

**LAPORAN PRAKTIKUM**  
**ALGORITMA DAN PEMROGRAMAN LANJUT**  
**POSTTEST 6**



**Informatika C'23**  
**Nama Diki Wahyudi**  
**2309106097**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2024**

# PEMBAHASAN

## 1.1 LATAR BELAKANG

Ini merupakan sebuah program tentang penjualan beras yang Dimana menggunakan CRUD pada menunya

- #include untuk memasukkan library iostream dan string  
#std untuk mempermudah penggunaan elemen-elemen standar

```
1  #include <iostream>
2  #include <string>
```

- Berisi informasi tentang jenis beras, harga per kilogram, dan stok.

```
struct BerasInfo {
    string jenisNama;
    double hargaBeras;
    int stokBeras;
};
```

- Fungsi ini digunakan untuk melakukan proses login dengan membandingkan username dan password yang dimasukkan pengguna dengan nilai yang sudah ditentukan. Jika login berhasil, fungsi mengembalikan nilai true, jika gagal (setelah 3 percobaan), mengembalikan nilai false.

```
bool prosesLogin() {
    string user, pw;
    int percobaan = 0;
    const string userBenar = "diki";
    const string pwBenar = "097";
```

- Fungsi ini menampilkan menu utama pada program toko beras.

```
void tampilMenu() {
    cout << "=== Toko Beras Diki Walla ===" << endl;
    cout << "1. Lihat Daftar Beras" << endl;
    cout << "2. Tambah Jenis Beras" << endl;
    cout << "3. Jual Beras" << endl;
    cout << "4. Hapus Jenis Beras" << endl;
    cout << "5. Ubah Jenis Beras" << endl;
    cout << "6. Keluar" << endl;
    cout << "Pilihan: ";
}
```

- Fungsi ini digunakan pengguna untuk menambahkan data jenis beras baru ke dalam daftar beras.

```
void tambahBerasData(BerasInfo daftarBeras[], int& jumlahBeras) {
    if (jumlahBeras < 10) {
        BerasInfo berasInput;
        cout << "Jenis Beras: ";
        cin.ignore();
        getline(cin, berasInput.jenisNama);
        cout << "Harga per Kg: ";
        cin >> berasInput.hargaBeras;
        cout << "Stok (Kg): ";
        cin >> berasInput.stokBeras;
        daftarBeras[jumlahBeras] = berasInput;
        jumlahBeras++;
        cout << "Jenis beras berhasil ditambahkan!" << endl;
    } else {
        cout << "Toko sudah penuh!" << endl;
    }
}
```

- Fungsi ini memungkinkan pengguna untuk menjual sejumlah beras dari jenis tertentu, mengurangi stok beras dan menampilkan total harga transaksi.

```
void jualBerasData(BerasInfo daftarBeras[], int jumlahBeras) {
    string jenisNama;
    int jumlahJual;
    cout << "Jenis Beras: ";
    cin.ignore();
    getline(cin, jenisNama);
    cout << "Jumlah Jual (Kg): ";
    cin >> jumlahJual;

    for (int i = 0; i < jumlahBeras; i++) {
        if (daftarBeras[i].jenisNama == jenisNama) {
            if (daftarBeras[i].stokBeras >= jumlahJual) {
                daftarBeras[i].stokBeras -= jumlahJual;
                double totalHarga = daftarBeras[i].hargaBeras * jumlahJual;
                cout << "Total Harga: " << totalHarga << endl;
                cout << "Transaksi berhasil!" << endl;
            } else {
                cout << "Stok tidak mencukupi!" << endl;
            }
        }
    }
    return;

    cout << "Jenis beras tidak ditemukan!" << endl;
}
```

- Fungsi ini memungkinkan pengguna untuk menghapus jenis beras dari daftar.

```
void hapusBerasData(BerasInfo daftarBeras[], int& jumlahBeras) {
    string jenisNama;
    cout << "Jenis Beras yang Akan Dihapus: ";
    cin.ignore();
    getline(cin, jenisNama);

    for (int i = 0; i < jumlahBeras; i++) {
        if (daftarBeras[i].jenisNama == jenisNama) {
            // Menggeser data setelah jenis beras yang dihapus ke kiri
            for (int j = i; j < jumlahBeras - 1; j++) {
                daftarBeras[j] = daftarBeras[j + 1];
            }
            jumlahBeras--;
            cout << "Jenis beras berhasil dihapus!" << endl;
            return;
        }
    }

    cout << "Jenis beras tidak ditemukan!" << endl;
}
```

