**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**

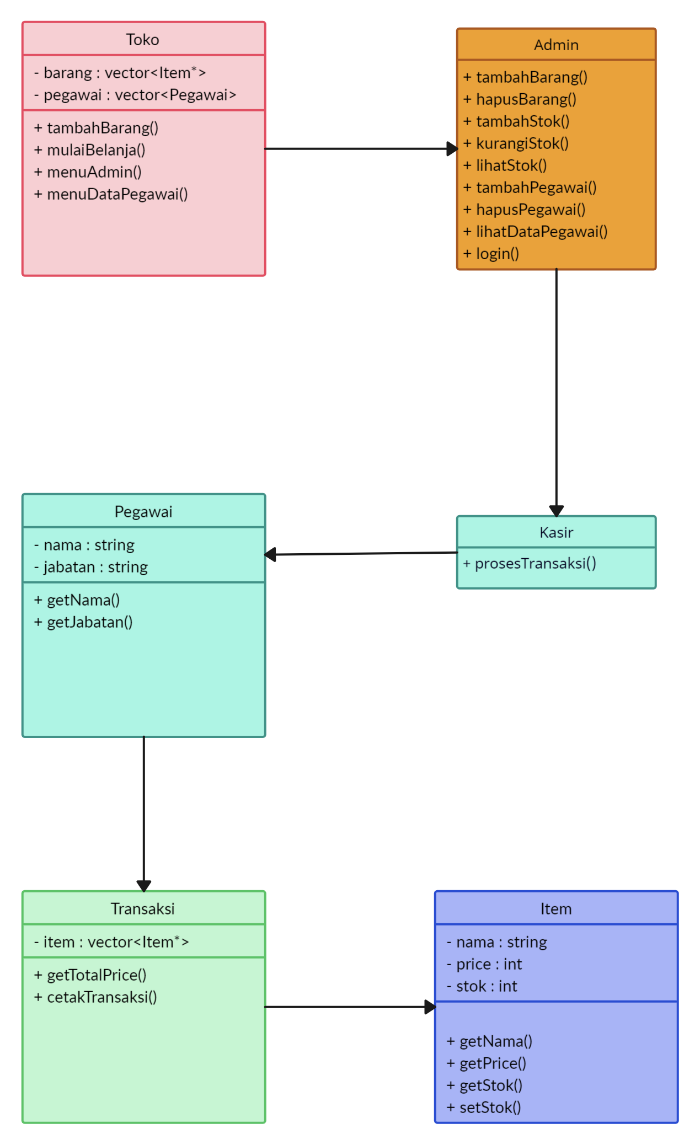
**TAHUN 2023**

1. **PENJELASN SINGKAT PROGRAM**

Program yang kami kerjakan ini mengambil tema tentang kasir. Di dalam program kasir ini, terdapat 2 menu untuk user dan admin, yaitu menu belanja dan menu admin. Di dalam menu belanja ini, user dapat membeli barang yang dibutuhkan, kemudian program akan menampilkan total biaya yang harus dibayar oleh user.

Sebelum masuk ke dalam menu admin, user harus memasukkan password terlebih dahulu, sehingga tidak sembarang orang dapat mengakses menu admin. Ketika user salah menginput password, maka program akan menampilkan “Akses ditolak” dan program akan mengembalikan ke program awal. Jika password yang diinput user benar, maka program akan menampilakn menu admin. Menu admin ini fungsinya adalah untuk mengatur semua hal yang berkaitan tentang toko, mulai dari barang, harga barang, stok barang dan juga pegawai. Admin dapat mengubah semuanya sesuka hatinya.

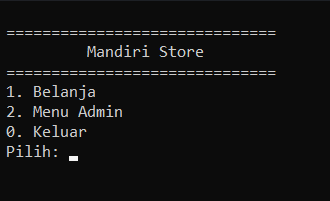
1. **CLASS DIAGRAM**

****

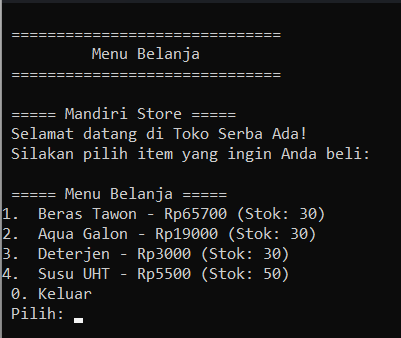
1. **SOURCE CODE**

|  |
| --- |
| #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 umur;  string kota;  public:  // 1 konstruktor, 2 parameter  Pegawai(string \_nama, string \_umur, string \_kota) : nama(\_nama), umur(\_umur), kota(\_kota) {}  string getNama() // mengembalikan nilai variabel Nama  {  return nama;  }  string getumur() // mengembalikan nilai variabel Umur  {  return umur;  }  string getkota() // mengembalikan nilai variabel Asal Kota  {  return kota;  }  };  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 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) // 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, 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() // 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() << " - " << 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;  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() // 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 << " ===== 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() // 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 << " ==============================" << 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 << " ==============================" << 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 << " ==============================" << 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;  } |

