

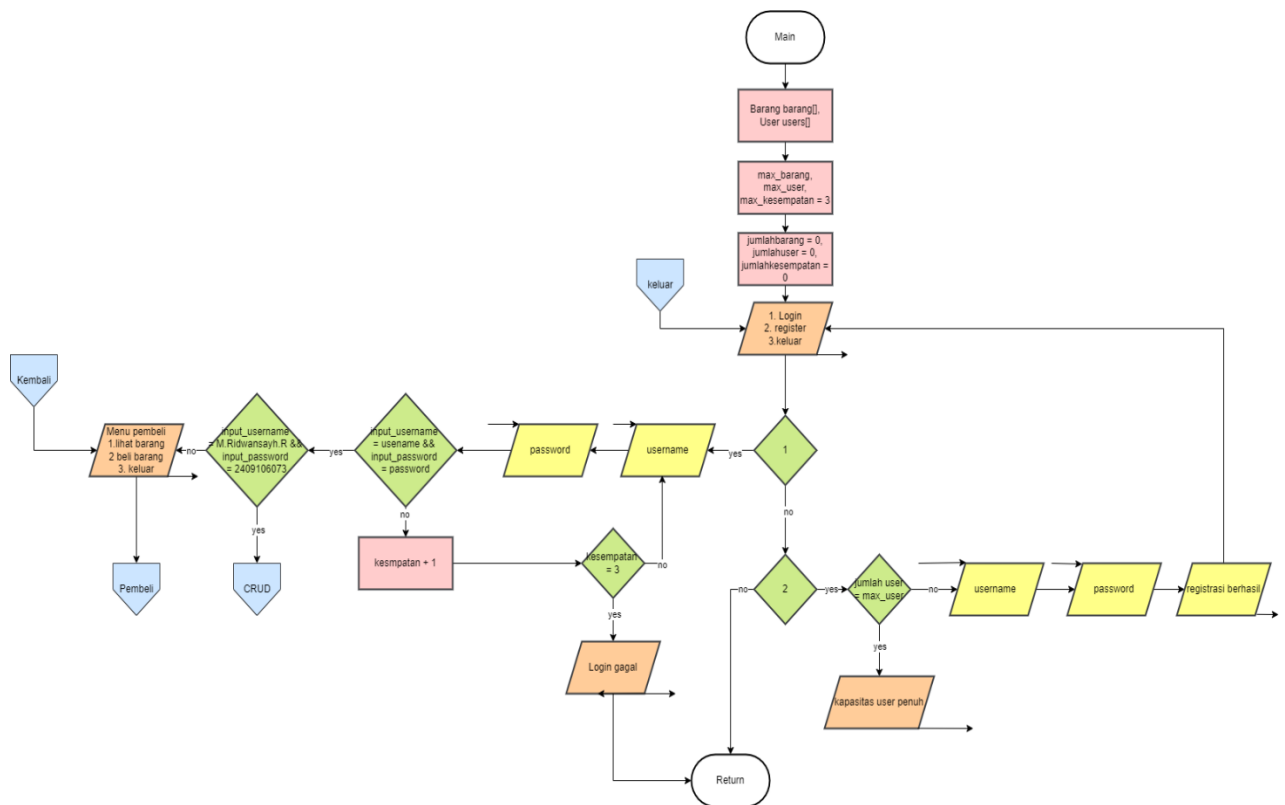
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



