**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. **PENJELASN SINGKAT 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.

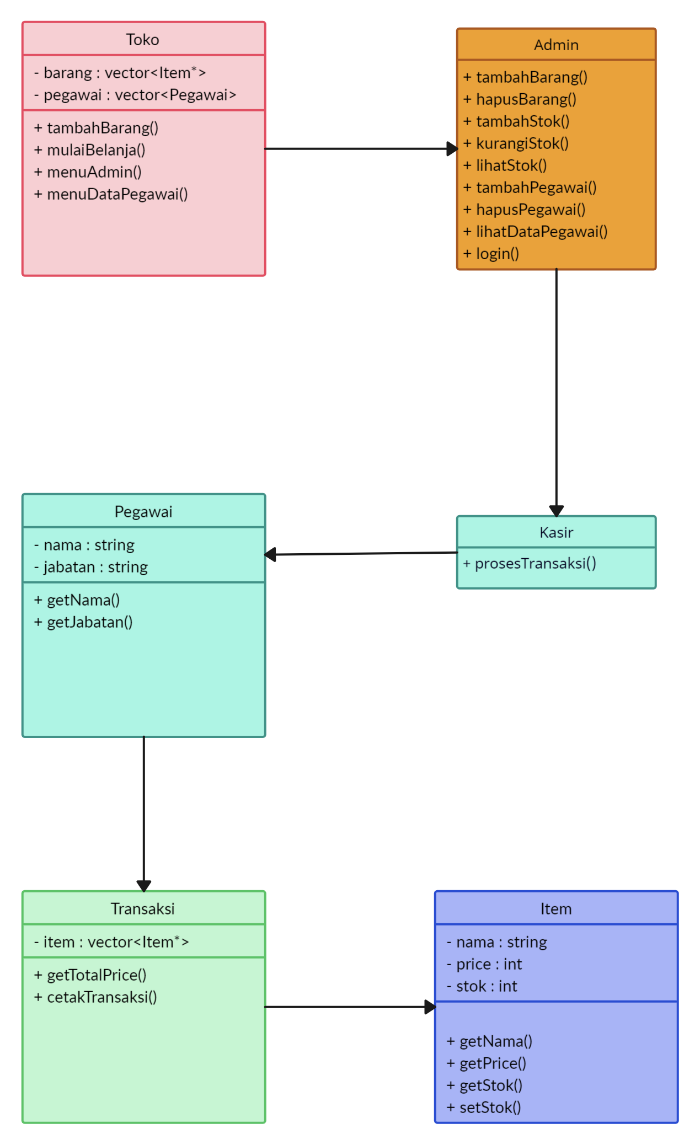
1. Bagian Fungsi atau method.

Ada 23 method.

1. Class item: 4 method, getNama(), getprice(), getStok(), setStok().
2. Class Pegawai: 2 method, getNama(), getJabatan().
3. Class Transaksi: 2 method, getTotalprice(), cetakTransaksi().
4. Class Kasir: 1 method, prosesTransaksi().
5. Class Admin: 9 method, tambahBarang(), hapusBarang(), tambahStok(), kurangiStok(), lihatStok(), tambahPegawai(), hapusPegawai(), lihatDataPegawai(), login().
6. Class Toko: 5 method, tambahBarang(), mulaiBelanja(), menuAdmin(), menuDataPegawai(), ~Toko().
7. Bagian Main

