

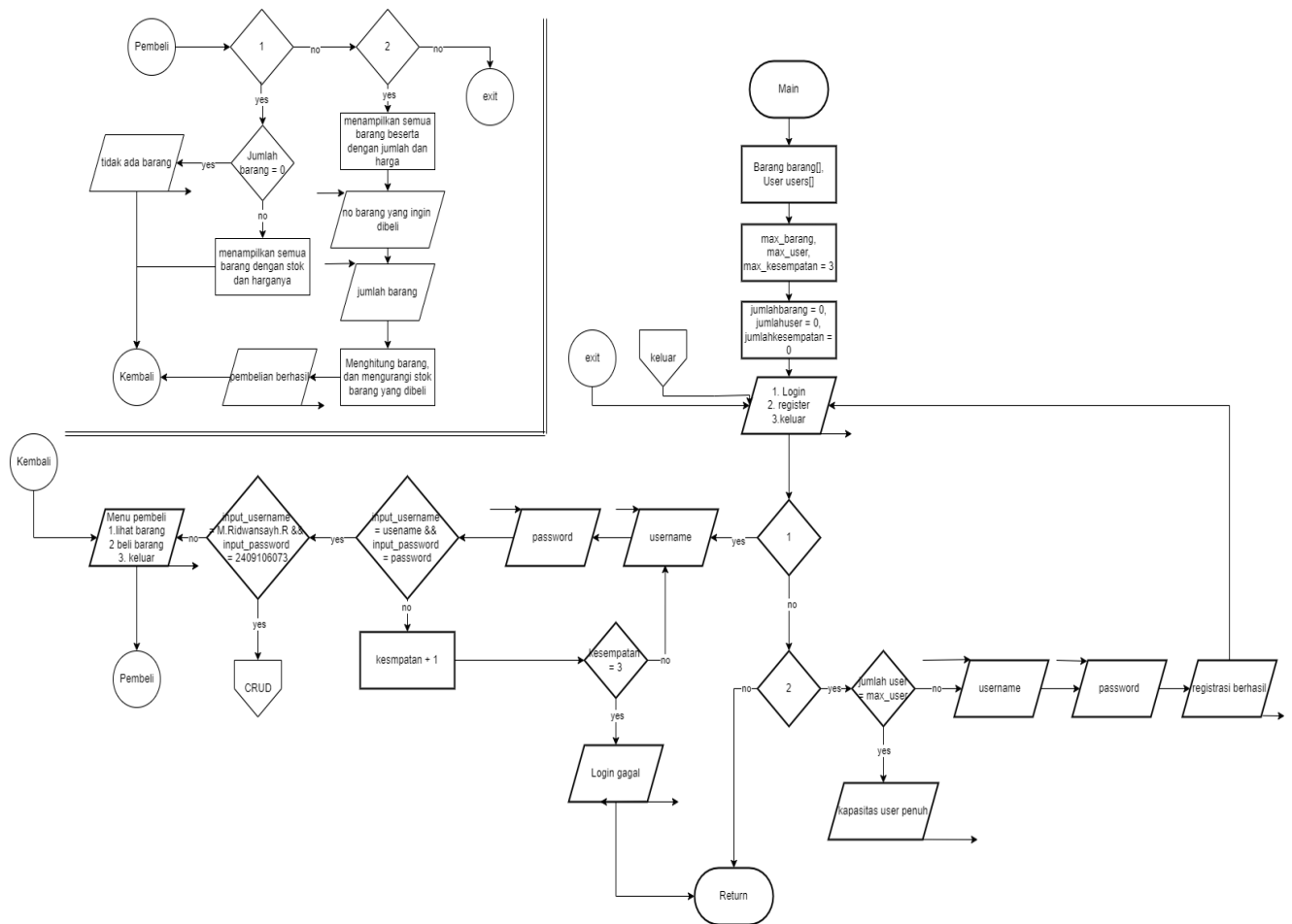
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



