

RESPONSI II
PRAKTIKUM DASAR PEMROGRAMAN
2023/2024

IDENTITAS

Nama : Muhammad Riza Zaidan
NIM : L0223031
Kelas : B Sains Data
Judul Program : My BUYATI App

DOKUMENTASI PROGRAM

SOURCE CODE

```
#include <iostream>
#include <vector>

using namespace std;

// Kelas untuk merepresentasikan barang
class Barang {
public:
    string nama;
    double hargaBeli;
    double hargaJual;
    int stock;

    // Konstruktor untuk inisialisasi barang
    Barang(string n, double hb, double hj, int s)
        : nama(n), hargaBeli(hb), hargaJual(hj), stock(s) {}

    // Menampilkan informasi barang
    void displayBarang() {
        cout << "_____ " << endl;
        cout << "Nama: " << nama << endl;
        cout << "Harga Beli: " << hargaBeli << endl;
        cout << "Harga Jual: " << hargaJual << endl;
        cout << "Stock: " << stock << endl;
    }

    // Mengupdate informasi barang
    void updateBarang(string n, double hb, double hj, int s) {
        nama = n;
```

```

        hargaBeli = hb;
        hargaJual = hj;
        stock = s;
    }

    // Menambah stok barang
    void restock(int jumlah) {
        stock += jumlah;
    }

    // Menangani penjualan barang
    void sell(int jumlah) {
        if (jumlah > stock) {
            cout << "ERROR [Jumlah barang yang dijual melebihi stock]" << endl;
        } else {
            stock -= jumlah;
            cout << "SUCCESS [Penjualan berhasil]" << endl;
        }
    }
};

// Kelas untuk merepresentasikan warung
class Warung {
private:
    vector<Barang> daftarBarang;

public:
    // Menambah barang baru ke dalam daftar barang
    void tambahBarang() {
        string nama;
        double hargaBeli, hargaJual;
        int stock;

        cout << "_____ " << endl;
        cout << "Masukkan nama barang: ";
        cin >> nama;
        cout << "Masukkan harga beli: ";
        cin >> hargaBeli;
        cout << "Masukkan harga jual: ";
        cin >> hargaJual;
        cout << "Masukkan stock: ";
        cin >> stock;

        Barang barangBaru(nama, hargaBeli, hargaJual, stock);
        daftarBarang.push_back(barangBaru);
        cout << "SUCCESS [Barang berhasil ditambahkan]" << endl;
    }
};

```

```

// Menampilkan semua barang dalam daftar
void tampilkanSemuaBarang() {
    if (daftarBarang.empty()) {
        cout << "ERROR [Tidak ada barang yang tersedia]" << endl;
        return;
    }
    cout << "_____ " << endl;
    cout << "Daftar Barang:" << endl;
    for (Barang barang : daftarBarang) {
        barang.displayBarang();
    }
}

// Menampilkan informasi satu barang berdasarkan nama
void tampilkanSatuBarang() {
    string namaBarang;
    bool barangDitemukan = false;

    do {
        cout << "_____ " << endl;
        cout << "Masukkan nama barang: ";
        cin >> namaBarang;

        for (Barang barang : daftarBarang) {
            if (barang.nama == namaBarang) {
                barang.displayBarang();
                barangDitemukan = true;
                break;
            }
        }

        if (!barangDitemukan) {
            cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl;
        }

    } while (!barangDitemukan);
}

// Mengubah informasi satu barang berdasarkan nama
void ubahBarang() {
    string namaBarang;
    bool barangDitemukan = false;

    do {
        cout << "_____ " << endl;

```

```

        cout << "Masukkan nama barang yang akan diubah: ";
        cin >> namaBarang;

        for (Barang &barang : daftarBarang) {
            if (barang.nama == namaBarang) {
                string nama;
                double hargaBeli, hargaJual;
                int stock;

                cout << "_____ " << endl;
                cout << "Masukkan nama baru: ";
                cin >> nama;
                cout << "Masukkan harga beli baru: ";
                cin >> hargaBeli;
                cout << "Masukkan harga jual baru: ";
                cin >> hargaJual;
                cout << "Masukkan stock baru: ";
                cin >> stock;

                barang.updateBarang(nama, hargaBeli, hargaJual, stock);
                cout << "SUCCESS [Barang berhasil diubah]" << endl;
                barangDitemukan = true;
                break;
            }
        }

        if (!barangDitemukan) {
            cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl;
        }

    } while (!barangDitemukan);
}

// Menghapus satu barang berdasarkan nama
void hapusBarang() {
    string namaBarang;
    bool barangDitemukan = false;

    do {
        cout << "_____ " << endl;
        cout << "Masukkan nama barang yang akan dihapus: ";
        cin >> namaBarang;

        for (auto it = daftarBarang.begin(); it != daftarBarang.end(); ++it) {
            if (it->nama == namaBarang) {
                it = daftarBarang.erase(it);
                cout << "SUCCESS [Barang berhasil dihapus]" << endl;
            }
        }
    } while (barangDitemukan == false);
}

