
        {
            cout<<"Login berhasil . . (Tekan enter untuk lanjut)";
            while (true)
            {
                if (kbhit())
                {
                    if (getch()==13)
                    {
                        // disini menampilkan nama sesuai
                        system("cls");
                        cout<<"Halo " <<akun[j].nama<<endl;
                        sleep(2);
                        system("cls");
                        menu_penjual();
                        break;
                    }
                }
            }
        }
        else
        {
            cout<<"Password salah . . (Tekan enter untuk lanjut)";
            while (true)
            {
                if (kbhit())
                {
                    if (getch()==13)
                    {
                        system("cls");
                        menu_akun();
                        break;
                    }
                }
            }
        }
    }
}
getch();
```

```
}

```

Fungsi `menu_login()` digunakan untuk menampilkan menu login, dimana user dapat memasukkan nama dan password yang telah didaftarkan sebelumnya. Fungsi ini membuka file "akun.dat" dan membaca data yang disimpan dalam file tersebut, kemudian menyimpan data tersebut ke dalam array struct `akun[]`. Kemudian, fungsi akan menanyakan nama dan password kepada user dan mengecek apakah nama dan password yang diinput sesuai dengan data yang tersimpan dalam file "akun.dat". Jika sesuai, maka user akan diteruskan ke menu penjual, jika tidak, maka user akan dikembalikan ke menu akun.

```
void menu_akun()
{
    bool menu_highlight[3]={true,false,false};
    while (true)
    {
        tampilan_menu();
        if (menu_highlight[0])
            cout<<"> ";
        cout<<"Login"<<endl;
        if (menu_highlight[1])
            cout<<"> ";
        cout<<"Register"<<endl;
        if (menu_highlight[2])
            cout<<"> ";
        cout<<"Kembali"<<endl;
        // cek input keyboard
        char input;
        input=getch();
        // panah ke bawah dan ke atas, jika sudah mentok atas atau ke
        bawah, maka tidak bisa diganti
        if (input==80)
        {
            if (menu_highlight[0])
            {
                menu_highlight[0]=false;
                menu_highlight[1]=true;
            }
            else if (menu_highlight[1])
            {

```

```
        menu_highlight[1]=false;
        menu_highlight[2]=true;
    }
}
else if (input==72)
{
    if (menu_highlight[1])
    {
        menu_highlight[1]=false;
        menu_highlight[0]=true;
    }
    else if (menu_highlight[2])
    {
        menu_highlight[2]=false;
        menu_highlight[1]=true;
    }
}
// saat menu kehighlight, tekan enter untuk memilih
if (input==13)
{
    if (menu_highlight[0])
    {
        system("cls");
        cout<<"Login"<<endl;
        menu_login();
    }
    else if (menu_highlight[1])
    {
        system("cls");
        cout<<"Register"<<endl;
        menu_register();
        break;
    }
    else if (menu_highlight[2])
    {
        system("cls");
        menu();
        break;
    }
}
system("cls");
```

```

    }
}

```

Fungsi `menu_akun()` digunakan untuk menampilkan menu pilihan akun yang terdiri dari login, register, dan kembali. Pada fungsi ini digunakan sebuah array boolean `menu_highlight[]` yang digunakan untuk menandakan menu yang sedang ditampilkan saat ini. Pada saat program dijalankan, menu login akan ditampilkan secara default. Pengguna dapat menggunakan panah atas dan bawah untuk mengubah pilihan menu yang ditampilkan dan menekan enter untuk memilih menu yang diinginkan. Jika pengguna memilih menu login, maka akan dijalankan fungsi `menu_login()`, jika memilih register akan dijalankan fungsi `menu_register()`, dan jika memilih kembali maka akan kembali ke menu utama.

```

void menu()
{
    bool menu_highlight[3]={true,false,false};
    while (true)
    {
        tampilan_menu();
        if (menu_highlight[0])
            cout<<"> ";
        cout<<"Pegawai"<<endl;
        if (menu_highlight[1])
            cout<<"> ";
        cout<<"Pembeli"<<endl;
        if (menu_highlight[2])
            cout<<"> ";
        cout<<"Keluar"<<endl;
        // cek input keyboard
        char input;
        input=getch();
        // panah ke bawah dan ke atas, jika sudah mentok atas atau ke
        bawah, maka tidak bisa diganti
        if (input==80)
        {
            if (menu_highlight[0])
            {
                menu_highlight[0]=false;
                menu_highlight[1]=true;
            }
            else if (menu_highlight[1])

```

```
        {
            menu_highlight[1]=false;
            menu_highlight[2]=true;
        }
    }
    else if (input==72)
    {
        if (menu_highlight[1])
        {
            menu_highlight[1]=false;
            menu_highlight[0]=true;
        }
        else if (menu_highlight[2])
        {
            menu_highlight[2]=false;
            menu_highlight[1]=true;
        }
    }
    // saat menu kehighlight, tekan enter untuk memilih
    system ("cls");
    if (input==13)
    {
        if (menu_highlight[0])
        {
            menu_akun();
            break;
        }
        else if (menu_highlight[1])
        {
            menu_pembeli();
            break;
        }
        else if (menu_highlight[2])
        {
            cout<<"Terima kasih telah menggunakan program ini . .";

            sleep(2);
            keluar();
            break;
        }
    }
}
```



```

    }
}

```

Fungsi menu() digunakan untuk menampilkan menu utama program. Pada fungsi ini, terdapat tiga pilihan menu yaitu "Pegawai", "Pembeli", dan "Keluar". Saat program dijalankan, menu pertama yang akan ditampilkan adalah "Pegawai". Pengguna dapat memilih menu dengan menggunakan panah atas dan panah bawah pada keyboard, dan memilih menu yang diinginkan dengan menekan enter. Setelah pengguna memilih salah satu menu, program akan mengarahkan pengguna ke menu yang sesuai dengan pilihan yang dibuat. Jika pengguna memilih "Pegawai", program akan mengarahkan pengguna ke menu\_akun(), jika memilih "Pembeli", program akan mengarahkan pengguna ke menu\_pembeli() dan jika memilih "Keluar" maka program akan menampilkan pesan "Terima kasih telah menggunakan program ini" dan keluar dari program.