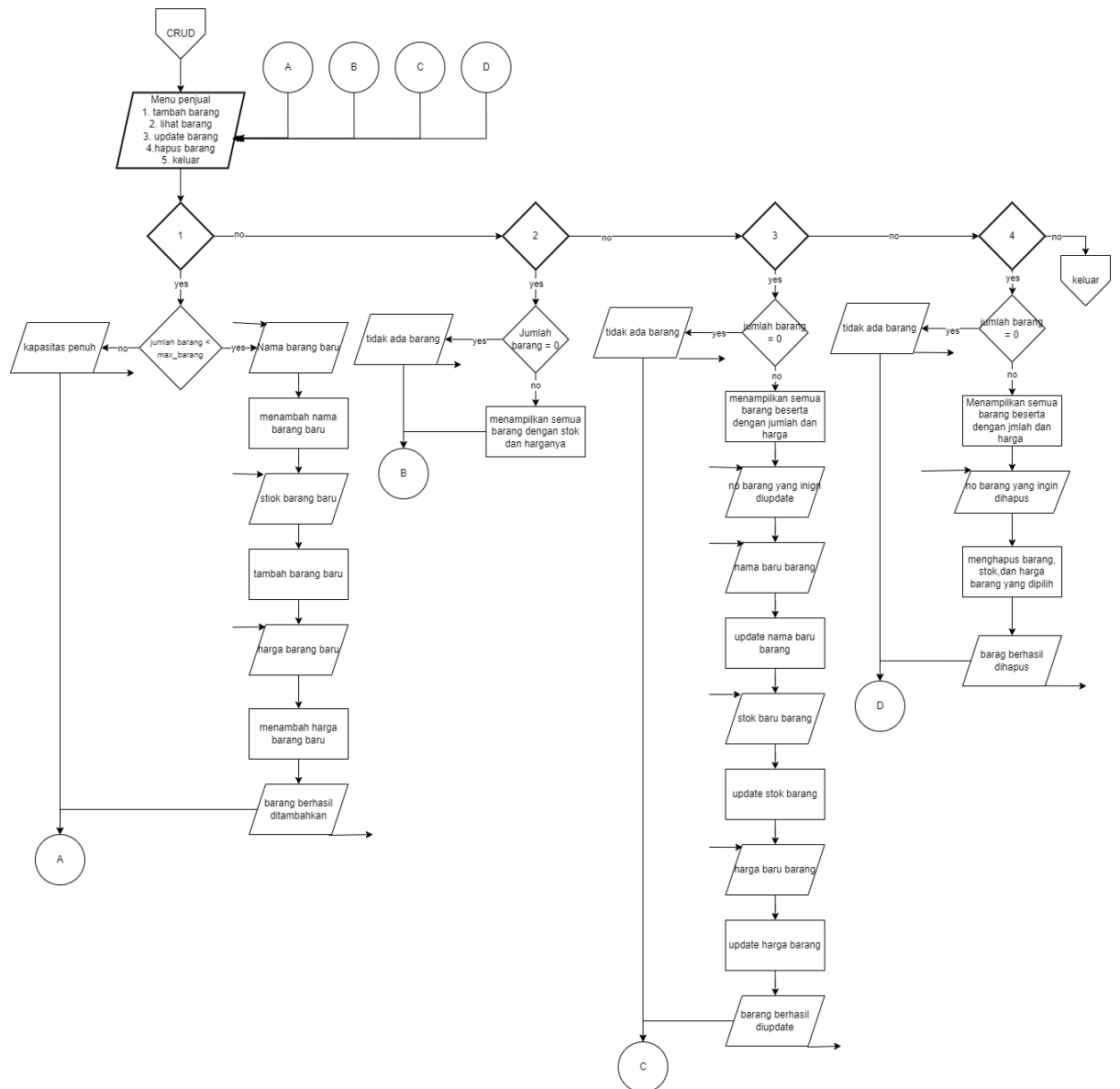
Disusun oleh:
Muhammad Ridwanansyah Rahman (2409106073)
Kelas (B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1.1 Flowchart Menu Login, registrasi, dan Pembeli



Gambar 1.2 Flowchart menu CRUD

2. Analisis Program

Tujuan utama program ini adalah membangun sistem yang membantu dalam manajemen barang sembako dan dapat digunakan oleh penjual maupun pembeli.

Fungsi utama program ini adalah:

- **Manajemen Barang:** Program ini memungkinkan penjual untuk menambah, melihat, mengupdate, dan menghapus barang sembako.
- **Pembelian Barang:** Program ini memungkinkan pembeli untuk melihat daftar barang sembako yang tersedia dan melakukan pembelian.
- **Manajemen Akun:** Program ini memungkinkan pengguna untuk mendaftar dan login ke dalam sistem.

Dengan demikian, program ini dapat membantu penjual dalam mengelola stok barang sembako dan memudahkan pembeli dalam melakukan pembelian.

3. Source Code

A. Fitur Registrasi

Fitur ini digunakan pembeli untuk membuat akun pertama kali, yang dimana program pertama-tama mengecek jumlah user yang terdaftar, kemudian program meminta input berupa username dan password yang kemudian akan mengeluarkan pesan output “Registrasi Berhasil” dan mengarahkan pembeli ke menu awal untuk melakukan login.