```

```

        barangDitemukan = true;
        break;
    }
}

if (!barangDitemukan) {
    cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl;
}

} while (!barangDitemukan);
}

// Menambah stok satu barang berdasarkan nama
void kelolaStok() {
    string namaBarang;
    bool barangDitemukan = false;

    do {
        cout << "_____ " << endl;
        cout << "Masukkan nama barang yang akan direstock: ";
        cin >> namaBarang;

        for (Barang &barang : daftarBarang) {
            if (barang.nama == namaBarang) {
                int jumlah;
                cout << endl << "Masukkan jumlah restock: ";
                cin >> jumlah;
                barang.restock(jumlah);
                cout << "SUCCESS [Stok berhasil ditambahkan]" << endl;
                barangDitemukan = true;
                break;
            }
        }

        if (!barangDitemukan) {
            cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl;
        }

    } while (!barangDitemukan);
}

// Menangani penjualan satu barang berdasarkan nama
void kelolaPenjualan() {
    string namaBarang;
    bool barangDitemukan = false;

    do {

```

```

        cout << "_____ " << endl;
        cout << "Masukkan nama barang yang akan dijual: ";
        cin >> namaBarang;

        for (Barang &barang : daftarBarang) {
            if (barang.nama == namaBarang) {
                int jumlah;
                cout << endl << "Masukkan jumlah penjualan: ";
                cin >> jumlah;
                barang.sell(jumlah);
                barangDitemukan = true;
                break;
            }
        }

        if (!barangDitemukan) {
            cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl;
        }

    } while (!barangDitemukan);
}

};

int main() {
    Warung warung;
    string perintah;

    while (true) {
        cout << endl << "===== " << endl;
        cout << "+++++ MyBUYATI App +++++" << endl;
        cout << "===== " << endl;
        cout << "Menu:" << endl;
        cout << "[1] Add Supply" << endl;
        cout << "[2] Show All Supplies" << endl;
        cout << "[3] Show A Supply" << endl;
        cout << "[4] Update Supplies" << endl;
        cout << "[5] Remove Supplies" << endl;
        cout << "[6] Restock Supplies" << endl;
        cout << "[7] Sell Supplies" << endl;
        cout << "[8] Exit" << endl;
        cout << "===== " << endl;
        cout << "Monggo pilih menu : ";
        cin >> perintah;

        switch (stoi(perintah)) {
            case 1:
                warung.tambahBarang();

```

```

        break;
    case 2:
        warung.tampilkanSemuaBarang();
        break;
    case 3:
        warung.tampilkanSatuBarang();
        break;
    case 4:
        warung.ubahBarang();
        break;
    case 5:
        warung.hapusBarang();
        break;
    case 6:
        warung.kelolaStok();
        break;
    case 7:
        warung.kelolaPenjualan();
        break;
    case 8:
        cout << "Program berhenti." << endl;
        return 0;
    default:
        cout << "Perintah tidak valid." << endl;
        break;
    }
}

return 0;
}

```

OUTPUT

```

=====
+++++ MyBUYATI App +++++
=====
Menu:
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit

```

```

=====
Monggo pilih menu : 1

Masukkan nama barang: Molto
Masukkan harga beli: 1000
Masukkan harga jual: 2000
Masukkan stock: 300

SUCCESS [Barang berhasil ditambahkan]

=====
+++++ MyBUYATI App +++++
=====
Menu:
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit

=====
Monggo pilih menu : 1

Masukkan nama barang: Rinso
Masukkan harga beli: 1500
Masukkan harga jual: 2000
Masukkan stock: 100

SUCCESS [Barang berhasil ditambahkan]

=====
+++++ MyBUYATI App +++++
=====
Menu:
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit

=====
Monggo pilih menu : 2

```


Daftar Barang:

Nama: Molto
Harga Beli: 1000
Harga Jual: 2000
Stock: 300

Nama: Rinso
Harga Beli: 1500
Harga Jual: 2000
Stock: 100

=====
+++++ MyBUYATI App +++++
=====

Menu:

[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit

=====
Monggo pilih menu : 3

Masukkan nama barang: Molto

Nama: Molto
Harga Beli: 1000
Harga Jual: 2000
Stock: 300

=====
+++++ MyBUYATI App +++++
=====

Menu:

[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit

=====

```

Monggo pilih menu : 4
_____
Masukkan nama barang yang akan diubah: Rinso
_____
Masukkan nama baru: SoKlin
Masukkan harga beli baru: 2000
Masukkan harga jual baru: 3000
Masukkan stock baru: 500
_____
SUCCESS [Barang berhasil diubah]

=====
+++++ MyBUYATI App +++++
=====
Menu:
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit
=====
Monggo pilih menu : 5
_____
Masukkan nama barang yang akan dihapus: Molto
_____
SUCCESS [Barang berhasil dihapus]

=====
+++++ MyBUYATI App +++++
=====
Menu:
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies[8] Exit
=====
Monggo pilih menu : 6
_____
Masukkan nama barang yang akan direstock: SoKlin
Masukkan jumlah restock: 250
_____

```

SUCCESS [Stok berhasil ditambahkan]

```
=====
+++++ MyBUYATI App +++++
=====
```

Menu:

```
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit
```

```
=====
Monggo pilih menu : 7
```

Masukkan nama barang yang akan dijual: SoKlin
Masukkan jumlah penjualan: 250

SUCCESS [Penjualan berhasil]

```
=====
+++++ MyBUYATI App +++++
=====
```

Menu:

```
[1] Add Supply
[2] Show All Supplies
[3] Show A Supply
[4] Update Supplies
[5] Remove Supplies
[6] Restock Supplies
[7] Sell Supplies
[8] Exit
```

```
=====
Monggo pilih menu : 8
Program berhenti.
```

INSTRUKSI PEMAKAIAN

1. Tambah Barang:

- Pilih opsi 1 pada menu.
- Program akan meminta Anda untuk memasukkan detail barang seperti nama, harga beli, harga jual, dan jumlah stock.
- Barang baru akan ditambahkan ke dalam daftar.

2. Tampilkan Semua Barang:

- Pilih opsi 2 pada menu.
- Program akan menampilkan semua barang yang telah ditambahkan beserta detailnya.

3. Tampilkan Satu Barang:

- Pilih opsi 3 pada menu.
- Program akan meminta Anda memasukkan nama barang yang ingin ditampilkan.
- Jika barang dengan nama tersebut ada, detail barang akan ditampilkan. Jika tidak, Anda akan diminta memasukkan nama kembali.

4. Ubah Barang:

- Pilih opsi 4 pada menu.
- Program akan meminta Anda memasukkan nama barang yang ingin diubah.
- Jika barang dengan nama tersebut ada, program akan meminta Anda memasukkan detail baru seperti nama, harga beli, harga jual, dan jumlah stock.
- Jika tidak, Anda akan diminta memasukkan nama kembali.

5. Hapus Barang:

- Pilih opsi 5 pada menu.
- Program akan meminta Anda memasukkan nama barang yang ingin dihapus.
- Jika barang dengan nama tersebut ada, barang akan dihapus dari daftar.
- Jika tidak, Anda akan diminta memasukkan nama kembali.

6. Restock:

- Pilih opsi 6 pada menu.
- Program akan meminta Anda memasukkan nama barang yang ingin direstock dan jumlah restock.
- Jika barang dengan nama tersebut ada, stok barang akan ditambahkan.
- Jika tidak, Anda akan diminta memasukkan nama kembali.

7. Jual:

- Pilih opsi 7 pada menu.
- Program akan meminta Anda memasukkan nama barang yang ingin dijual dan jumlah penjualan.
- Jika barang dengan nama tersebut ada dan stok cukup, penjualan berhasil dan stok barang berkurang.
- Jika tidak atau stok tidak mencukupi, Anda akan diminta memasukkan nama kembali atau stok barang akan ditampilkan sebagai habis.

8. Keluar:

- Pilih opsi 8 pada menu untuk keluar dari program.

Note:

- Pastikan untuk memasukkan angka yang sesuai dengan opsi menu yang diinginkan.
- Input yang diminta harus sesuai dengan format yang diharapkan, seperti angka untuk harga dan stok.
- Program akan terus berjalan hingga Anda memilih opsi 8 untuk keluar.
- Program akan memberikan pesan kesalahan jika ada operasi yang tidak dapat dilakukan (misalnya, mencoba menjual barang yang stoknya habis).
- Jika daftar barang kosong, opsi untuk menampilkan semua barang akan memberi tahu bahwa tidak ada barang yang tersedia.