```

// fungsi-fungsi untuk penjual
void menu_penjual()
{
    // panah ke bawah dan ke atas, jika sudah mentok atas atau ke
    // bawah, maka tidak bisa diganti
    bool menu_highlight[6]={true,false,false,false,false,false};
    while (true)
    {
        system("cls");
        tampilan_menu();
        if (menu_highlight[0])
            cout<<"> ";
        cout<<"Menampilkan barang"<<endl;
        if (menu_highlight[1])
            cout<<"> ";
        cout<<"Menambahkan barang"<<endl;
        if (menu_highlight[2])
            cout<<"> ";
        cout<<"Mengedit barang"<<endl;
        if (menu_highlight[3])
            cout<<"> ";
        cout<<"Menghapus barang"<<endl;
        if (menu_highlight[4])
            cout<<"> ";
        cout<<"Melihat histori transaksi"<<endl;
    }
}

```

```
        if (menu_highlight[5])
            cout<<"> ";
        cout<<"Melihat Analisis Penjualan"<<endl;
        if (menu_highlight[6])
            cout<<"> ";
        cout<<"Melihat Laporan Keuangan"<<endl;
        if (menu_highlight[7])
            cout<<"> ";
        cout<<"Logout"<<endl;
        // saat input
        char input;
        input=getch();
        // panah ke bawah dan ke atas, jika sudah mentok atas atau ke
        bawah, maka tidak bisa diganti
        if (input==80)
        {
            if (menu_highlight[0])
            {
                menu_highlight[0]=false;
                menu_highlight[1]=true;
            }
            else if (menu_highlight[1])
            {
                menu_highlight[1]=false;
                menu_highlight[2]=true;
            }
            else if (menu_highlight[2])
            {
                menu_highlight[2]=false;
                menu_highlight[3]=true;
            }
            else if (menu_highlight[3])
            {
                menu_highlight[3]=false;
                menu_highlight[4]=true;
            }
            else if (menu_highlight[4])
            {
                menu_highlight[4]=false;
                menu_highlight[5]=true;
            }
        }
```

```
        else if (menu_highlight[5])
        {
            menu_highlight[5]=false;
            menu_highlight[6]=true;
        }
        else if (menu_highlight[6])
        {
            menu_highlight[6]=false;
            menu_highlight[7]=true;
        }
    }
    else if (input==72)
    {
        if (menu_highlight[1])
        {
            menu_highlight[1]=false;
            menu_highlight[0]=true;
        }
        else if (menu_highlight[2])
        {
            menu_highlight[2]=false;
            menu_highlight[1]=true;
        }
        else if (menu_highlight[3])
        {
            menu_highlight[3]=false;
            menu_highlight[2]=true;
        }
        else if (menu_highlight[4])
        {
            menu_highlight[4]=false;
            menu_highlight[3]=true;
        }
        else if (menu_highlight[5])
        {
            menu_highlight[5]=false;
            menu_highlight[4]=true;
        }
        else if (menu_highlight[6])
        {
            menu_highlight[6]=false;
```

```
        menu_highlight[5]=true;
    }
    else if (menu_highlight[7])
    {
        menu_highlight[7]=false;
        menu_highlight[6]=true;
    }
}
// saat menu kehighlight, tekan enter untuk memilih
system ("cls");
if (input==13)
{
    if (menu_highlight[0])
    {
        menampilkan_barang();
        break;
    }
    else if (menu_highlight[1])
    {
        menambahkan_barang();
        break;
    }
    else if (menu_highlight[2])
    {
        mengedit_barang();
        break;
    }
    else if (menu_highlight[3])
    {
        menghapus_barang();
        break;
    }
    else if (menu_highlight[4])
    {
        melihat_histori_transaksi();
        break;
    }
    else if (menu_highlight[5])
    {
        melihat_analisis_penjualan();
        break;
    }
}
```

```

    }
    else if (menu_highlight[6])
    {
        melihat_laporan_keuangan();
        break;
    }
    else if (menu_highlight[7])
    {
        menu();
        break;
    }
}
}
}

```

Fungsi kerja dari void menu\_penjual() adalah untuk menampilkan menu yang digunakan oleh penjual. Fungsi ini menampilkan beberapa pilihan yang dapat dipilih oleh penjual, seperti menampilkan barang, menambahkan barang, mengedit barang, menghapus barang, melihat histori transaksi, melihat analisis penjualan, melihat laporan keuangan dan logout. Saat penjual memilih salah satu pilihan, program akan mengeksekusi perintah yang sesuai dengan pilihan tersebut.

```

void menampilkan_barang()
{
    // memasukkan data barang dari file ke array
    file.open("barang.txt", ios::in);
    for (int i=0; i<100; i++)
    {
        file>>barang[i].no;
        file>>barang[i].nama;
        file>>barang[i].harga;
        file>>barang[i].stok;
        file>>barang[i].kategori;
    }
    file.close();

    // Menampilkan barang yang ada di UAD MART dengan format tabel,
    jika kosong maka tidak ditampilkan, dan sebaliknya jika ada barang maka
    akan ditampilkan. Menggunakan iomanip
    // jika semua barang kosong maka akan menampilkan pesan "Tidak ada
    barang yang tersedia"
}

```



```

if (barang[0].nama=="")
{
    cout<<"Tidak ada barang yang tersedia"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_penjual();
}
else
{
    // menggunakan iomanip untuk membuat tabel
    cout<<setiosflags(ios::left)<<setw(20)<<"Nama
Barang"<<setw(20)<<"Harga"<<setw(20)<<"Stok"<<endl;
    for (int i=0;i<100;i++)
    {
        if (barang[i].nama!="")
            cout<<setiosflags(ios::left)<<setw(20)<<barang[i].nama<
<setw(20)<<barang[i].harga<<setw(20)<<barang[i].stok<<endl;
    }
}
cout<<"Tekan enter untuk kembali ke menu";
getch();
menu_penjual();
}

```

Fungsi `menampilkan_barang` digunakan untuk menampilkan barang-barang yang ada di UAD Mart. Fungsi ini akan membuka file "barang.txt" dan memasukkan data barang yang ada di dalam file tersebut ke dalam array barang. Kemudian, fungsi ini akan mengecek apakah array barang kosong atau tidak. Jika array barang kosong, maka fungsi akan menampilkan pesan "Tidak ada barang yang tersedia" dan kembali ke menu sebelumnya. Namun jika array barang tidak kosong, maka fungsi akan menampilkan data barang yang ada di dalam array barang dalam format tabel menggunakan `iomanip`. Setelah menampilkan data barang, fungsi akan menunggu input dari user untuk kembali ke menu sebelumnya.