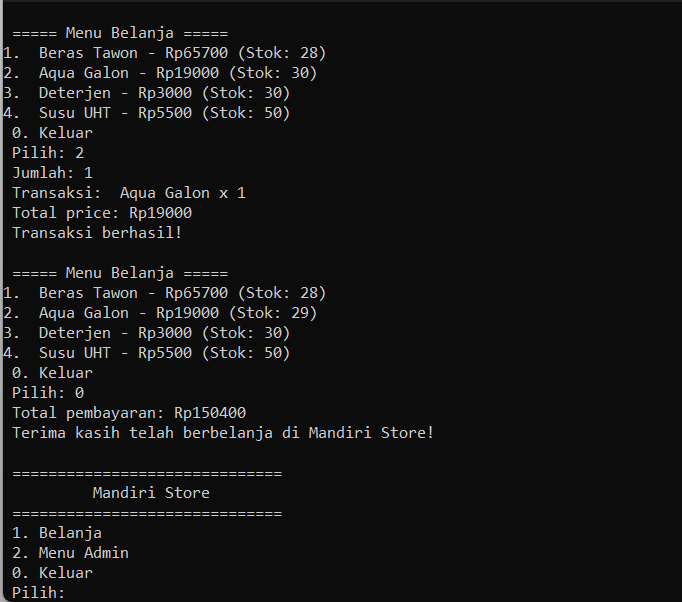
1. **OUTPUT**

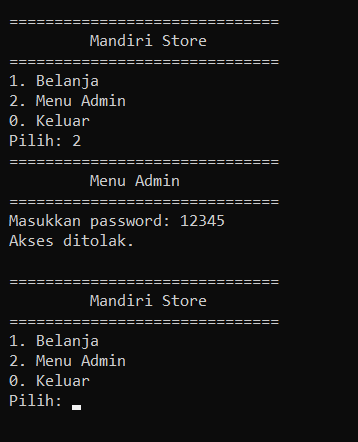


Gambar di atas adalah tampilan awal dari program kami. Terdapat 3 menu, yaitu menu belanja, menu admin, dan juga menu keluar.

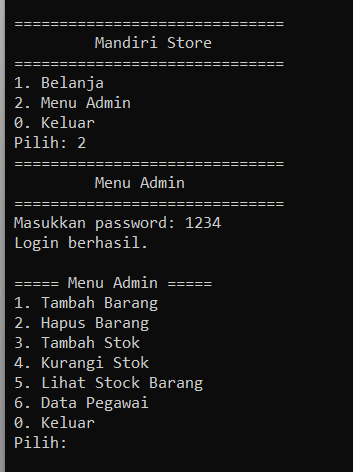
****

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

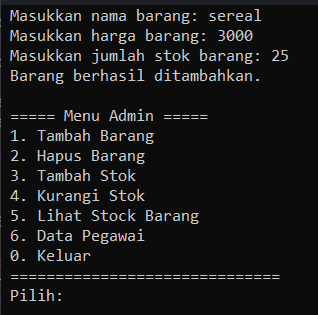
****

****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.

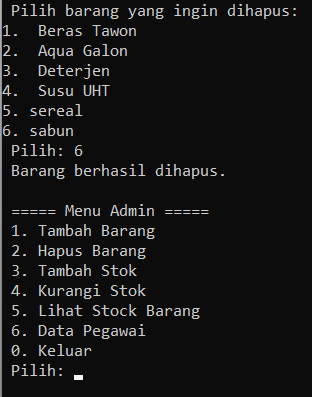
Gambar di atas adalah tampilan ketika user memilih menu admin. User akan diminta memasukkan password, jika password salah maka akan muncul tulisan “Akses ditolak” dan user tidak bisa masuk ke menu admin dan akan kembali pada tampilan awal.