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

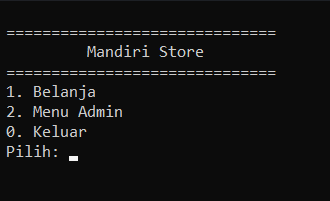
1. **CLASS DIAGRAM**

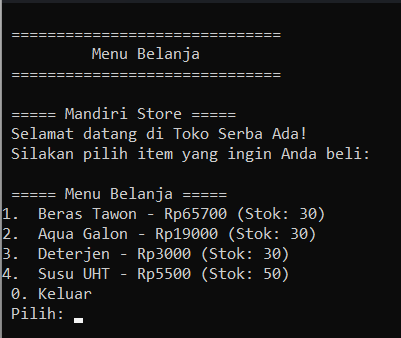
****

1. **SOURCE CODE**

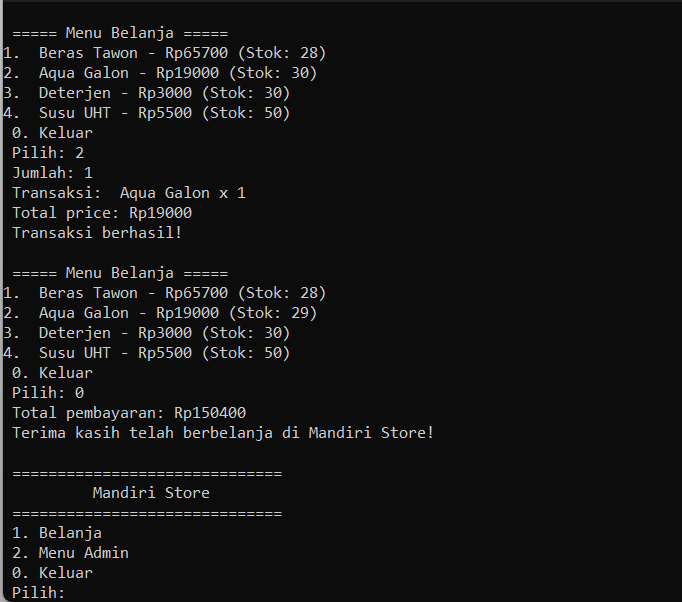
|  |
| --- |
| #include <iostream>  #include <windows.h>  #include <string>  #include <vector>  using namespace std;  //Bagian Deklarasi Class  class Item //pendeklarasian class item  {  private:  string nama;  int price;  int stok;  public:  //ada 1 konstruktor menerima 3 parameter, \_nama, \_stok, \_price.  Item(string \_nama, int \_price, int \_stok) : nama(\_nama), price(\_price), stok(\_stok) {}  string getNama() //method (agar bisa mengakses nama)  {  return nama;  }  int getprice()  {  return price;  }  int getStok() //mengembalikan nilai stok  {  return stok;  }  void setStok(int \_stok) //mengubah nilai stok  {  stok = \_stok;  }  };  //deklarasi class pegawai  class Pegawai  {  private:  string nama;  string jabatan;  public:  //1 konstruktor, 2 parameter  Pegawai(string \_nama, string \_jabatan) : nama(\_nama), jabatan(\_jabatan) {}  string getNama() //mengembalikan nilai variabel Nama  {  return nama;  }  string getJabatan() //mengembalikan nilai variabel Jabatan  {  return jabatan;  }  };  class Transaksi  {  private:  Item \*item; //pointer  int jumlah;  int totalprice; // Menambahkan atribut totalprice untuk menyimpan total pembayaran  public:  Transaksi(Item \*\_item, int \_jumlah) : item(\_item), jumlah(\_jumlah)  {  totalprice = item->getprice() \* jumlah; // Menghitung totalprice saat membuat objek Transaksi  }  int getTotalprice() //method  {  return totalprice;  }  void cetakTransaksi() //method  {  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; // atribut password  vector<Pegawai> daftarPegawai; // add vector pegawai  public:  Admin(string \_password) : password(\_password) {} // add constructor dengan parameter password  void tambahBarang(vector<Item \*> &daftarItem) //bagian implementasi fungi / method  {  system ("cls");  string nama;  int price, stok;  cout << " Masukkan nama barang: ";  cin >> nama;  cout << " Masukkan price 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) //method  {  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) //method  {  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) //method  {  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) //method  {  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() //method  {  system ("cls");  string nama, jabatan;  cout << " Masukkan nama pegawai: ";  cin >> nama;  cout << " Masukkkan jabatan pegawai: ";  cin >> jabatan;  Pegawai pegawai(nama, jabatan);  daftarPegawai.push\_back(pegawai);  cout << " Berhasil ditambahkan." << endl;  }  void hapusPegawai() //method  {  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() << " - " << daftarPegawai[i].getJabatan() << 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() //method  {  system ("cls");  cout << " ===== Data Pegawai =====" << endl;  for (int i = 0; i < daftarPegawai.size(); i++)  {  cout << " Nama: " << daftarPegawai[i].getNama() << endl;  cout << " Jabatan: " << daftarPegawai[i].getJabatan() << endl;  }  }  bool login() //method  {  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) //method  {  daftarItem.push\_back(item);  }  void mulaiBelanja() //method  {  Kasir kasir;  int pilihan;  int jumlah;  int totalPembayaran = 0;  cout << " " << endl;  cout << " ===== Toko Serba Ada =====" << endl;  cout << " Selamat datang di Toko Serba Ada!" << endl;  cout << " Silakan pilih item yang ingin Anda beli:" << endl;  do  {  cout << " " << endl;  cout << " ===== Menu Belanja =====" << endl;  for (int i = 0; i < daftarItem.size(); i++)  {  cout << i + 1 << ". " << daftarItem[i]->getNama() << " - Rp" << daftarItem[i]->getprice() << " (Stok: " << daftarItem[i]->getStok() << ")" << endl;  }  cout << " 0. Keluar" << 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 Toko Serba Ada!" << endl;  }  void menuAdmin() //method  {  Admin admin("1234"); // passwordnya  if (!admin.login())  {  return; // jika salah, akses ditolak  }  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 << " 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) //method  {  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 << " 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);  }  };  //Bagian main (fungsi utama)  int main()  {  Toko toko;  // Inisialisasi Item  Item item1(" Beras Tawon", 65700, 30);  Item item2(" Aqua Galon", 19000, 30);  Item item3(" Deterjen", 3000, 30);  Item item4(" Susu UHT", 5500, 50);  // Tambahkan Item ke Toko  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 << " 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;  } |