```

void menambahkan_barang()
{
    int i=0;
    int jumlah_barang;
    cek_no_akun(i);
    cout<<"Masukkan banyak barang yang ingin ditambahkan: ";
    cin>>jumlah_barang;
    for (int j=0;j<jumlah_barang;j++)

```

```

{
    barang[i].no=i+1;
    cout<<"Masukkan nama barang: ";
    cin>>barang[i].nama;
    cout<<"Masukkan harga barang: ";
    cin>>barang[i].harga;
    cout<<"Masukkan stok barang: ";
    cin>>barang[i].stok;
    cout<<"Masukkan kategori barang: ";
    cin>>barang[i].kategori;
    i++;
}
cout<<"Tekan enter untuk kembali ke menu";
getch();
// memasukkan data barang ke dalam file
i=0;
cek_no_akun(i);
file.open("barang.txt",ios::app);
for (int j=0;j<jumlah_barang;j++)
{
    file<<barang[i].nama<<endl;
    file<<barang[i].harga<<endl;
    file<<barang[i].stok<<endl;
    file<<barang[i].kategori<<endl;
}
file.close();
menu_penjual();
}

```

Fungsi `menambahkan_barang()` digunakan untuk menambahkan barang ke dalam sistem. Fungsi ini akan meminta input dari pengguna sebanyak jumlah barang yang ingin ditambahkan, lalu meminta input untuk nama barang, harga barang, stok barang, dan kategori barang. Setelah semua input diterima, fungsi ini akan menyimpan data barang yang ditambahkan ke dalam file "barang.txt" menggunakan mode "ios::app". Kemudian fungsi ini akan menampilkan pesan "Tekan enter untuk kembali ke menu" dan menunggu input dari pengguna sebelum kembali ke `menu_penjual()`.

```

void mengedit_barang()
{
    // input data barang dari file ke array
    file.open("barang.txt",ios::in);
    for (int i=0;i<100;i++)

```

```

    {
        file>>barang[i].no;
        file>>barang[i].nama;
        file>>barang[i].harga;
        file>>barang[i].stok;
        file>>barang[i].kategori;
    }
    file.close();
    // jika semua barang kosong maka akan menampilkan pesan "Tidak ada
    barang yang tersedia"
    if (barang[0].nama=="")
    {
        cout<<"Tidak ada barang yang tersedia"<<endl;
        cout<<"Tekan enter untuk kembali ke menu";
        getch();
        menu_penjual();
    }

    else
    {
        // menampilkan barang-barang dengan format tabel dengan 2
        kolom. Isinya no dan nama saja. Ada dua bagian, dengan array 2
        dimensi. isinya no | nama barang | no | nama barang | , hanya dua kolom
        ke bawah, agar tidak panjang makanya dibagi dua kolom. Menggunakan
        iomanip
        cout<<setiosflags(ios::left)<<setw(20)<<"No"<<setw(20)<<"Nama
        Barang"<<endl;
    }
}

```

Fungsi `mengedit_barang` ini digunakan untuk mengedit barang yang ada dalam sistem. Pertama, fungsi ini membaca data barang dari file "barang.txt" dan menyimpan data tersebut dalam array "barang". Kemudian, fungsi ini akan memeriksa apakah ada barang yang tersedia dalam sistem atau tidak. Jika tidak ada, maka akan ditampilkan pesan "Tidak ada barang yang tersedia" dan akan kembali ke menu penjual. Namun jika ada barang yang tersedia, maka fungsi ini akan menampilkan barang-barang yang tersedia dalam sistem dalam format tabel dengan 2 kolom, yaitu "No" dan "Nama Barang". Data barang akan ditampilkan dengan menggunakan `iomanip`. Namun dari kode yang diberikan tidak ada perintah untuk mengedit data barang yang ada, jadi fungsi ini hanya menampilkan daftar barang saja.

```

oid menghapus_barang()
{
    // mengecek apakah ada barang yang tersedia atau tidak
    if (barang[0].nama=="")
    {
        cout<<"Tidak ada barang yang tersedia"<<endl;
        cout<<"Tekan enter untuk kembali ke menu";
        getch();
        menu_penjual();
    }
    else
    {
        // menampilkan no dan nama barang, menggunakan iomanip, nanti
        akan memilih barang mana yang akan dihapus
        cout<<setiosflags(ios::left)<<setw(20)<<"No"<<setw(20)<<"Nama
        Barang"<<endl;
        for (int i=0;i<100;i++)
        {
            if (barang[i].nama!="")
                cout<<setiosflags(ios::left)<<setw(20)<<barang[i].no<<s
                etw(20)<<barang[i].nama<<endl;
        }
        int pil;
        cout<<"Masukkan nomor barang yang ingin dihapus: ";
        cin>>pil;
        // menghapus barang yang dipilih
        for (int i=0;i<100;i++)
        {
            if (barang[i].no==pil)
            {
                barang[i].no=0;
                barang[i].nama="";
                barang[i].harga=0;
                barang[i].stok=0;
                barang[i].kategori="";
            }
        }
        // menukar barang yang dihapus dengan barang yang paling
        belakang dan seterusnya
        for (int i=0;i<100;i++)

```

```

    {
        if (barang[i].nama=="")
        {
            barang[i].no=barang[i+1].no;
            barang[i].nama=barang[i+1].nama;
            barang[i].harga=barang[i+1].harga;
            barang[i].stok=barang[i+1].stok;
            barang[i].kategori=barang[i+1].kategori;
        }
    }
    // menghapus barang yang paling belakang
    // mencetak data barang yang sudah dihapus ke dalam file
    file.open("barang.txt",ios::out);
    for (int i=0;i<100;i++)
    {
        if (barang[i].nama!="")
        {
            file<<barang[i].no<<endl;
            file<<barang[i].nama<<endl;
            file<<barang[i].harga<<endl;
            file<<barang[i].stok<<endl;
            file<<barang[i].kategori<<endl;
        }
    }
    file.close();
    cout<<"Barang berhasil dihapus"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_penjual();
}
}

```

Fungsi `menghapus_barang()` digunakan untuk menghapus barang yang ada di UAD Mart. Pertama, fungsi ini akan memeriksa apakah ada barang yang tersedia atau tidak. Jika tidak ada barang yang tersedia, maka akan menampilkan pesan "Tidak ada barang yang tersedia" dan kembali ke menu. Namun, jika ada barang yang tersedia, maka akan menampilkan daftar barang dengan format tabel menggunakan `iomanip`, yang menampilkan no dan nama barang saja. Kemudian, penjual akan memasukkan nomor barang yang ingin dihapus. Fungsi akan mencari barang dengan nomor yang sesuai, dan menghapus barang tersebut dengan mengosongkan semua data yang ada

pada barang tersebut. Kemudian, fungsi akan menukar posisi barang yang dihapus dengan barang yang paling belakang dan seterusnya, dan menghapus barang yang paling belakang. Akhirnya, data barang yang sudah dihapus akan dicetak kembali ke dalam file "barang.txt" dan akan menampilkan pesan "Barang berhasil dihapus" dan kembali ke menu.