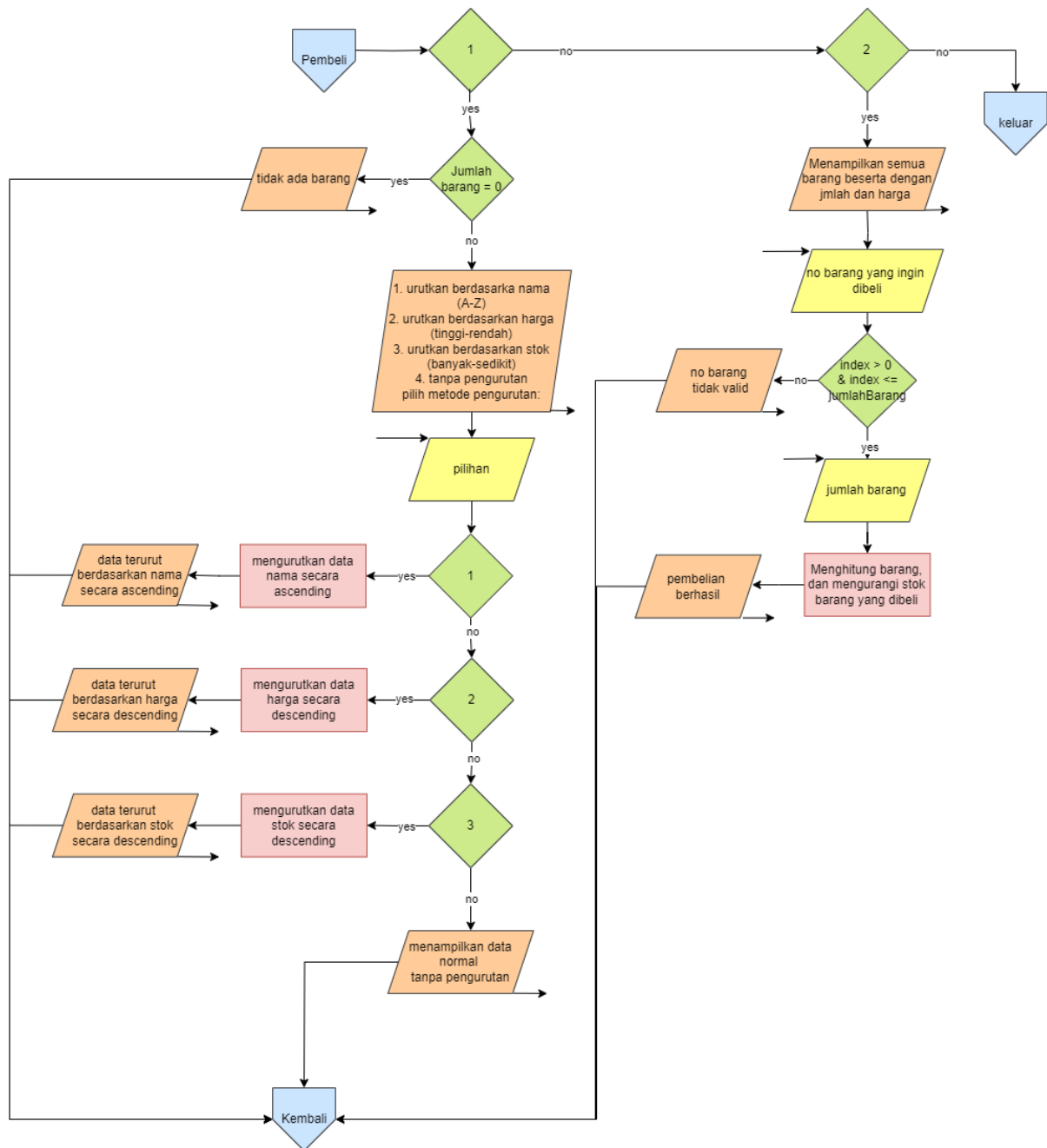
Disusun oleh:
Muhammad Ridwanansyah Rahman (2409106073)
Kelas (B2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

