**LAPORAN PRAKTIKUM**

**PEMROGRAMAN BERORIENTASI OBJEK**



**Ahmad Sofian Aris S (21081010211)**

**Abdul Azis Naufal Farizqi (21081010210)**

**Kenandra Aurello (21081010184)**

**KELAS E081**

**PRODI INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UPN “VETERAN” JAWA TIMUR**

**SURABAYA**

**TAHUN 2023**

1. **PENJELASAN PROGRAM**
2. Bagian library

#include <vector> digunakan untuk menambah, menghapus, dan mengakses elemen-elemen vector. Penggunaan dalam program: menambah dan menghapus barang, menambah dan menghapus data pegawai.

1. Bagian Class

Ada 6 class, yaitu Item, Pegawai, Transaksi, Kasir, Admin dan Toko.

1. Class Item

Kelas ini menggambarkan barang yang dijual. Memiliki nama, harga, dan stok.

1. Class Pegawai

Kelas ini menggambarkan pegawai toko. Setiap pegawai memiliki nama dan jabatan.

1. Class Transaksi

Menggambarkan transaksi pembelian. Setiap transaksi terkait dengan satu item, jumlah, dan total harga.

1. Class Kasir

Memproses transaksi pembelian barang. Ketika diproses, stok item dikurangi.

1. Class Admin

Mengelola data barang, stok, dan data pegawai. Dapat menambah atau menghapus barang, menambah atau mengurangi stok, dan menambah atau menghapus pegawai.

1. Class Toko

Berisi output untuk berbelanja di toko dan mengakses menu admin.

1. Bagian Fungsi atau method.

Ada 36 method.

* 1. Class Item terdapat 4 method, diantaranya :
* getNama()

Untuk mengambil atau mendapatkan variabel dari string nama item (barang)

* getPrice()

Untuk mengambil atau mendapatkan variabel dari int price item (barang)

* getStok()

Untuk mengambil atau mendapatkan variabel dari string stok item (barang)

* setStok()

Digunakan untuk mengatur atau mengubah nilai stok suatu item atau produk dalam konteks manajemen persediaan atau penjualan

* 1. Class Pegawai terdapat 3 method, diantaranya :
* getNama()

Untuk memanggil fungsi yang berada pada string nama dalam class pegawai

* getUmur()

Untuk memanggil fungsi yang berada pada string umur dalam class pegawai

* getKota()

Untuk memanggil fungsi yang berada pada string kota dalam class pegawai

* 1. Class Transaksi terdapat 2 method, diantaranya :
* getTotalPrice()

Untuk menghitung total belanja dari user.

* cetakTransaksi()

Mencetak hasil dari total belanja user.

* 1. Class Kasir terdapat 1 method, yaitu :
* prosesTransaksi()

method ini berfungsi untuk mengatur total biaya yang harus dibayar oleh user ketika sudah selesai berbelanja.

* 1. Class Admin terdapat 9 method, diantaranya :
* tambahBarang()

Berfungsi untuk menambah barang (nama, harga, dan stok barang)

* hapusBarang()

Untuk menghapus barang yang telah dtambahkan

* tambahStok()

Untuk menambah stok barang yang telah ditambahkan oleh admin

* kurangiStok()

Untuk mengurangi stok barang yang telah ditambahkan oleh admin

* lihatStok()

Untuk melihat stok barang yang masih ada

* tambahPegawai()

Untuk menambah pegawai baru (nama, umur, dan asal kota pegawai)

* hapusPegawai()

Untuk menghapus pegawai yang telah ditambahkan oleh admin

* lihatDataPegawai()

Untuk melihat data pegawai yang telah ditambahkan

* login()

Untuk masuk ke dalam menu admin (password : 1234), jika password salah maka tidak bisa masuk ke dalam menu admin dan akan kembali ke menu awal

* 1. Class Toko terdapat 5 method, diantaranya :
* tambahBarang()

Method ini berfungsi untuk menambah barang dalam toko.

* mulaiBelanja()

Method ini berfungsi untuk menampilkan menu belanja ketika user menmilih menu belanja.