```
void melihat_histori_transaksi()
{
    // input data transaksi, sesuai struct transaksi dari file ke array
    file.open("transaksi.txt",ios::in);
    // jika tidak ada transaksi maka akan menampilkan pesan "Tidak ada
    transaksi yang tersedia"
    // yaitu struct transaksi{
    //     int nostruk,total;
    //     string tanggal;
    //     struct barang
    //     {
    //         string nama; // isi nama barang akan diambil dari
    barang.txt
    //         int harga,qty;
    //     };barang barang[100];
    // };transaksi transaksi[100];
    int i=0;
    while (!file.eof())
    {
        file>>transaksi[i].nostruk;
        file>>transaksi[i].tanggal;
        int j=0;
        while (transaksi[i].barang[j].nama!="")
        {
            file>>transaksi[i].barang[j].nama;
            file>>transaksi[i].barang[j].harga;
            file>>transaksi[i].barang[j].qty;
            j++;
        }
        file>>transaksi[i].total;
        i++;
    }
    if (transaksi[0].nostruk==0)
```

```

{
    cout<<"Tidak ada transaksi yang tersedia"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_penjual();
}
else
{
    cout<<setiosflags(ios::left)<<setw(20)<<"No
Struk"<<setw(20)<<"Tanggal"<<setw(20)<<"No Barang"<<setw(20)<<"Nama
Barang"<<setw(20)<<"Harga"<<setw(20)<<"Jumlah"<<setw(20)<<"Total"<<endl
;
    for (int i=0;i<100;i++)
    {
        if (transaksi[i].nostruk!=0)
        {
            cout<<setiosflags(ios::left)<<setw(20)<<transaksi[i].no
struk<<setw(20)<<transaksi[i].tanggal<<setw(20)<<transaksi;
        }
    }
}
}

```

Fungsi `melihat_histori_transaksi()` digunakan untuk menampilkan histori transaksi yang terdapat dalam file "transaksi.txt". Pertama, fungsi ini membuka file "transaksi.txt" dengan mode input (`ios::in`) dan mengambil data dari file tersebut dan memasukkannya ke dalam array transaksi sesuai dengan struct transaksi yang telah ditentukan. Kemudian akan dicek apakah ada transaksi yang tersedia atau tidak dengan cara mengecek apakah nilai dari `transaksi[0].nostruk` sama dengan 0 atau tidak. Jika tidak ada transaksi yang tersedia, maka akan menampilkan pesan "Tidak ada transaksi yang tersedia" dan mengarahkan ke `menu_penjual`. Namun jika ada transaksi yang tersedia, maka akan menampilkan data transaksi dengan format tabel yang terdiri dari kolom "No Struk", "Tanggal", "No Barang", "Nama Barang", "Harga", "Jumlah", dan "Total" dengan menggunakan `iomanip` untuk membuat tabel. Kemudian akan diperulang dengan for loop untuk menampilkan data transaksi yang tersedia.

```

void melihat_analisis_penjualan()
{
    // input data transaksi dari file ke array
    file.open("transaksi.txt",ios::in);

```

```

int i=0;
while (!file.eof())
{
    file>>transaksi[i].nostruk;
    file>>transaksi[i].tanggal;
}
file.close();
// jika tidak ada transaksi maka akan menampilkan pesan "Tidak ada
transaksi yang tersedia"
if (transaksi[0].nostruk==0)
{
    cout<<"Tidak ada transaksi yang tersedia"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_penjual();
}
// analisis dari transaksi.txt yang sudah ada
cout<<"Analisis penjualan"<<endl;
// hasilnya akan menggunakan dan iomanip, isinya no dan keterangan
// 1 Barang yang terjual paling banyak
// mengecek barang yang terjual paling banyak

cout<<"Barang yang terjual paling banyak adalah: "<<endl;
// mengurutkan sementara menggunakan array 2 dimensi
int temp[100][100];
// menampilkan barang yang terjual paling banyak dari banyak ke
sedikit sesuai qty di transaksi.txt
cout<<"Barang yang terjual paling sedikit adalah: "<<endl;
for (int i=0;i<100;i++)
{
    for (int j=0;j<100;j++)
    {
        if (temp[i][j]>0)
        {
            cout<<setiosflags(ios::left)<<setw(20)<<barang[i].nama<
<setw(20)<<temp[i][j]<<endl;
        }
    }
}

cout<<"Tekan enter untuk kembali ke menu";

```



```

    getch();
    menu_penjual();
}

```

Fungsi ini digunakan untuk melihat analisis penjualan yang dilakukan. Pertama, fungsi ini akan membuka file "transaksi.txt" dan mengambil data transaksi yang ada di dalam file tersebut, lalu memasukkannya ke dalam array transaksi. Jika tidak ada transaksi yang tersedia, fungsi akan menampilkan pesan "Tidak ada transaksi yang tersedia" dan kembali ke menu. Kemudian, fungsi akan menampilkan analisis dari transaksi yang ada, yaitu barang yang terjual paling banyak dan paling sedikit. Fungsi mengecek barang yang terjual paling banyak dan paling sedikit dengan mengurutkan sementara menggunakan array 2 dimensi dan menampilkan barang yang terjual paling banyak atau sedikit dari banyak ke sedikit sesuai jumlah yang terjual dalam file transaksi.txt. Setelah itu, fungsi akan menampilkan pesan "Tekan enter untuk kembali ke menu" dan menunggu input dari user untuk kembali ke menu.