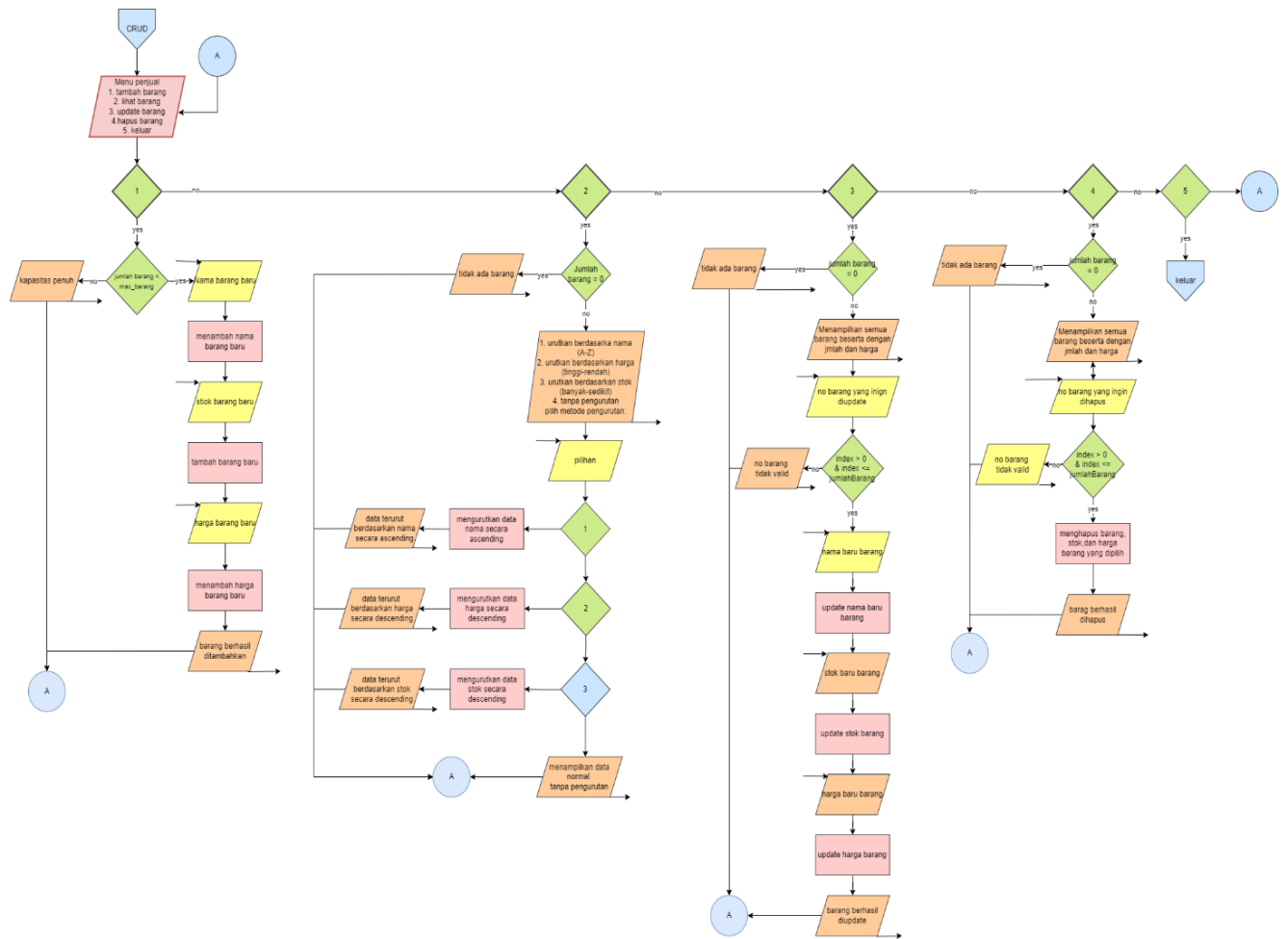
1. Flowchart



Gambar 1.1 Flowchart Menu Login dan Registrasi



Gambar 1.2 Flowchart Menu Pembeli



Gambar 1.3 Flowchart Menu CRUD Admin

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. Fungsi 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;
    }
}

```



```

        cout << "Masukkan Stok Barang: ";
        cin >> stok;
        barang[jumlahBarang].detail.stok = stok;

        cout << "Masukkan Harga Barang: ";
        cin >> harga;
        barang[jumlahBarang].detail.harga = harga;

        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 nama barang sembako beserta stok dan harganya berdasarkan urutan yang kita pilih. Terdapat 4 opsi pengurutan yang tersedia. Pertama, mengurutkan data sembako berdasarkan nama secara ascending (A-Z) dengan menggunakan metode bubble sort. Kedua, mengurutkan data sembako berdasarkan harga secara descending (Tertinggi-Terendah) dengan menggunakan metode selection sort. Ketiga, mengurutkan data sembako berdasarkan stok secara descending (Terbanyak-Tersedikit)

menggunakan metode insertion sort. Dan yang keempat untuk menampilkan data barang sembako tanpa pengurutan.

Source Code:

```
void SortNamabubble(Barang arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j].nama > arr[j + 1].nama) {
                Barang temp = arr[j];
                arr[j] = arr[j + 1];
                arr[j + 1] = temp;
            }
        }
    }
}

void SortHargaSelection(Barang arr[], int n) {
    for (int i = 0; i < n - 1; i++) {
        int max_idx = i;
        for (int j = i + 1; j < n; j++)
        {
            if (arr[j].detail.harga > arr[max_idx].detail.harga)
            {
                max_idx = j;
            }
        }
        Barang temp = arr[max_idx];
        arr[max_idx] = arr[i];
        arr[i] = temp;
    }
}

void SortStokInsertion(Barang arr[], int n) {
    for (int i = 1; i < n; i++)
    {
        Barang key = arr[i];
        int j = i - 1;
        while (j >= 0 && arr[j].detail.stok < key.detail.stok)
        {
            arr[j + 1] = arr[j];
            j = j - 1;
        }
        arr[j + 1] = key;
    }
}
```

```

void lihatBarang() {
    if (jumlahBarang == 0) {
        cout << "Tidak ada barang yang tersedia.\n";
    } else {
        system("cls");
        Barang tempBarang[MAX_BARANG];
        for (int i = 0; i < jumlahBarang; i++) {
            tempBarang[i] = barang[i];
        }

        cout << "\n===== PENGURUTAN
=====\\n";
        cout << "Pilih metode pengurutan :\\n";
        cout << "-----\\n";
        cout << " 1. Urutkan berdasarkan Nama (A-Z)\\n";
        cout << " 2. Urutkan berdasarkan Harga (Tertinggi-Terendah)\\n";
        cout << " 3. Urutkan berdasarkan Stok (Terbanyak-Tersedikit)\\n";
        cout << " 4. Tanpa pengurutan\\n";
        cout << "-----\\n";
        cout << "Pilihan: ";

        int sortpilihan;
        cin >> sortpilihan;

        switch(sortpilihan) {
            case 1:
                system("cls");
                SortNamabubble(tempBarang, jumlahBarang);
                cout << "Diurutkan berdasarkan Nama (A-Z)\\n";
                break;
            case 2:
                system("cls");
                SortHargaSelection(tempBarang, jumlahBarang);
                cout << "Diurutkan berdasarkan Harga (Tertinggi-
Terendah)\\n";
                break;
            case 3:
                system("cls");
                SortStokInsertion(tempBarang, jumlahBarang);
                cout << "Diurutkan berdasarkan Stok (Terbanyak-
Tersedikit)\\n";
                break;
            case 4:
                system("cls");
                cout << "Menampilkan tanpa pengurutan\\n";