* menuAdmin()

Method ini berfungsi untuk menampilkan menu admin ketika user memilih opsi admin.

* menuDataPegawai()

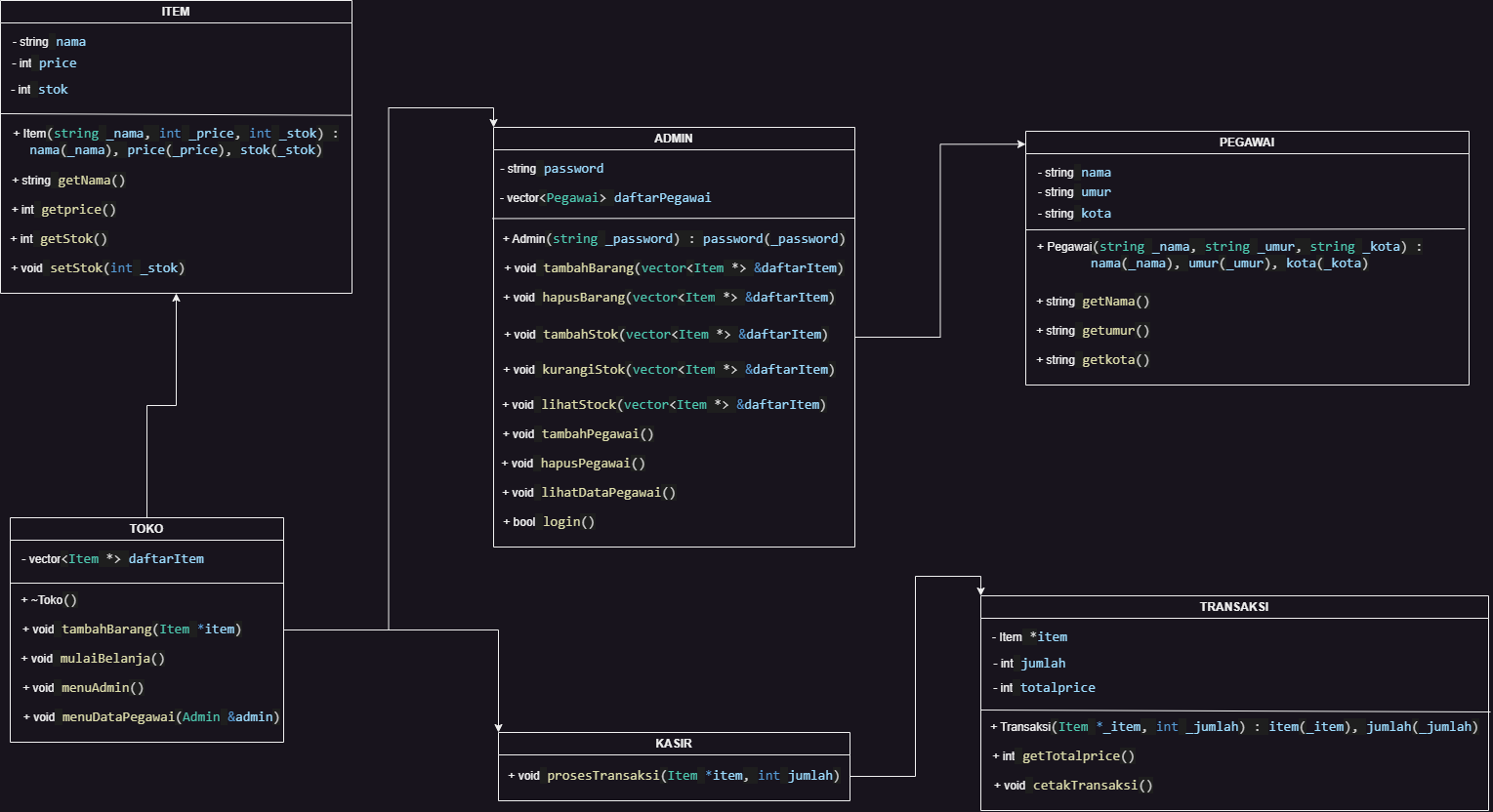
methood ini berisi menu data pegawai yang didalamnya terdapat tambah, hapus, dan lihat pegawai.