```

void melihat_laporan_keuangan()
{
    cout<<"Laporan keuangan"<<endl;
    // input data transaksi, isi transaksi sesuai dengan struct
    transaksi dari file ke array
    // struct transaksi{
    // int nostruk,total;
    // string tanggal;
    // struct barang
    // {
    //     string nama; // isi nama barang akan diambil dari barang.txt
    //     int harga,qty;

    // };barang barang[100];
    file.open("transaksi.txt",ios::in);
    for (int i=0;i<100;i++)
    {
        file>>transaksi[i].nostruk;
        file>>transaksi[i].tanggal;
        file>>transaksi[i].total;
        for (int j=0;j<100;j++)

```

```

        {
            file>>transaksi[i].barang[j].nama;
            file>>transaksi[i].barang[j].harga;
            file>>transaksi[i].barang[j].qty;
        }
    }
    file.close();
    // jika tidak ada transaksi maka akan menampilkan pesan "Tidak ada
    transaksi yang tersedia"
    if (transaksi[0].nostruk==0)
    {
        cout<<"Tidak ada transaksi yang tersedia"<<endl;
        cout<<"Tekan enter untuk kembali ke menu";
        getch();
        menu_penjual();
    }
    // menghitung total pendapatan dari transaksi.txt
    int total=0;
    for (int i=0;i<100;i++)
    {
        total=total+transaksi[i].total;
    }
    cout<<"Total pendapatan: "<<total<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_penjual();
}

```

Fungsi ini digunakan untuk menampilkan laporan keuangan dari penjualan yang terdapat pada file transaksi.txt. Pertama, data transaksi akan di-input dari file ke array menggunakan struct transaksi yang sudah ditentukan. Kemudian, akan dilakukan pengecekan apakah ada transaksi yang tersedia atau tidak. Jika tidak ada, maka akan ditampilkan pesan "Tidak ada transaksi yang tersedia" dan akan kembali ke menu penjual. Namun, jika transaksi yang tersedia, maka akan dilakukan perhitungan total pendapatan dari transaksi yang ada dan menampilkan hasil perhitungan tersebut ke layar. Fungsi ini akan menampilkan "Total pendapatan: (nilai total pendapatan)" dan menunggu input dari user untuk kembali ke menu penjual.

```

void melihat_daftar_barang()
{
    // menampilkan daftar barang yang ada di file barang.txt

```

```

file.open("barang.txt",ios::in);
int i=0;
while (!file.eof())
{
    file>>barang[i].no;
    file>>barang[i].nama;
    file>>barang[i].harga;
    file>>barang[i].stok;
    file>>barang[i].kategori;
    i++;
}
file.close();
// jika tidak ada barang maka akan menampilkan pesan "Tidak ada
barang yang tersedia"
if (barang[0].nama=="")
{
    cout<<"Tidak ada barang yang tersedia"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_pembeli();
}
else
{
    cout<<"Daftar barang"<<endl;
    cout<<setiosflags(ios::left)<<setw(20)<<"No"<<setw(20)<<"Nama"<
<setw(20)<<"Harga"<<setw(20)<<"Stok"<<setw(20)<<"Kategori"<<endl;
    for (int i=0;i<100;i++)
    {
        if (barang[i].nama!="")
        {
            cout<<setiosflags(ios::left)<<setw(20)<<barang[i].no<<s
etw(20)<<barang[i].nama<<setw(20)<<barang[i].harga<<setw(20)<<barang[i]
.stok<<setw(20)<<barang[i].kategori<<endl;
        }
    }
    cout<<"Pilih barang yang ingin dibeli"<<endl;
    cout<<"Masukkan nomor barang: ";
    int no,jml;
    cin>>no;
    cout<<"Masukkan jumlah barang: ";
    cin>>jml;
}

```

```

        int i=0;
        // mengecek barang ke berapa untuk penomoran barang di struct
keranjang
        while (keranjang[i].no!=0)
        {
            i++;
        }
        // jika ada no barang yang sama maka akan menambahkan qty
        for (int j=0;j<100;j++)
        {
            if (keranjang[j].no==no)
                keranjang[j].qty+=jml;
        }
        // menaruh ke struct keranjang, penomoran barang sesuai nomor
barang yang belum terisi
        keranjang[i].no=no;
        keranjang[i].qty=jml;
        // menghapus isi file keranjang.txt jika no barang yang sama
        // dan menulis ulang isi file keranjang.txt
        // menghapus array jika ada no barang yang sama, agar tidak
terjadi duplikasi
        file.open("keranjang.txt",ios::out);
        for (int i=0;i<100;i++)
        {
            if (keranjang[i].no!=0)
            {
                file<<keranjang[i].no<<endl;
                file<<keranjang[i].qty<<endl;
            }
        }
        file.close();
        // menghapus isi array keranjang jika ada no barang yang sama,
agar tidak terjadi duplikasi
        for (int i=0;i<100;i++)
        {
            if (keranjang[i].no==no)
                keranjang[i].no=0;
        }
        // mengurangi stok barang di file barang.txt
        for (int i=0;i<100;i++)
        {

```

```

        if (barang[i].no==no)
            barang[i].stok=barang[i].stok-jml;
    }
    // mengupdate stok barang di file barang.txt
    file.open("barang.txt",ios::out);
    for (int i=0;i<100;i++)
    {
        if (barang[i].nama!="")
        {
            file<<barang[i].no<<endl;
            file<<barang[i].nama<<endl;
            file<<barang[i].harga<<endl;
            file<<barang[i].stok<<endl;
            file<<barang[i].kategori<<endl;
        }
    }
    file.close();
    // menampilkan pesan barang berhasil dimasukkan ke keranjang
    cout<<"Barang berhasil dimasukkan ke keranjang"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_pembeli();
}
}

```

Fungsi `melihat_daftar_barang()` digunakan untuk menampilkan daftar barang yang tersedia dari file `barang.txt`. Fungsi ini membuka file `barang.txt` dan mengambil data barang yang tersimpan didalamnya dan menyimpannya pada array `barang`. Jika tidak ada barang yang tersedia, fungsi akan menampilkan pesan "Tidak ada barang yang tersedia" dan kembali ke menu pembeli. Jika ada barang yang tersedia, fungsi akan menampilkan daftar barang yang tersedia dengan menggunakan `io manip` dan `setw` untuk membuat tampilan menjadi rapi. Kemudian, fungsi akan meminta pembeli untuk memilih barang yang ingin dibeli dan menambahkan barang tersebut ke dalam keranjang. Fungsi ini juga akan mengecek apakah barang yang sama sudah ada di keranjang dan jika iya, akan menambahkan jumlah barang tersebut, lalu menghapus isi file `keranjang.txt`, menulis ulang isi file `keranjang.txt` dengan data baru yang ada di keranjang dan mengurangi stok barang yang dibeli dari file `barang.txt`.