Source Code:

```
void registrasi() {
    if (jumlahUser < MAX_USER) {
        cin.ignore();
        system("cls");
        cout << "\n===== " << endl;
        cout << "          REGISTRASI AKUN          " << endl;
        cout << "===== " << endl;
        cout << "Username: ";
        getline(cin, users[jumlahUser].username);
        cout << "Password: ";
        getline(cin, users[jumlahUser].password);
        users[jumlahUser].role = "pembeli";
        jumlahUser++;
        system("cls");
        cout << "\n===== " << endl;
        cout << "          REGISTRASI BERHASIL          " << endl;
        cout << "===== " << endl;
        cout << "Selamat, Anda telah berhasil mendaftar!" << endl;
        cout << "===== " << endl;
        cout << "Klik Enter untuk kembali ke menu...";
        cin.ignore();
    } else {
        cout << "Kapasitas user penuh!\n";
        cout << "Klik Enter untuk melanjutkan...";
        cin.ignore();
        cin.get();
    }
}
```

B. Fitur Login

Fitur ini digunakan pengguna untuk masuk program dengan cara menginput username dan password yang sesuai dan nantinya pengguna akan diarahkan kemenu utama sesuai dengan role yang dimiliki user. Pengguna diberikan 3 kesempatan untuk memasukkan username dan password dengan benar, jika lewat dari 3 maka pengguna otomatis keluar dari program.

Source Code:

```
void login() {
    string username, password;
    int kesempatan = 0;
    const int max_kesempatan = 3;
    bool loginBerhasil = false;
    string role;

    while (kesempatan < max_kesempatan && !loginBerhasil) {
        cin.ignore();
        system("cls");
        cout << "\n===== " << endl;
        cout << "                Login Program                " << endl;
        cout << "===== " << endl;
        cout << "Username: ";
        getline(cin, username);
        cout << "Password: ";
        getline(cin, password);

        if (username == "Muhammad Ridwanansyah Rahman" && password ==
"2409106073") {
            loginBerhasil = true;
            role = "penjual";
        } else {
            for (int i = 0; i < jumlahUser; i++) {
                if (users[i].username == username && users[i].password ==
password) {
                    loginBerhasil = true;
                    role = users[i].role;
                    break;
                }
            }
        }
        if (loginBerhasil) {
            system("cls");
            cout << "\n===== " << endl;
            cout << "                Login Berhasil                " << endl;
            cout << "===== " << endl;
            cout << "Selamat datang, " << username << "!" << endl;
            cout << "Anda login sebagai " << role << "." << endl;
        }
    }
}
```

```

        cout << "=====" << endl;
        cout << "Klik Enter untuk melanjutkan...";
        cin.ignore();

        if (role == "penjual") {
            menuPenjual();
        } else if (role == "pembeli") {
            menuPembeli();
        }
    } else {
        kesempatan++;
        cout << "Login gagal. Sisa percobaan: " << max_kesempatan -
kesempatan << "\n";
        cout << "Klik Enter...";
        cin.clear();
    }
}

if (kesempatan == max_kesempatan) {
    system("cls");
    cout << "\n=====" << endl;
    cout << "          Login Gagal          " << endl;
    cout << "=====" << endl;
    cout << "Anda telah mencoba login " << kesempatan << " kali." << endl;
    cout << "Program berhenti karena batas" << endl;
    cout << "percobaan telah tercapai." << endl;
    cout << "=====" << endl;
    running = false;
}
}

```

C. Menambah Barang

kode dalam program ini digunakan penjual untuk menambah barang sembako baru beserta dengan stok dan harganya.

Source Code:

```

void tambahBarang() {
    if (jumlahBarang < MAX_BARANG) {
        cin.ignore();
        cout << "Masukkan Nama Barang: ";
        getline(cin, barang[jumlahBarang].nama);
    }
}