* ~Toko()

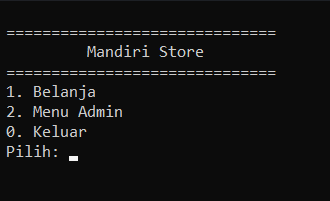
Method ini berfungsi untuk memastikan memori yang digunakan telah dibebaskan yang bertujuan untuk mengoptimalkan penggunaan memori.

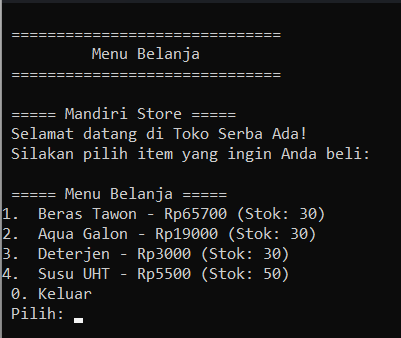
1. Bagian Main

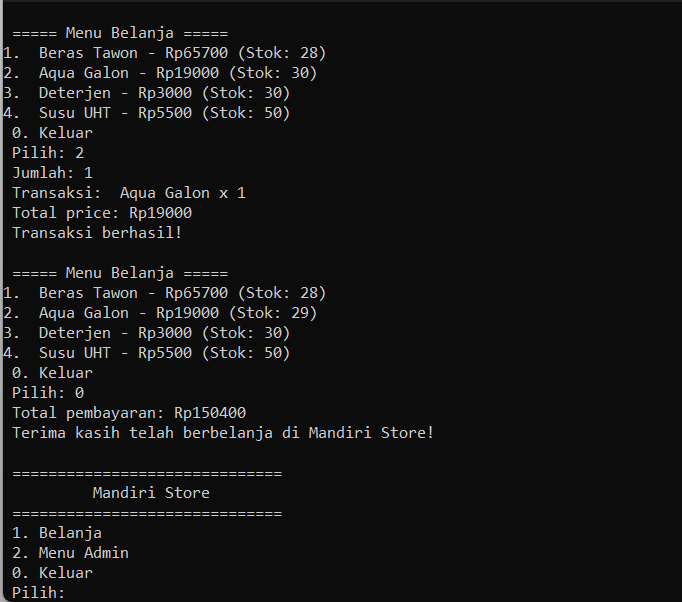
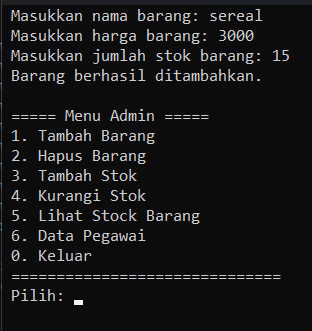
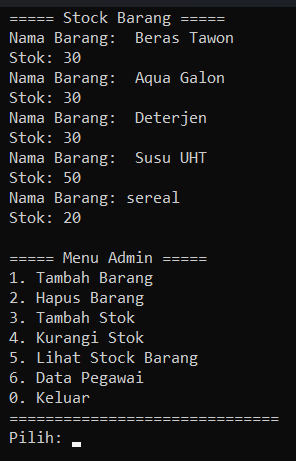
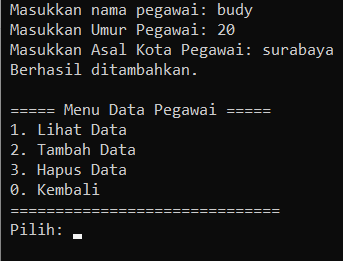
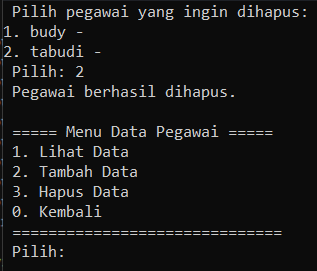