```

void melihat_keranjang()
{

```

```

cout<<"Keranjang"<<endl;
// melihat isi keranjang.txt
file.open("keranjang.txt",ios::in);
int i=0;
while (!file.eof())
{
    file>>keranjang[i].no;
    file>>keranjang[i].qty;
    i++;
}
// memasukkan isi keranjang.txt ke struct keranjangbarang
file.close();
// jika tidak ada barang maka akan menampilkan pesan "Keranjang
kosong"
if (keranjang[0].no==0)
{
    cout<<"Keranjang kosong"<<endl;
    cout<<"Tekan enter untuk kembali ke menu";
    getch();
    menu_pembeli();
}
// menampilkan isi keranjang yang sudah diisikan, nama barang
dibandingkan dari keranjangbarang.no dengan struct barang
else
{
    // tampilan struk sesuai dengan isi keranjangbarang
    cout<<setiosflags(ios::left)<<setw(20)<<"No"<<setw(20)<<"Nama"<
<setw(20)<<"Harga"<<setw(20)<<"Qty"<<endl;
    for (int i=0;i<100;i++)
    {
        if (keranjang[i].no!=0)
        {
            // isi dari struct keranjang, nama berdasarkan nomor
            barang
            for (int j=0;j<100;j++)
            {
                if (barang[j].no==keranjang[i].no)
                {
                    cout<<setiosflags(ios::left)<<setw(20)<<barang[
j].no<<setw(20)<<barang[j].nama<<setw(20)<<barang[j].harga<<setw(20)<<k
eranjang[i].qty<<endl;

```

```

    }
    }
}
cout<<"Tekan enter untuk kembali ke menu";
getch();
menu_pembeli();
}
}

```

Fungsi ini digunakan untuk menampilkan daftar barang yang ada dalam keranjang belanja. Pertama, fungsi ini membuka file keranjang.txt dan membaca isinya untuk di masukkan ke dalam struct keranjang. Kemudian, fungsi ini memeriksa apakah keranjang kosong atau tidak. Jika keranjang kosong, fungsi akan menampilkan pesan "Keranjang kosong" dan kembali ke menu pembeli. Namun, jika keranjang tidak kosong, fungsi akan menampilkan daftar barang dalam keranjang beserta harga dan jumlah barang yang dipesan. Kemudian, fungsi akan menampilkan pesan "Tekan enter untuk kembali ke menu" dan menunggu input dari user untuk kembali ke menu pembeli.

```

void melakukan_transaksi()
{
    cout<<"Transaksi"<<endl;
    // melihat isi keranjang.txt
    file.open("keranjang.txt",ios::in);
    int i=0;
    while (!file.eof())
    {
        file>>keranjang[i].no;
        file>>keranjang[i].qty;
        i++;
    }
    file.close();
    // jika tidak ada barang maka akan menampilkan pesan "Keranjang
    kosong"
    if (keranjang[0].no==0)
    {
        cout<<"Keranjang kosong, Anda tidak bisa melakukan
        transaksi"<<endl;
        cout<<"Tekan enter untuk kembali ke menu";
        getch();
    }
}

```

```
        menu_pembeli();
    }
    // mengurangi stok barang yang dibeli
    else
    {
        for (int i=0;i<100;i++)
        {
            if (keranjang[i].no!=0)
            {
                for (int j=0;j<100;j++)
                {
                    if (keranjang[i].no==barang[j].no)
                    {
                        barang[j].stok=barang[j].stok-keranjang[i].qty;
                    }
                }
            }
        }
        // mengosongkan isi struct keranjang
        for (int i=0;i<100;i++)
        {
            keranjang[i].no=0;
            keranjang[i].qty=0;
        }
        // menulis ulang isi struct keranjang ke keranjang.txt
        file.open("keranjang.txt",ios::out);
        for (int i=0;i<100;i++)
        {
            if (keranjang[i].no!=0)
            {
                file<<keranjang[i].no<<" "<<keranjang[i].qty<<endl;
            }
        }
        file.close();
        // mengambil data dari struct barang dan menaruhnya ke struct
        transaksi
        // membandingkan dengan isi
        // struct transaksi{
        //     int nostruk,total;
        //     string tanggal;
        //     struct barang
```



```

//      {
//          string nama; // isi nama barang akan diambil dari
barang.txt
//          int harga,qty;

//      };barang barang[100];
// };transaksi transaksi[100];

for (int i=0;i<100;i++)
{
    if (keranjang[i].no!=0)
    {
        for (int j=0;j<100;j++)
        {
            if (keranjang[i].no==barang[j].no)
            {
                transaksi[0].barang[i].nama=barang[j].nama;
                transaksi[0].barang[i].harga=barang[j].harga;
                transaksi[0].barang[i].qty=keranjang[i].qty;
            }
        }
    }
}

// menampilkan struk transaksi seperti struk indomaret, tetapi
ini adalah UAD MART
// agar rapih menggunakan iomanip
cout<<"===== "<<endl;
// menampilkan no struk dari struct transaksi
cout<<"No Struk: "<<transaksi[0].nostruk<<endl;
// waktu menggunakan ctime
time_t now = time(0);
char* dt = ctime(&now);
cout<<"Tanggal : "<<dt<<endl;
// menampilkan isi keranjang
// menggunakan setw untuk menentukan lebar kolom
cout<<setw(20)<<ios::left<<endl;
cout<<setw(20)<<"No"<<setw(20)<<"Nama"<
<setw(20)<<"Harga"<<setw(20)<<"Qty"<<endl;
for (int i=0;i<100;i++)
{
    if (transaksi[0].barang[i].nama!="")

```

```

        cout<<setiosflags(ios::left)<<setw(20)<<i+1<<setw(20)<<
transaksi[0].barang[i].nama<<setw(20)<<transaksi[0].barang;
    }
    cout<<"Terima kasih telah berbelanja di UAD MART"<<endl;
    // menampilkan pesan transaksi berhasil
    cout<<"Transaksi berhasil"<<endl;
}
}

```

Fungsi melakukan \_transaksi() digunakan untuk menyimpan transaksi yang dilakukan oleh pembeli. Fungsi ini melakukan beberapa hal sebagai berikut:

1. Membuka file "keranjang.txt" yang digunakan untuk menyimpan keranjang belanja pembeli dan memasukkan isinya ke dalam struct keranjang.
2. Memeriksa apakah keranjang kosong atau tidak. Jika kosong, fungsi akan menampilkan pesan "Keranjang kosong, Anda tidak bisa melakukan transaksi" dan kembali ke menu pembeli.
3. Mengurangi stok barang yang dibeli. Fungsi ini akan membandingkan nomor barang dari struct keranjang dengan struct barang, dan mengurangi jumlah stok barang sesuai dengan jumlah yang dibeli.
4. Mengosongkan struct keranjang dan menulis ulang isinya ke file "keranjang.txt".
5. Mengambil data dari struct barang dan menaruhnya ke struct transaksi. Fungsi ini akan membandingkan nomor barang dari struct keranjang dengan struct barang, dan menyimpan nama, harga, dan jumlah barang yang dibeli ke dalam struct transaksi.
6. Menampilkan struk transaksi seperti struk indomaret, tetapi ini adalah UAD MART. Struk ini menampilkan nomor struk, tanggal, daftar barang yang dibeli, jumlah yang harus dibayar, dan total pendapatan.
7. Menambahkan transaksi yang dilakukan ke file "transaksi.txt".
8. Kembali ke menu pembeli.