```

```

int stok, harga;
bool valid = false;

while (!valid) {
    cout << "Masukkan Stok Barang: ";
    if (cin >> stok) {
        if (stok > 0) {
            valid = true;
            barang[jumlahBarang].detail.stok = stok;
        } else {
            cout << "Stok tidak boleh 0 atau kurang!\n";
            cin.clear();
            cin.ignore(10000, '\n');
        }
    } else {
        cout << "Input harus berupa ANGKA!!!\n";
        cin.clear();
        cin.ignore(10000, '\n');
    }
}

valid = false;
while (!valid){
    cout << "Masukkan Harga Barang: ";
    if (cin >> harga) {
        if (harga > 0) {
            valid = true;
            barang[jumlahBarang].detail.harga = harga;
        } else {
            cout << "Harga tidak boleh 0 atau kurang!\n";
            cin.clear();
            cin.ignore(10000, '\n');
        }
    } else {
        cout << "Input harus berupa ANGKA!!!\n";
        cin.clear();
        cin.ignore(10000, '\n');
    }
}

jumlahBarang++;
system("cls");
cout << "\n===== " << endl;
cout << "          BARANG BERHASIL DITAMBAHKAN          " << endl;
cout << "===== " << endl;
cout << "Barang dengan nama " << barang[jumlahBarang-1].nama << endl;
cout << "Jumlah barang: " << barang[jumlahBarang-1].detail.stok << endl;
cout << "Harga barang: " << barang[jumlahBarang-1].detail.harga << endl;
cout << "===== " << endl;
cout << "Klik Enter untuk kembali ke menu utama..." ;
cin.ignore();

```



```

        cin.get();
    } else {
        cout << "Kapasitas barang penuh!\n";
        cout << "Klik Enter untuk kembali ke menu...";
        cin.ignore();
        cin.get();
    }
}

```

D. Menampilkan Barang

Kode ini digunakan untuk menampilkan daftar barang sembako beserta stok dan harganya dalam format tabel dengan menggunakan perulangan for

Source Code:

```

void tampilkanBarang() {
    for (int i = 0; i < jumlahBarang; i++) {
        cout << setw(5) << i + 1 << setw(20) << barang[i].nama
            << setw(10) << barang[i].detail.stok
            << setw(10) << barang[i].detail.harga << "\n";
    }
}

void lihatBarang() {
    if (jumlahBarang == 0) {
        cout << "Tidak ada barang yang tersedia.\n";
    } else {
        system("cls");
        cout << "\n===== " << endl;
        cout << "                DAFTAR BARANG SEMBAKO                " << endl;
        cout << "===== " << endl;
        cout << setw(5) << "No" << setw(20) << "Nama Barang" << setw(10) <<
"Stok" << setw(10) << "Harga" << "\n";
        tampilkanBarang();
        cout << "===== " << endl;
    }
    cout << "Klik Enter untuk kembali ke menu...";
    cin.ignore();
    cin.get();
}

```

E. Update Barang

Kode ini memungkinkan pengguna untuk melakukan update/mengganti data nama, stok serta harga dari barang sembako yang dipilih dan sudah ada sebelumnya.

Pada fungsi `updateBarang`, pointer digunakan untuk memodifikasi data barang secara langsung melalui alamat memori. Ketika pengguna memilih barang yang ingin diupdate, program menggunakan operator `&` untuk mendapatkan alamat elemen array barang yang dipilih. Alamat ini kemudian diteruskan ke fungsi `updateStokBarang` dan `updateHargaBarang`, yang memanfaatkan pointer untuk mengakses dan memperbarui nilai stok dan harga barang secara langsung tanpa membuat salinan data. Hal ini memastikan efisiensi memori dan perubahan langsung pada data asli.

Source Code:

```
void updateStokBarang(Barang *barangPtr, int newStok) {
    if (barangPtr != nullptr) {
        barangPtr->detail.stok = newStok;
    }
}

void updateHargaBarang(Barang *barangPtr, int newHarga) {
    if (barangPtr != nullptr) {
        barangPtr->detail.harga = newHarga;
    }
}

void updateBarang() {
    if (jumlahBarang > 0) {
        int index;
        bool valid = false;
        while (!valid) {
            system("cls");
            cout << "\n===== " << endl;
            cout << "          DAFTAR BARANG SEMBAKO          " << endl;
            cout << "===== " << endl;
            cout << setw(5) << "No" << setw(20) << "Nama Barang" << setw(10) <<
"Stok" << setw(10) << "Harga" << "\n";
            tampilkanBarang();
            cout << "===== " << endl;

            cout << "Masukkan nomor barang yang ingin diupdate: ";
            if (cin >> index) {
```

```

        if (index > 0 && index <= jumlahBarang) {
            valid = true;
        } else {
            cout << "Nomor barang tidak valid. Silakan pilih nomor yang
tersedia.\n";

            cout << "Klik Enter...";
            cin.ignore();
            cin.get();
            return;
        }
    } else {
        cout << "Input harus berupa angka. Silakan pilih nomor yang
tersedia.\n";

        cout << "Klik Enter untuk coba lagi...";
        cin.clear();
        cin.ignore(10000, '\n');
        cin.get();
    }
}

cout << "Masukkan Nama Barang Baru: ";
cin.ignore();
getline(cin, barang[index - 1].nama);
int stokBaru;
valid = false;
while (!valid) {
    cout << "Masukkan Stok Barang Baru: ";
    if (cin >> stokBaru) {
        if (stokBaru > 0) {
            valid = true;
            updateStokBarang(&barang[index - 1], stokBaru);
        } else {
            cout << "Stok barang tidak boleh 0 atau kurang!!\n";
            cin.clear();
            cin.ignore(10000, '\n');
        }
    } else {
        cout << "Input harus berupa angka!!\n";
        cin.clear();
        cin.ignore(10000, '\n');
    }
}

int hargaBaru;
valid = false;
while (!valid) {
    cout << "Masukkan Harga Barang Baru: ";
    if (cin >> hargaBaru) {
        if (hargaBaru > 0) {
            valid = true;
            updateHargaBarang(&barang[index - 1], hargaBaru);
        } else {
            cout << "Harga barang tidak boleh 0 atau kurang!!\n";