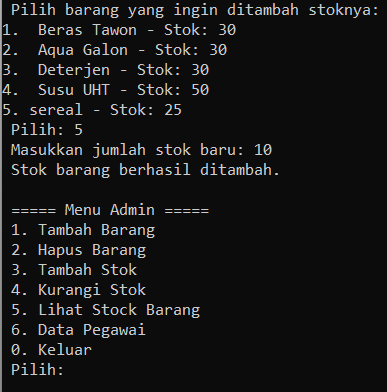
Ketika password yang diinput user benar, maka user bisa memasuki menu admin. Dalam menu admin, user dapat menambah dan menghapus barang, menambah, mengurangi, dan melihat stok barang dan mengakses data pegawai.



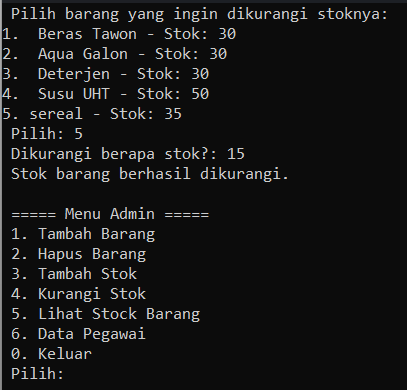
Ketika user memilih menu tambah barang, maka user akan disuruh menginput nama barang, harga barang, dan jumlah stok. Misalnya user menginput sereal dengan harga 3000 dan jumlah stoknya 25.



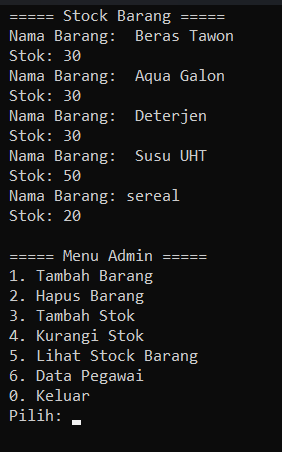
Dalam menu admin ini, user juga dapat menghapus barang. Ketika memilih menu hapus barang, program akan menampilkan daftar barang, kemudian user dapat memilih barang yang ingin dihapus.



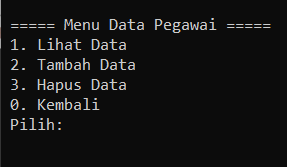
User juga dapat menambah stok, program akan menampilkan daftar barang dan jumlah stoknya, kemudian user dapat memilih barang yang ingin ditambah stoknya. Setelah itu user dapat menambahkan stok barang baru.



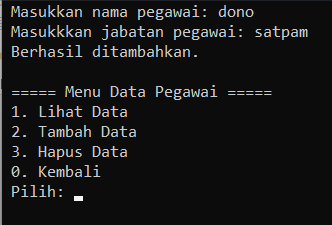
Selain menambah stok, user juga dapat mengurangi stok barang. Tampilannya hampir sama seperti tambah stok, di sini user dapat memilih barang yang ingin dikurangi stoknya, kemudian user dapat memasukkan ingin dikurangi berapa stok.



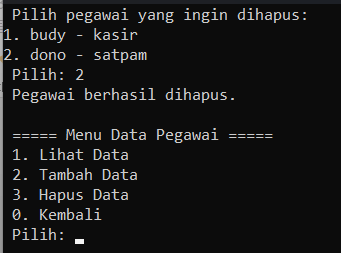
Setelah itu, user dapat melihat stok barang pada menu “lihat stok barang”. Program akan menampilkan semua daftar barang, harga dan jumlah stok yang tersedia.



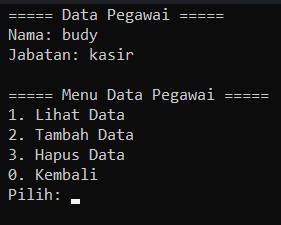
Di dalam menu data pegawai ini, user dapat menambah, menghapus, dan melihat data pegawai.



Dalam menu tambah data, user dapat menambahkan pegawai. Dimulai dari memasukkan nama pegawai baru dan jabatan pegawai.



User juga dapat menghapus data pegawai melalui menu “hapus data”. Maka program akan menampilkan daftar pegawai. Kemudian user akan memasukkan pegawai mana yang ingin dihapus.



Setelah menambah dan menghapus data pegawai, user dapat melihat seluruh data pegawai melalui menu “Lihat data”. Di sini program akan menampilkan seluruh data pegawai dan jabatan.