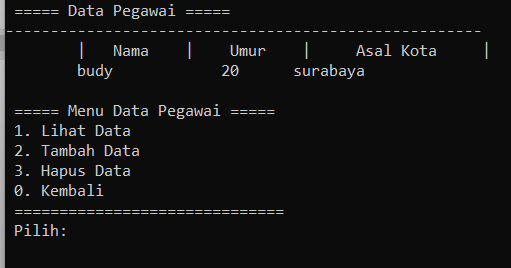
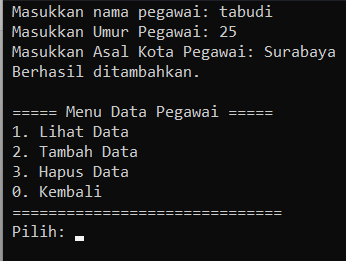
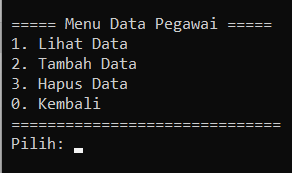
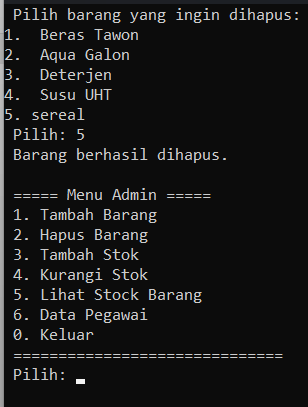
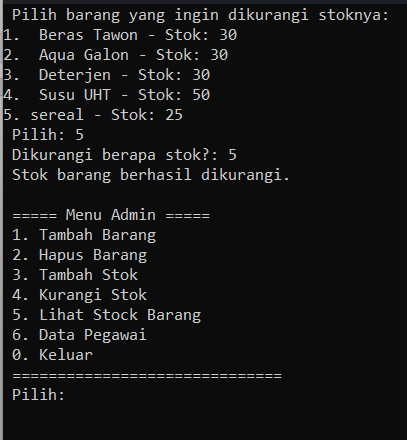
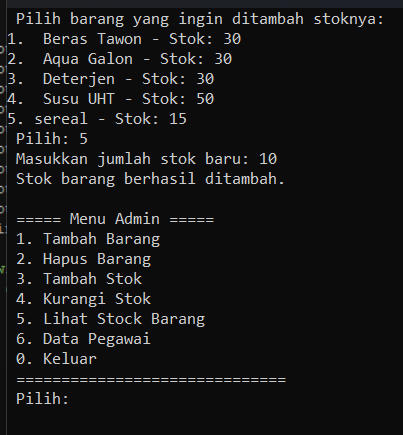
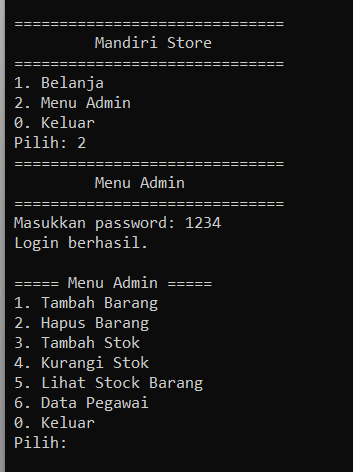
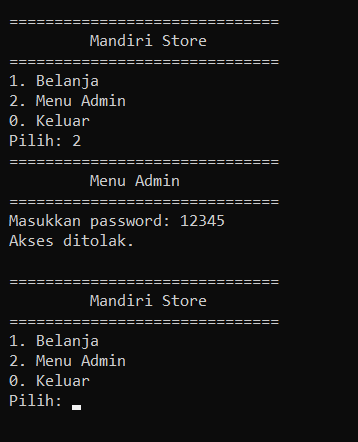
Merupakan bagian utama program. Pengguna diberi pilihan untuk memilih antara berbelanja atau mengakses menu admin.