```

```

        cin.clear();
        cin.ignore(10000, '\n');
    }
} else {
    cout << "Input harus berupa angka!!\n";
    cin.clear();
    cin.ignore(10000, '\n');
}
}
system("cls");
cout << "\n===== " << endl;
cout << "                BARANG BERHASIL DIUPDATE                " << endl;
cout << "===== " << endl;
cout << "Klik Enter untuk kembali ke menu...";
cin.ignore();
cin.get();
} else {
    cout << "Tidak ada barang yang tersedia.\n";
    cout << "Klik Enter untuk kembali ke menu...";
    cin.ignore();
    cin.get();
}
}
}

```

F. Hapus Barang

Kode ini memungkinkan pengguna untuk melakukan delete/menghapus data sembako yang dipilih pengguna, berupa nama, stok, serta harga barang sembako yang ada sebelumnya.

Pointer hapusBarang pada void hapusBarang() digunakan untuk menunjuk langsung ke elemen array barang yang akan dihapus berdasarkan indeks yang dipilih oleh pengguna (index - 1). Dengan menggunakan pointer ini, Data barang yang akan dihapus tetap dapat diakses meskipun elemen array digeser. Informasi barang yang dihapus (seperti nama, stok, dan harga) dapat ditampilkan kepada pengguna setelah proses penghapusan selesai.

Source Code:

```

void hapusBarang() {
    if (jumlahBarang > 0) {
        int index;
        bool valid = false;
        while (!valid) {
            system("cls");
            cout << "\n===== " <<

```

```

endl;
        cout << "                DAFTAR BARANG SEMBAKO                " << endl;
        cout << "===== " <<
endl;
        cout << setw(5) << "No" << setw(20) << "Nama Barang" << setw(10)
<< "Stok" << setw(10) << "Harga" << "\n";
        tampilkanBarang();
        cout << "===== " <<
endl;

        cout << "Masukkan nomor barang yang ingin dihapus: ";
        if (cin >> index) {
            if (index > 0 && index <= jumlahBarang) {
                valid = true;
            } else {
                cout << "Nomor barang tidak valid. Silakan pilih nomor
yang tersedia.\n";
                cout << "Klik Enter...";
                cin.ignore();
                cin.get();
                return;
            }
        } else {
            cout << "Input harus berupa angka. Silakan pilih nomor yang
tersedia.\n";

            cout << "Klik Enter untuk coba lagi...";
            cin.clear();
            cin.ignore(10000, '\n');
            cin.get();
        }
    }
    Barang *hapusBarang = &barang[index - 1];
    for (int i = index - 1; i < jumlahBarang - 1; i++) {
        barang[i] = barang[i + 1];
    }
    jumlahBarang--;
    system("cls");
    cout << "\n===== " << endl;
    cout << "                BARANG BERHASIL DIHAPUS                " << endl;
    cout << "===== " << endl;
    cout << "Barang dengan nama " << hapusBarang->nama << "\n";
    cout << "Jumlah barang : " << hapusBarang->detail.stok << "\n";
    cout << "Harga barang : " << hapusBarang->detail.harga << "\n";
    cout << "===== " << endl;
    cout << "Klik Enter untuk kembali ke menu...";
    cin.ignore();
    cin.get();
} else {

```

```

        cout << "Tidak ada barang yang tersedia.\n";
        cout << "Klik Enter untuk kembali ke menu...";
        cin.ignore();
        cin.get();
    }
}

```

G. Membeli Barang

Pada kode ini program akan menampilkan daftar barang yang beserta sisa stok dan harganya, yang kemudian pembeli disuruh untuk menginput nomor barang yang ingin dibeli serta banyaknya. Kemudian program akan mengeluarkan output berupa total harga yang harus dibayar pembeli.