FITUR-FITUR

```
class Barang {
public:
    string nama;
    double hargaBeli;
    double hargaJual;
    int stock;
    // ...
};

class Warung {
private:
    vector<Barang> daftarBarang;

public:
    // ...
}
```

1. Kelas Barang (Class Barang)

- Attributes:
 - nama: Nama barang.
 - hargaBeli: Harga beli barang.
 - hargaJual: Harga jual barang.
 - stock: Jumlah stok barang.
- Methods:
 - displayBarang(): Menampilkan informasi barang ke layar.
 - updateBarang(): Mengupdate informasi barang.
 - restock(): Menambah stok barang.
 - sell(): Menangani penjualan barang.

2. Kelas Warung (Class Warung)

- Attributes:
 - daftarBarang: Vector yang menyimpan objek-objek Barang.
- Methods:
 - tambahBarang(): Menambah barang baru ke dalam daftar.
 - tampilkanSemuaBarang(): Menampilkan semua barang dalam daftar.
 - tampilkanSatuBarang(): Menampilkan informasi satu barang berdasarkan nama.
 - ubahBarang(): Mengubah informasi satu barang berdasarkan nama.
 - hapusBarang(): Menghapus satu barang berdasarkan nama.
 - kelolaStok(): Menambah stok satu barang berdasarkan nama.
 - kelolaPenjualan(): Menangani penjualan satu barang berdasarkan nama.

```

int main() {
    Warung warung;
    string perintah;

    while (true) {
        cout << endl << "======" << endl;
        cout << "+++++ MyBUYATI App +++++" << endl;
        cout << "======" << endl;
        cout << "Menu:" << endl;
        cout << "[1] Add Supply" << endl;
        cout << "[2] Show All Supplies" << endl;
        cout << "[3] Show A Supply" << endl;
        cout << "[4] Update Supplies" << endl;
        cout << "[5] Remove Supplies" << endl;
        cout << "[6] Restock Supplies" << endl;
        cout << "[7] Sell Supplies" << endl;
        cout << "[8] Exit" << endl;
        cout << "======" << endl;
        cout << "Monggo pilih menu : ";
        cin >> perintah;

        switch (stoi(perintah)) {
            case 1:
                warung.tambahBarang();
                break;
            case 2:
                warung.tampilkanSemuaBarang();
                break;
            case 3:
                warung.tampilkanSatuBarang();
                break;
            case 4:
                warung.ubahBarang();
                break;
            case 5:
                warung.hapusBarang();
                break;
            case 6:
                warung.kelolaStok();
                break;
            case 7:
                warung.kelolaPenjualan();
                break;
            case 8:
                cout << "Program berhenti." << endl;
                return 0;
        }
    }
}

```

```

        default:
            cout << "Perintah tidak valid." << endl;
            break;
    }
}

return 0;
}

```

3. main() Function

- Loop Utama (While Loop)
 - Program berjalan dalam loop tak terbatas hingga pengguna memilih opsi "Keluar."
- Switch Case
 - Digunakan untuk memproses pilihan yang diinput oleh pengguna.
- Input/Output menggunakan cin dan cout
 - Menggunakan cin untuk menerima input dari pengguna.
 - Menggunakan cout untuk menampilkan output ke layar.
- Penggunaan vector
 - vector digunakan untuk menyimpan daftar objek Barang.
- Error Handling
 - Program memberikan pesan kesalahan jika pengguna melakukan operasi yang tidak valid.
- Penggunaan Class dan Object
 - Memanfaatkan konsep pemrograman berorientasi objek (OOP) dengan membuat kelas Barang dan Warung.

4. Komentar dalam Source Code

- Setiap bagian dari kode diberikan komentar untuk memberikan penjelasan tentang fungsionalitasnya.
- Komentar membantu pembaca yang tidak terbiasa dengan kode tersebut untuk lebih mudah memahaminya.

5. Validasi Input

- Beberapa bagian program memiliki validasi input untuk memastikan input pengguna sesuai dengan yang diharapkan.

```

do {
    // ...
} while (!barangDitemukan);

```

6. Penggunaan do-while Loop

- Beberapa operasi, seperti mencari barang, menggunakan do-while loop agar pengguna dapat mencoba lagi jika input tidak valid.


```
case 8:
    cout << "Program berhenti." << endl;
    return 0;
```

7. Penggunaan return 0

- Program mengakhiri eksekusi dengan return 0 ketika pengguna memilih untuk keluar dari program.

```
stock += jumlah;
```

8. Menangani Ekspresi Kondisional

- Beberapa operasi, seperti menambah stok dan menjual barang, menggunakan ekspresi kondisional untuk menangani kasus khusus.