- Fungsi ini memungkinkan pengguna untuk mengubah informasi tentang suatu jenis beras.

```
void ubahBerasData(BerasInfo daftarBeras[], int jumlahBeras) {
    string jenisNama;
    cout << "Jenis Beras yang Akan Diubah: ";
    cin.ignore();
    getline(cin, jenisNama);

    for (int i = 0; i < jumlahBeras; i++) {
        if (daftarBeras[i].jenisNama == jenisNama) {
            BerasInfo berasUbah;
            cout << "Jenis Beras Baru: ";
            getline(cin, berasUbah.jenisNama);
            cout << "Harga per Kg: ";
            cin >> berasUbah.hargaBeras;
            cout << "Stok (Kg): ";
            cin >> berasUbah.stokBeras;
            daftarBeras[i] = berasUbah;
            cout << "Jenis beras berhasil diubah!" << endl;
            return;
        }
    }

    cout << "Jenis beras tidak ditemukan!" << endl;
}
```

- Fungsi utama berisi inisialisasi array `daftarBeras` yang menyimpan informasi beras, variabel `jumlahBeras` yang menyimpan jumlah jenis beras, dan variabel `pilihan` untuk menyimpan pilihan menu yang dimasukkan pengguna. Program diawali dengan proses login, dan setelah login berhasil, user dapat memilih berbagai opsi menu yang disediakan.

```
int main() {
```

- Fungsi menampilkan menu utama secara rekursif

```
void tampilMenuUtama(BerasInfo daftarBeras[], int jumlahBeras) {
    tampilMenu();
    int pilihan;
    cin >> pilihan;
    cin.ignore();
}
```

- prinsip pointer pada program.

```
// Fungsi mencari jenis beras dan mengembalikan pointer ke objek beras
BerasInfo* cariBerasData(const string& jenisNama) {
    for (auto& beras : daftarBeras) {
        if (beras.jenisNama == jenisNama) {
            return &beras;
        }
    }
    return nullptr;
}
```

- Terdapat fungsi dengan parameter address-of.

```
// Fungsi dengan parameter address-of
void tampilkanInfoBeras(const BerasInfo& beras) {
    cout << "Jenis Beras: " << beras.jenisNama << endl;
    cout << "Harga per Kg: " << beras.hargaBeras << endl;
    cout << "Stok: " << beras.stokBeras << " Kg" << endl;
}
```

- Terdapat fungsi dengan parameter dereference.

```
// Fungsi dengan parameter dereference
void tambahStokBeras(BerasInfo* beras, int tambahanStok) {
    if (beras != nullptr) {
        beras->stokBeras += tambahanStok;
        cout << "Stok beras berhasil ditambahkan!" << endl;
    } else {
        cout << "Jenis beras tidak ditemukan!" << endl;
    }
}
```

- Fungsi sorting ascending

```
// Mengurutkan daftar beras berdasarkan jenis nama (ascending)
sort(daftarBeras.begin(), daftarBeras.end(), compareBerasNama);

for (const auto& beras : daftarBeras) {
    cout << "Jenis Beras: " << beras.jenisNama << endl;
    cout << "Harga per Kg: " << beras.hargaBeras << endl;
    cout << "Stok: " << beras.stokBeras << " Kg" << endl;
    cout << "-----" << endl;
}
tampilMenuUtama();
```

## 1.2 OUTPUT PROGRAM

```
1.C++ Posttest_2.C++ 2309106097_Diki Wahyudi_Posttest 2.C++ posttest3.c++ posttest 4.c++ posttes.c++
posttest7.c++
217 void tampilMenuUtama() {
223     try {
228         switch (pilihan) {
263             case 7: // Kasus untuk sorting ascending
272                 cout << "Jenis Beras: " << beras.jenisNama << endl;
273                 cout << "Harga per Kg: " << beras.hargaBeras << endl;
274                 cout << "Stok: " << beras.stokBeras << " Kg" << endl;
275                 cout << "-----" << endl;
276             }
277         }
278         tampilMenuUtama();
279         break;
280     case 8:
281         cout << "Terima kasih telah belanja di toko kami!" << endl;
282         break;
283     }
284 } catch (const invalid_argument& e) {
285     cout << e.what() << endl;
}

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\File C++> cd "c:\File C++\" ; if ($?) { g++ posttest7.c++ -o posttest7 } ; if ($?) { .\posttest7 }
Username: diki
Password: 097
=== Toko Beras Diki Walla ===
1. Lihat Daftar Beras
2. Tambah Jenis Beras
3. Jual Beras
4. Hapus Jenis Beras
5. Ubah Jenis Beras
6. Cari Jenis Beras
7. Urutkan Daftar Beras (Ascending)
8. Keluar
Pilihan: 
```