Pada fungsi `hapusBarang`, pointer digunakan untuk menunjuk langsung ke elemen array barang yang akan dihapus berdasarkan indeks yang dipilih oleh pengguna. Program menggunakan operator `&` untuk mendapatkan alamat elemen array tersebut. Pointer ini memungkinkan program tetap mengakses data barang yang akan dihapus, seperti nama, stok, dan harga, meskipun elemen array telah digeser untuk mengisi posisi yang kosong. Hal ini memastikan efisiensi memori karena penggunaan pointer memungkinkan program untuk bekerja langsung dengan alamat memori data asli, tanpa membuat salinan data. Misalnya, pada fungsi seperti `updateBarang`, `hapusBarang`, atau `beliBarang`, pointer digunakan untuk menunjuk ke elemen array barang. Dengan cara ini, program tidak perlu menduplikasi elemen array saat memodifikasi atau mengaksesnya, sehingga menghemat penggunaan memori dan meningkatkan kinerja program.

Source Code:

```

int hitungTotalHarga(int* harga, int* jumlah) {
    if (*jumlah == 0) {
        return 0;
    }
    (*jumlah)--;
    return *harga + hitungTotalHarga(harga, jumlah);
}

void beliBarang() {

```

```

if (jumlahBarang > 0) {
    int index, jumlahBeli;
    bool valid = false;
    while (!valid) {
        system("cls");
        cout << "\n===== " << endl;
        cout << "                DAFTAR BARANG SEMBAKO                " << endl;
        cout << "===== " << endl;
        cout << setw(5) << "No" << setw(20) << "Nama Barang" << setw(10) << "Stok"
<< setw(10) << "Harga" << "\n";
        tampilkanBarang();
        cout << "===== " << endl;

        cout << "Masukkan nomor barang yang ingin dibeli: ";
        if (cin >> index) {
            if (index > 0 && index <= jumlahBarang) {
                cout << "Masukkan jumlah yang ingin dibeli: ";
                if (cin >> jumlahBeli) {
                    if (jumlahBeli > 0 && jumlahBeli <= barang[index -
1].detail.stok) {
                        valid = true;
                        Barang *barangBeli = &barang[index - 1];
                        barangBeli->detail.stok -= jumlahBeli;
                        int totalHarga = hitungTotalHarga(&barangBeli-
>detail.harga, &jumlahBeli);

                        system("cls");
                        cout << "\n===== "
<< endl;

                        cout << "                PEMBELIAN BERHASIL                " <<
endl;

                        cout << "===== " <<
endl;

                        cout << "Total harga: Rp " << totalHarga << ",-" << endl;
                        cout << "Silahkan lakukan pembayaran..." << endl;
                        cout << "===== " <<
endl;

                        cout << "Klik Enter untuk kembali ke menu...";
                        cin.ignore();
                        cin.get();
                    } else {
                        if (jumlahBeli > barang[index - 1].detail.stok) {
                            cout << "Maaf, jumlah stok tidak mencukupi.\n";
                            cout << "Klik Enter...";
                            cin.ignore();
                            cin.get();
                        } else {
                            cout << "Jumlah yang dibeli tidak boleh 0 atau
negatif!!.\n";

                            cout << "Klik Enter...";
                            cin.ignore();

```

```

        cin.get();
    }
}
} else {
    cout << "Input harus berupa angka.\n";
    cout << "Klik Enter...";
    cin.clear();
    cin.ignore(10000, '\n');
    cin.get();
}
} else {
    cout << "Nomor barang tidak valid. Silakan pilih nomor yang
tersedia.\n";

    cout << "Klik Enter...";
    cin.ignore();
    cin.get();
    return;
}
} else {
    cout << "Input harus berupa angka. Silakan pilih nomor yang
tersedia.\n";

    cout << "Klik Enter untuk coba lagi...";
    cin.clear();
    cin.ignore(10000, '\n');
    cin.get();
}
}
} else {
    cout << "Tidak ada barang yang tersedia.\n";
    cout << "Klik Enter untuk kembali ke menu...";
    cin.ignore();
    cin.get();
}
}
}