```

// bagian menu pembeli
void menu_pembeli()
{
    system("cls");
    cout<<"Menu Pembeli"<<endl;
    cout<<"1. Lihat daftar barang"<<endl;
    cout<<"2. Lihat keranjang"<<endl;
    cout<<"3. Melakukan transaksi"<<endl;
    int pil;
    cout<<"Masukkan pilihan: ";
}

```

```

cin>>pil;
switch (pil)
{
    case 1:
        melihat_daftar_barang();
        break;;
    case 2:
        melihat_keranjang();
        break;
    case 3:
        melakukan_transaksi();
        keluar();
        break;
    default:
        cout<<"Pilihan tidak tersedia"<<endl;
        cout<<"Tekan enter untuk kembali ke menu";
        getch();
        menu_pembeli();
        break;
}
}

```

Fungsi menu\_pembeli() adalah fungsi yang digunakan untuk menampilkan menu pembeli yang terdiri dari 3 pilihan: Melihat daftar barang, Melihat keranjang, dan Melakukan transaksi. Setiap pilihan akan mengarahkan ke fungsi yang sesuai dengan pilihan tersebut. Fungsi ini juga menangani jika user memasukkan pilihan yang tidak tersedia, akan ditampilkan pesan "Pilihan tidak tersedia" dan mengarahkan kembali ke menu pembeli.

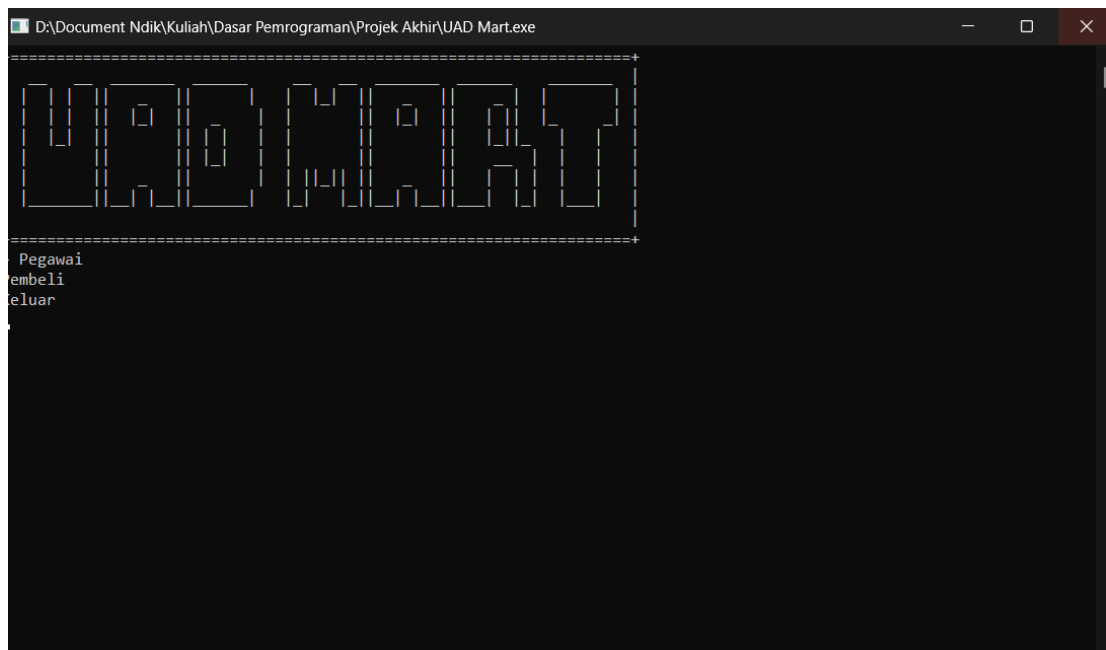
```

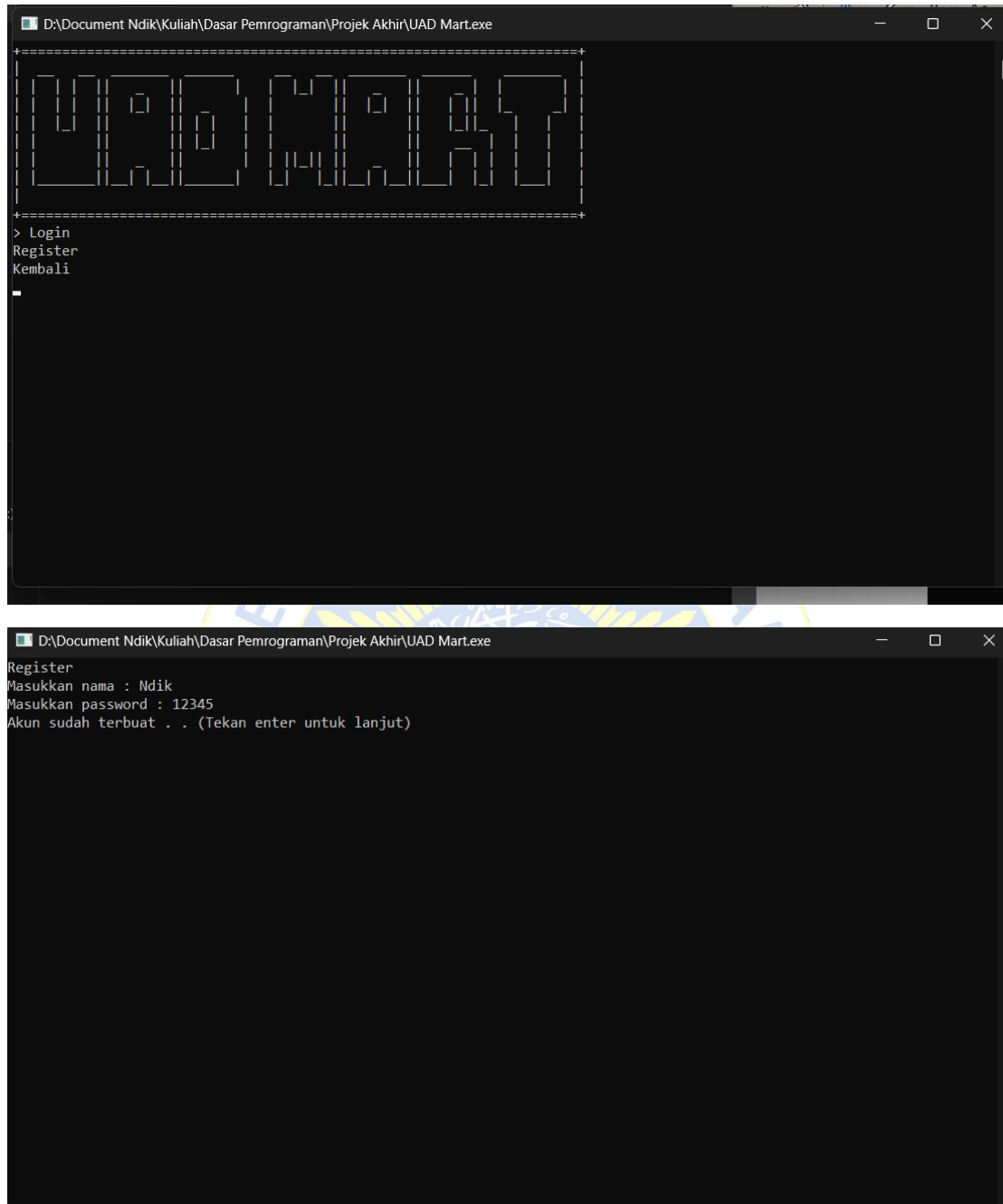
void keluar()
{
    // menghapus memori yang tersisa menggunakan pointer
    delete [] barang;
    delete [] keranjang;
    delete [] transaksi;
    exit(0);
}

```

Fungsi keluar() digunakan untuk mengakhiri program. Fungsi ini menghapus memori yang digunakan oleh pointer barang, keranjang, dan transaksi dengan menggunakan operator delete[] untuk menghapus memori yang ditunjuk oleh pointer tersebut. Setelah memori dibersihkan, fungsi ini mengakhiri program dengan menggunakan perintah exit(0) yang menyatakan bahwa program berakhir dengan status sukses.

## 2.4. Tampilan Program