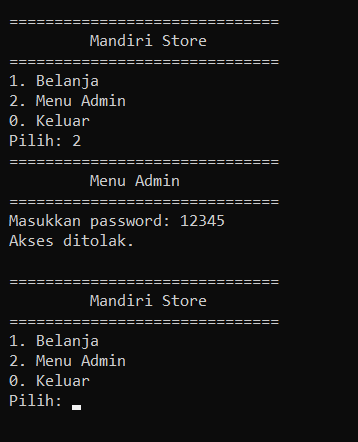
1. **OUTPUT**



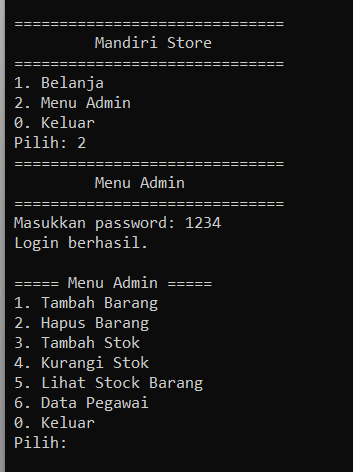
****Gambar di atas adalah tampilan awal dari program kami. Terdapat 3 menu, yaitu menu belanja, menu admin, dan juga menu keluar.

Ketika user memilih menu 1, maka outputnya seperti pada gambar di atas. Program akan menampilkan barang yang dijual, harga barang, dan juga stok barangnya.

****

****User harus memasukkan barang yang ingin di beli dan jumlahnya. User bisa menginput 0 untuk keluar dari menu belanja, dan program akan menampilkan total harga yang harus dibayar oleh user.

Gambar di atas adalah tampilan ketika user memilih menu admin. User akan diminta memasukkan password, jika password salah maka akan muncul tulisan “Akses ditolak” dan user tidak bisa masuk ke menu admin dan akan kembali pada tampilan awal.



Ketika password yang diinput user benar, maka user bisa memasuki menu admin. Dalam menu admin, user dapat menambah dan menghapus barang, menambah, mengurangi, dan melihat stok barang dan mengakses data pegawai.