```


4. Uji Coba dan Hasil Output

```
PS E:\kerja posttest> cd "e:\kerja posttest\" ; if ($?) {
=====
                        SELAMAT DATANG
=====
1. Login
2. Register
3. Keluar
=====
Pilih menu: 2

=====
                        REGISTRASI AKUN
=====
Username: ridwan
Password: 12345

=====
                        REGISTRASI BERHASIL
=====
Selamat, Anda telah berhasil mendaftar!
=====
Klik Enter untuk kembali ke menu...
=====
```

Gambar 4.1 Registrasi akun pembeli

```
=====
                        SELAMAT DATANG
=====
1. Login
2. Register
3. Keluar
=====
Pilih menu: 1

=====
                        Login Program
=====
Username: Muhammad Ridwanansyah Rahman
Password: 2409106073

=====
                        Login Berhasil
=====
Selamat datang, Muhammad Ridwanansyah Rahman!
Anda login sebagai penjual.
=====
Klik Enter untuk melanjutkan...
=====
```

Gambar 4.2 Login penjual

```
=====
                        SELAMAT DATANG
=====
1. Login
2. Register
3. Keluar
=====
Pilih menu: 1

=====
                        Login Program
=====
Username: ridwan
Password: 12345

=====
                        Login Berhasil
=====
Selamat datang, ridwan!
Anda login sebagai pembeli.
=====
Klik Enter untuk melanjutkan...|
```

Gambar 4.3 Login pembeli

```
=====
                        MENU MANAJEMEN BARANG SEMBAKO
=====
1. Tambah Barang
2. Lihat Barang
3. Update Barang
4. Hapus Barang
5. Keluar
=====
Pilih menu: 1
Masukkan Nama Barang: Cuka manis
Masukkan Stok Barang: 25
Masukkan Harga Barang: 15000

=====
                        BARANG BERHASIL DITAMBAHKAN
=====
Barang dengan nama Cuka manis
Jumlah barang: 25
Harga barang: 15000
=====
Klik Enter untuk kembali ke menu utama...|
```

Gambar 4.4 Tambah barang

```
=====
                        MENU MANAJEMEN BARANG SEMBAKO
=====
1. Tambah Barang
2. Lihat Barang
3. Update Barang
4. Hapus Barang
5. Keluar
=====
Pilih menu: 2

=====
                        DAFTAR BARANG SEMBAKO
=====
No      Nama Barang   Stok   Harga
1       Beras        100    5000
2       Gula Pasir     50     2000
3       Minyak Goreng 20     15000
4       Cuka manis     25     15000
=====
Klik Enter untuk kembali ke menu...|
```

Gambar 4.5 Tampilkan data barang

```
=====
MENU MANAJEMEN BARANG SEMBAKO
=====
1. Tambah Barang
2. Lihat Barang
3. Update Barang
4. Hapus Barang
5. Keluar
=====
Pilih menu: 3

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2        Gula Pasir       50        2000
3        Minyak Goreng   20        15000
4        Cuka manis       25        15000
=====
Masukkan nomor barang yang ingin diupdate: 4
Masukkan Nama Barang Baru: Gula Pahit
Masukkan Stok Barang Baru: 25
Masukkan Harga Barang Baru: 10000

=====
BARANG BERHASIL DIUPDATE
=====
Klik Enter untuk kembali ke menu...
```

Gambar 4.6 Update data barang

```
=====
MENU MANAJEMEN BARANG SEMBAKO
=====
1. Tambah Barang
2. Lihat Barang
3. Update Barang
4. Hapus Barang
5. Keluar
=====
Pilih menu: 4

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2        Gula Pasir       50        2000
3        Minyak Goreng   20        15000
4        Gula Pahit       25        10000
=====
Masukkan nomor barang yang ingin dihapus: 4

=====
BARANG BERHASIL DIHAPUS
=====
Barang dengan nama Gula Pahit
Jumlah barang : 25
Harga barang : 10000
=====
Klik Enter untuk kembali ke menu...
```