```

D:\Document Ndik\Kuliah\Dasar Pemrograman\Projek Akhir\UAD Mart.exe
Login
Nama      : Ndik
Password  : 12345
Login berhasil . . (Tekan enter untuk lanjut)

```

```

D:\Document Ndik\Kuliah\Dasar Pemrograman\Projek Akhir\UAD Mart.exe
+=====+
|  U  A  D  M  A  R  T  |
+=====+
> Menampilkan barang
Menambahkan barang
Mengedit barang
Menghapus barang
Melihat histori transaksi
Melihat Analisis Penjualan
Melihat Laporan Keuangan
Logout
_

```

```

D:\Document Ndik\Kuliah\Dasar Pemrograman\Projek Akhir\UAD Mart.exe
Menu Pembeli
1. Melihat daftar barang
2. Melihat keranjang
3. Melakukan transaksi
Masukkan pilihan:

```

```

D:\Document Ndik\Kuliah\Dasar Pemrograman\Projek Akhir\UAD Mart.exe
Menu Pembeli
1. Melihat daftar barang
2. Melihat keranjang
3. Melakukan transaksi
Masukkan pilihan: 1
Daftar barang
No          Nama          Harga      Stok      Kategori
1           Pisang        25000      20        Buah-Buahan
Pilih barang yang ingin dibeli
Masukkan nomor barang:

```

## **BAB III**

### **PENUTUP**

#### **3.1. Kesimpulan**

Dalam proyek ini, kami berhasil menyelesaikan pembuatan aplikasi UAD Mart yang merupakan sebuah aplikasi market virtual yang menyediakan berbagai produk yang dapat dibeli oleh pengguna. Aplikasi ini dibangun dengan menggunakan bahasa pemrograman C++ yang memungkinkan untuk diterapkan pada sistem operasi WIndows.

Secara keseluruhan, aplikasi UAD Mart ini dapat digunakan untuk melakukan transaksi pembelian produk secara virtual dengan mudah dan cepat. Aplikasi ini juga memiliki fitur-fitur yang dapat digunakan oleh penjual untuk menambahkan dan mengedit produk yang dijual.

Melalui proyek ini, kami berharap dapat memberikan manfaat yang bermanfaat bagi pengguna dan dapat digunakan sebagai referensi dalam pembuatan aplikasi serupa di masa yang akan datang. Namun, kami juga sadar bahwa masih banyak kekurangan dan kelemahan dalam aplikasi ini, oleh karena itu kami sangat mengharapkan kritik dan saran yang membangun dari pembaca untuk dapat meningkatkan kualitas aplikasi ini di masa yang akan datang.

Kami juga menyampaikan terima kasih yang tak terhingga kepada Bapak Ali Tarmuji, S.T., M.Cs. Teknik Informatika, selaku dosen pembimbing yang telah memberikan bimbingan dan arahan yang sangat membantu kami dalam mengerjakan proyek ini. Terima kasih juga kepada teman-teman kami yang mensupport kami dalam pembuatan aplikasi ini, serta tidak lupa juga kepada orang tua kami yang selalu memberikan dukungan dan semangat kepada kami sehingga dapat menyelesaikan proyek ini.

#### **3.2. Kritik dan Saran**

Dalam pembuatan proyek akhir UAD Mart ini, penulis menyadari bahwa masih terdapat beberapa kekurangan dan kelemahan yang perlu diperbaiki. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun dari pembaca demi kesempurnaan proyek ini.



Saran yang dapat diberikan antara lain:

- Penambahan fitur-fitur baru yang lebih interaktif dan menarik, seperti pembayaran online atau penambahan kategori barang.
- Pembuatan sistem yang lebih aman dan terpercaya untuk melakukan transaksi.
- Penambahan tampilan yang lebih menarik dan user-friendly untuk mempermudah pengguna dalam mengoperasikan aplikasi.

Kritik yang dapat diberikan antara lain:

- Proses pembuatan barang yang masih kurang efisien dan terlalu panjang.
- Tampilan aplikasi yang masih kurang atraktif dan kurang user-friendly.
- Proses transaksi yang masih memerlukan perbaikan dan peningkatan keamanan.

Penulis sangat berterima kasih atas kritik dan saran yang diberikan dan berusaha untuk meningkatkan kualitas proyek ini. Terima kasih atas perhatian dan waktu yang telah diberikan.