9. Penggunaan string

- Variabel string digunakan untuk menyimpan perintah dari pengguna.

10. Penggunaan std::endl

- Menggantikan penggunaan \n dengan std::endl untuk memberikan baris baru dalam output.

```
for (auto it = daftarBarang.begin(); it != daftarBarang.end(); ++it) {
    // ...
}
```

11. Penggunaan auto

- Pada beberapa bagian, auto digunakan untuk mempermudah deklarasi dan iterasi.

12. Penggunaan empty()

- Pada fungsi tampilkanSemuaBarang(), empty() digunakan untuk memeriksa apakah daftar barang kosong.

13. Validasi Input Angka

- Ketika meminta harga, stok, jumlah restock, dan jumlah penjualan, program memastikan bahwa input yang diberikan adalah angka.

NO	MATERI	PERAN
1.	Header Files dan Namespace	<pre>#include <iostream> #include <vector> using namespace std;</pre> <p>Header files (#include <iostream> dan #include <vector>) digunakan untuk menyertakan file-file header yang memuat definisi fungsi dan objek yang dibutuhkan. using namespace std; digunakan agar tidak perlu menuliskan std:: sebelum setiap penggunaan objek atau fungsi dari namespace std.</p>
2.	Class dan Object	<pre>class Barang { // ... }; class Warung { // ... }</pre> <p>Dua kelas (`Barang` dan `Warung`) didefinisikan. Kelas `Barang` digunakan untuk merepresentasikan informasi tentang satu barang, sementara `Warung` digunakan untuk mengelola daftar barang-barang.</p>
3.	Constructor	<pre>Barang(string n, double hb, double hj, int s) : nama(n), hargaBeli(hb), hargaJual(hj), stock(s) {}</pre> <p>Konstruktor digunakan untuk menginisialisasi objek saat objek tersebut dibuat. Konstruktor `Barang` menerima beberapa parameter untuk menginisialisasi atribut-atributnya.</p>
4.	Vector	<pre>vector<Barang> daftarBarang;</pre> <p>`vector` digunakan untuk menyimpan dan mengelola daftar objek `Barang`. Dengan menggunakan `vector`, kita dapat dengan mudah menambah, menghapus, dan mengakses elemen-elemen di dalamnya.</p>

5.	`while` Loop	<pre>while (true) { // ... }</pre> <p>`while` loop digunakan untuk membuat program berjalan terus menerus selama kondisi yang diberikan (dalam hal ini, `true`) terpenuhi.</p>
6.	`switch` Statement	<pre>switch (stoi(perintah)) { // ... }</pre> <p>`switch` statement digunakan untuk membuat percabangan berdasarkan nilai ekspresi. Dalam hal ini, nilai ekspresi adalah pilihan menu yang dimasukkan oleh pengguna.</p>
7.	`do-while` Loop	<pre>do { // ... } while (!barangDitemukan);</pre> <p>`do-while` loop digunakan untuk membuat loop yang akan dieksekusi minimal satu kali. Dalam hal ini, digunakan untuk meminta input ulang jika barang tidak ditemukan.</p>
8.	Input/Output	<pre>cout << "Masukkan nama barang: "; cin >> nama;</pre> <p>`cout` digunakan untuk menampilkan pesan ke layar, sedangkan `cin` digunakan untuk menerima input dari pengguna. Ini digunakan untuk berinteraksi dengan pengguna saat menambah, mengubah, atau menghapus barang.</p>
9.	Error Handling	<pre>if (!barangDitemukan) { cout << "_____ " << endl; cout << "ERROR [Barang tidak ditemukan. Coba lagi]" << endl; }</pre>

		Menggunakan kondisi untuk memberikan pesan kesalahan kepada pengguna dan meminta input ulang jika suatu operasi tidak dapat dilakukan.
10.	`for` Loop	<pre>for (Barang barang : daftarBarang) { barang.displayBarang(); }</pre> <p>Menggunakan `for` loop untuk mengakses dan menampilkan setiap barang dalam `vector daftarBarang`.</p>
11.	`main` Function	<pre>int main() { // ... return 0; }</pre> <p>Fungsi `main` adalah titik masuk utama program. Di dalamnya, program melakukan loop untuk memproses pilihan-pilihan menu yang dimasukkan oleh pengguna.</p>