Gambar 4.7 Hapus data barang

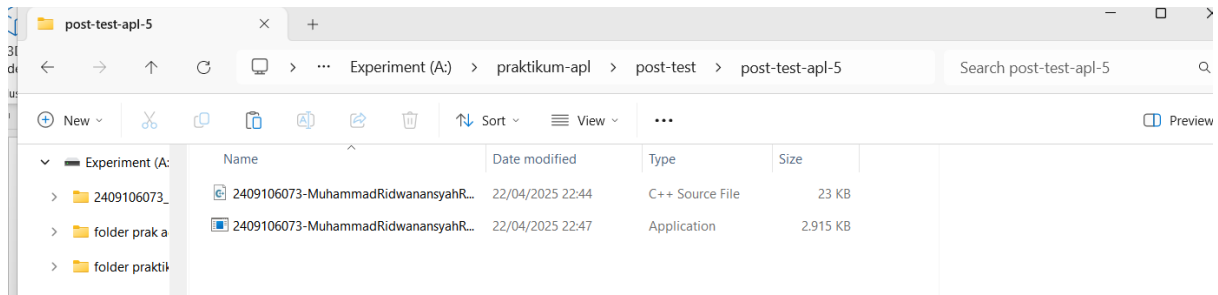
```
=====
                        DAFTAR BARANG SEMBAKO
=====
No          Nama Barang      Stok      Harga
1           Beras            96        5000
2           Gula Pasir       50        2000
3           Minyak Goreng    20       15000
=====
Masukkan nomor barang yang ingin dibeli: 3
Masukkan jumlah yang ingin dibeli: 3

=====
                        PEMBELIAN BERHASIL
=====
Total harga: Rp 45000,-
Silahkan lakukan pembayaran...

=====
Klik Enter untuk kembali ke menu...|
```

Gambar 4.8 Proses Pembelian

5. Langkah-Langkah Git



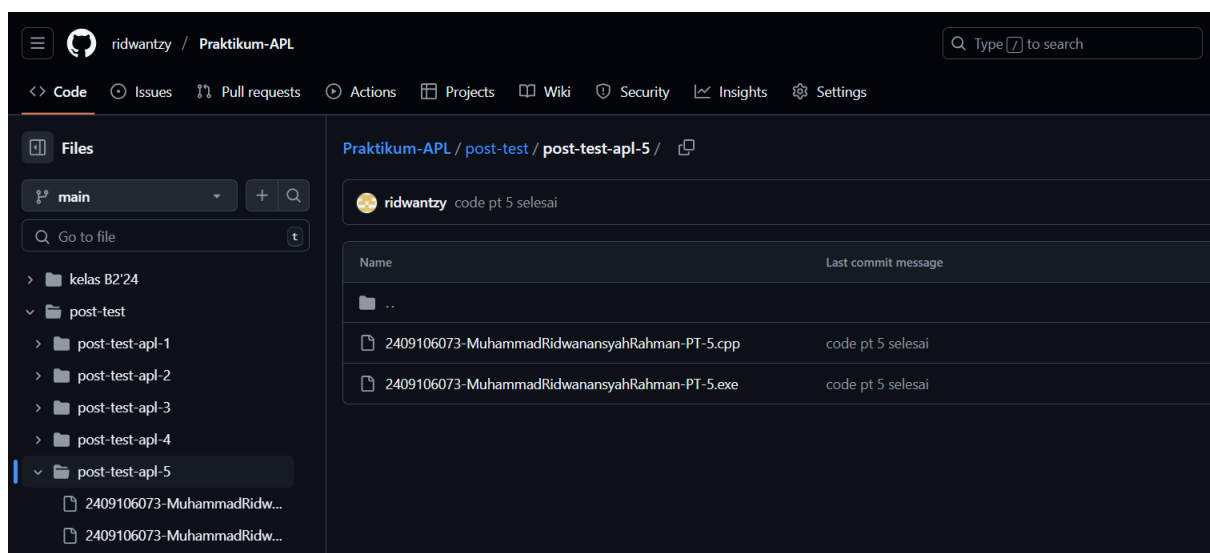
Gambar 5.1 Membuat folder post-test-apl-5 yang berisi file .cpp dan .exe dalam file lokal praktikum-apl

```
PS A:\praktikum-apl> git add .
PS A:\praktikum-apl> git commit -m "code pt 5 selesai"
[main 20706bc] code pt 5 selesai
2 files changed, 629 insertions(+)
create mode 100644 post-test/post-test-apl-5/2409106073-MuhammadRidwanansyahRahman-PT-5.cpp
create mode 100644 post-test/post-test-apl-5/2409106073-MuhammadRidwanansyahRahman-PT-5.exe
```

Gambar 5.2 Buka folder praktikum-apl dengan terminal lalu melakukan git add dan git commit

```
PS A:\praktikum-apl> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 12 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (6/6), 683.39 KiB | 4.24 MiB/s, done.
Total 6 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/ridwantzy/Praktikum-APL.git
 3f0bf9e..20706bc  main -> main
branch 'main' set up to track 'origin/main'.
PS A:\praktikum-apl> |
```

Gambar 5.3 Melakukan git push



Gambar 5.4 Posttest 5 berhasil ditambahkan