1. **CLASS DIAGRAM**
2. **SOURCE CODE**

|  |
| --- |
| #include <iostream>  #include <windows.h>  #include <string>  #include <vector>  using namespace std;  class Item  {  private:  string nama;  int price;  int stok;  public:  Item(string \_nama, int \_price, int \_stok) : nama(\_nama), price(\_price), stok(\_stok) {}  string getNama()  {  return nama;  }  int getprice()  {  return price;  }  int getStok()  {  return stok;  }  void setStok(int \_stok)  {  stok = \_stok;  }  };  class Pegawai  {  private:  string nama;  string umur;  string kota;  public:  Pegawai(string \_nama, string \_umur, string \_kota) : nama(\_nama), umur(\_umur), kota(\_kota) {}  string getNama()  {  return nama;  }  string getumur()  {  return umur;  }  string getkota()  {  return kota;  }  };  class Transaksi  {  private:  Item \*item;  int jumlah;  int totalprice;  public:  Transaksi(Item \*\_item, int \_jumlah) : item(\_item), jumlah(\_jumlah)  {  totalprice = item->getprice() \* jumlah;  }  int getTotalprice()  {  return totalprice;  }  void cetakTransaksi()  {  cout << " Transaksi: " << item->getNama() << " x " << jumlah << endl;  cout << " Total price: Rp" << getTotalprice() << endl;  }  };  class Kasir  {  public:  void prosesTransaksi(Item \*item, int jumlah)  {  if (item->getStok() >= jumlah)  {  item->setStok(item->getStok() - jumlah);  Transaksi transaksi(item, jumlah);  transaksi.cetakTransaksi();  }  else  {  cout << " Stok tidak cukup!" << endl;  }  }  };  class Admin  {  private:  string password;  vector<Pegawai> daftarPegawai;  public:  Admin(string \_password) : password(\_password) {}  void tambahBarang(vector<Item \*> &daftarItem)  {  system("cls");  string nama;  int price, stok;  cout << " Masukkan nama barang: ";  cin >> nama;  cout << " Masukkan harga barang: ";  cin >> price;  cout << " Masukkan jumlah stok barang: ";  cin >> stok;  Item \*itemBaru = new Item(nama, price, stok);  daftarItem.push\_back(itemBaru);  cout << " Barang berhasil ditambahkan." << endl;  }  void hapusBarang(vector<Item \*> &daftarItem)  {  system("cls");  int pilihan;  cout << " Pilih barang yang ingin dihapus: " << endl;  for (int i = 0; i < daftarItem.size(); i++)  {  cout << i + 1 << ". " << daftarItem[i]->getNama() << endl;  }  cout << " Pilih: ";  cin >> pilihan;  if (pilihan >= 1 && pilihan <= daftarItem.size())  {  delete daftarItem[pilihan - 1];  daftarItem.erase(daftarItem.begin() + pilihan - 1);  cout << " Barang berhasil dihapus." << endl;  }  else  {  cout << " Pilihan tidak valid!" << endl;  }  }  void tambahStok(vector<Item \*> &daftarItem)  {  system("cls");  int pilihan;  cout << " Pilih barang yang ingin ditambah stoknya: " << endl;  for (int i = 0; i < daftarItem.size(); i++)  {  cout << i + 1 << ". " << daftarItem[i]->getNama() << " - Stok: " << daftarItem[i]->getStok() << endl;  }  cout << " Pilih: ";  cin >> pilihan;  if (pilihan >= 1 && pilihan <= daftarItem.size())  {  int stokBaru;  cout << " Masukkan jumlah stok baru: ";  cin >> stokBaru;  daftarItem[pilihan - 1]->setStok(daftarItem[pilihan - 1]->getStok() + stokBaru);  cout << " Stok barang berhasil ditambah." << endl;  }  else  {  cout << " Pilihan tidak valid!" << endl;  }  }  void kurangiStok(vector<Item \*> &daftarItem)  {  system("cls");  int pilihan;  cout << " Pilih barang yang ingin dikurangi stoknya: " << endl;  for (int i = 0; i < daftarItem.size(); i++)  {  cout << i + 1 << ". " << daftarItem[i]->getNama() << " - Stok: " << daftarItem[i]->getStok() << endl;  }  cout << " Pilih: ";  cin >> pilihan;  if (pilihan >= 1 && pilihan <= daftarItem.size())  {  int stokKurang;  cout << " Dikurangi berapa stok?: ";  cin >> stokKurang;  if (daftarItem[pilihan - 1]->getStok() >= stokKurang)  {  daftarItem[pilihan - 1]->setStok(daftarItem[pilihan - 1]->getStok() - stokKurang);  cout << " Stok barang berhasil dikurangi." << endl;  }  else  {  cout << " Stok tidak cukup!" << endl;  }  }  else  {  cout << " Pilihan tidak valid!" << endl;  }  }  void lihatStock(vector<Item \*> &daftarItem)  {  system("cls");  cout << " ===== Stock Barang =====" << endl;  for (int i = 0; i < daftarItem.size(); i++)  {  cout << " Nama Barang: " << daftarItem[i]->getNama() << endl;  cout << " Stok: " << daftarItem[i]->getStok() << endl;  }  }  void tambahPegawai()  {  system("cls");  string nama, umur, kota;  cout << " Masukkan nama pegawai: ";  cin >> nama;  cout << " Masukkan Umur Pegawai: ";  cin >> umur;  cout << " Masukkan Asal Kota Pegawai: ";  cin >> kota;  Pegawai pegawai(nama, umur, kota);  daftarPegawai.push\_back(pegawai);  cout << " Berhasil ditambahkan." << endl;  }  void hapusPegawai()  {  system("cls");  int pilihan;  cout << " Pilih pegawai yang ingin dihapus: " << endl;  for (int i = 0; i < daftarPegawai.size(); i++)  {  cout << i + 1 << ". " << daftarPegawai[i].getNama() << " - " << endl;  }  cout << " Pilih: ";  cin >> pilihan;  if (pilihan >= 1 && pilihan <= daftarPegawai.size())  {  daftarPegawai.erase(daftarPegawai.begin() + pilihan - 1);  cout << " Pegawai berhasil dihapus." << endl;  }  else  {  cout << " Pilihan tidak valid!" << endl;  }  }  void lihatDataPegawai()  {  system("cls");  cout << " ===== Data Pegawai =====" << endl;  cout << "-----------------------------------------------------" << endl;  cout << "\t| Nama | Umur | Asal Kota |" << endl;  for (int i = 0; i < daftarPegawai.size(); i++)  {  cout << "\t" << daftarPegawai[i].getNama() << "\t\t" << daftarPegawai[i].getumur() << "\t" << daftarPegawai[i].getkota() << endl;  }  }  bool login()  {  string inputPassword;  cout << " Masukkan password: ";  cin >> inputPassword;  if (inputPassword == password)  {  cout << " Login berhasil." << endl;  return true;  }  else  {  cout << " Akses ditolak." << endl;  return false;  }  }  };  class Toko  {  private:  vector<Item \*> daftarItem;  public:  ~Toko()  {  for (Item \*item : daftarItem){  delete item;  }  }  void tambahBarang(Item \*item)  {  daftarItem.push\_back(item);  }  void mulaiBelanja()  {  Kasir kasir;  int pilihan;  int jumlah;  int totalPembayaran = 0;  cout << " " << endl;  cout << " ===== Mandiri Store =====" << endl;  cout << " Selamat datang di Toko Serba Ada!" << endl;  do  {  cout << " " << endl;  cout << " ===== Menu Belanja =====" << endl;  cout << "No | Nama Barang | Harga Barang |" << endl;  cout << "-----------------------------------------------------" << endl;  for (int i = 0; i < daftarItem.size(); i++){  cout << i + 1 << "\t" << daftarItem[i]->getNama() << "\t\tRp" << daftarItem[i]->getprice() << endl;  }  cout << "-----------------------------------------------------" << endl;  cout << " 0. Keluar" << endl;  cout << " Silakan pilih item yang ingin Anda beli:" << endl;  cout << " Pilih: ";  cin >> pilihan;  if (pilihan >= 1 && pilihan <= daftarItem.size()){  cout << " Jumlah: ";  cin >> jumlah;  if (jumlah > 0){  Item \*selectedItem = daftarItem[pilihan - 1];  kasir.prosesTransaksi(selectedItem, jumlah);  totalPembayaran += selectedItem->getprice() \* jumlah;  cout << " Transaksi berhasil!" << endl;  }  else{  cout << " Jumlah tidak valid!" << endl;  }  }  else if (pilihan != 0){  cout << " Pilihan tidak valid!" << endl;  }  }  while (pilihan != 0);  cout << " Total pembayaran: Rp" << totalPembayaran << endl;  cout << " Terima kasih telah berbelanja di Mandiri Store!" << endl;  }  void menuAdmin()  {  Admin admin("1234");  if (!admin.login()){  return;  }  int pilihan;  do  {  cout << " " << endl;  cout << " ===== Menu Admin =====" << endl;  cout << " 1. Tambah Barang" << endl;  cout << " 2. Hapus Barang" << endl;  cout << " 3. Tambah Stok" << endl;  cout << " 4. Kurangi Stok" << endl;  cout << " 5. Lihat Stock Barang" << endl;  cout << " 6. Data Pegawai" << endl;  cout << " 0. Keluar" << endl;  cout << " ==============================" << endl;  cout << " Pilih: ";  cin >> pilihan;  switch (pilihan){  case 1:  system("cls");  admin.tambahBarang(daftarItem);  break;  case 2:  system("cls");  admin.hapusBarang(daftarItem);  break;  case 3:  system("cls");  admin.tambahStok(daftarItem);  break;  case 4:  system("cls");  admin.kurangiStok(daftarItem);  break;  case 5:  system("cls");  admin.lihatStock(daftarItem);  break;  case 6:  system("cls");  menuDataPegawai(admin);  break;  case 0:  break;  default:  cout << " Pilihan tidak valid!" << endl;  break;  }  } while (pilihan != 0);  }  void menuDataPegawai(Admin &admin)  {  int pilihan;  do  {  cout << " " << endl;  cout << " ===== Menu Data Pegawai =====" << endl;  cout << " 1. Lihat Data" << endl;  cout << " 2. Tambah Data" << endl;  cout << " 3. Hapus Data" << endl;  cout << " 0. Kembali" << endl;  cout << " ==============================" << endl;  cout << " Pilih: ";  cin >> pilihan;  switch (pilihan){  case 1:  admin.lihatDataPegawai();  break;  case 2:  admin.tambahPegawai();  break;  case 3:  admin.hapusPegawai();  break;  case 0:  break;  default:  cout << " Pilihan tidak valid!" << endl;  break;  }  } while (pilihan != 0);  }  };  int main()  {  Toko toko;  Item item1(" Beras Tawon", 65700, 30);  Item item2(" Aqua Galon", 19000, 30);  Item item3(" Deterjen", 3000, 30);  Item item4(" Susu UHT", 5500, 50);  toko.tambahBarang(&item1);  toko.tambahBarang(&item2);  toko.tambahBarang(&item3);  toko.tambahBarang(&item4);  int menu;  do  {  cout << " " << endl;  cout << " ==============================" << endl;  cout << " Mandiri Store " << endl;  cout << " ==============================" << endl;  cout << " 1. Belanja" << endl;  cout << " 2. Menu Admin" << endl;  cout << " 0. Keluar" << endl;  cout << " ==============================" << endl;  cout << " Pilih: ";  cin >> menu;  switch (menu){  case 1:  system("cls");  cout << " " << endl;  cout << " ==============================" << endl;  cout << " Menu Belanja " << endl;  cout << " ==============================" << endl;  toko.mulaiBelanja();  break;  case 2:  cout << " ==============================" << endl;  cout << " Menu Admin " << endl;  cout << " ==============================" << endl;  toko.menuAdmin();  break;  case 0:  cout << " Terima kasih telah mengunjungi Mandiri Store!" << endl;  break;  default:  cout << " Pilihan tidak valid!" << endl;  break;  }  } while (menu != 0);  return 0;  } |

1. **OUTPUT**





1. **Link GitHub**

<https://github.com/iyansaputra/PBO>