```

```

        break;
    default:
        cout << "Pilihan tidak valid, menampilkan tanpa sorting.\n";
    }

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

Source Code:

```

void updateBarang() {
    if (jumlahBarang > 0) {
        int index;

        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: ";
    cin >> index;
    if (index < 1 || index > jumlahBarang) {
        cout << "Nomor barang tidak valid. Kembali ke menu CRUD...\n";
        cout << "Klik Enter...";
        cin.ignore();
        cin.get();
        return;
    }

    cout << "Masukkan Nama Barang Baru: ";
    cin.ignore();
    getline(cin, barang[index - 1].nama);

    int stokBaru;
    cout << "Masukkan Stok Barang Baru: ";
    cin >> stokBaru;
    updateStokBarang(&barang[index - 1], stokBaru);

    int hargaBaru;
    cout << "Masukkan Harga Barang Baru: ";
    cin >> hargaBaru;
    updateHargaBarang(&barang[index - 1], hargaBaru);

    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.

Source Code:

```
void hapusBarang() {
    if (jumlahBarang > 0) {
        int index;

        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: ";
        cin >> index;
        if (index < 1 || index > jumlahBarang) {
            cout << "Nomor barang tidak valid. Kembali ke menu CRUD...\n";
            cout << "Klik Enter...";
            cin.ignore();
            cin.get();
            return;
        }
        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.

Source Code:

```
void beliBarang() {
    if (jumlahBarang > 0) {
        int index, jumlahBeli;

        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: ";
        cin >> index;
        if (index < 1 || index > jumlahBarang) {
            cout << "Nomor barang tidak valid. Kembali ke menu Pembeli...\n";
            cout << "Klik Enter...";
            cin.ignore();
            cin.get();
            return;
        }

        cout << "Masukkan jumlah yang ingin dibeli: ";
        cin >> jumlahBeli;

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


4. Uji Coba dan Hasil Output

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

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

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

Gambar 4.1 Login Berhasil Admin

```
=====
                        MENU MANAJEMEN BARANG SEMBAKO
=====
1. Tambah Barang
2. Lihat Barang
3. Update Barang
4. Hapus Barang
5. Keluar
=====
Pilih menu: 1
Masukkan Nama Barang: Masako
Masukkan Stok Barang: 25
Masukkan Harga Barang: 2500

=====
                        BARANG BERHASIL DITAMBAHKAN
=====
Barang dengan nama Masako
Jumlah barang: 25
Harga barang: 2500
=====
Klik Enter untuk kembali ke menu utama...
```

Gambar 4.2 Menambahkan Barang Sembako Baru

```

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

===== PENGURUTAN =====
Pilih metode pengurutan :
-----
1. Urutkan berdasarkan Nama (A-Z)
2. Urutkan berdasarkan Harga (Tertinggi-Terendah)
3. Urutkan berdasarkan Stok (Terbanyak-Tersedikit)
4. Tanpa pengurutan
-----
Pilihan: 1
Diurutkan berdasarkan Nama (A-Z)

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2         Garam           50       5000
3      Gula Pasir       50       2000
4        Masako        25       2500
5     Minyak Goreng    20      15000
6     Telur 1 piring    10      60000
=====
Tekan Enter untuk kembali ke menu...

```

Gambar 4.3 Lihat Daftar Barang Terurut Berdasarkan Nama Dari A-Z

```

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

===== PENGURUTAN =====
Pilih metode pengurutan :
-----
1. Urutkan berdasarkan Nama (A-Z)
2. Urutkan berdasarkan Harga (Tertinggi-Terendah)
3. Urutkan berdasarkan Stok (Terbanyak-Tersedikit)
4. Tanpa pengurutan
-----
Pilihan: 2
Diurutkan berdasarkan Harga (Tertinggi-Terendah)

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1     Telur 1 piring    10      60000
2     Minyak Goreng    20      15000
3         Garam           50       5000
4        Beras          100       5000
5        Masako        25       2500
6      Gula Pasir       50       2000
=====
Tekan Enter untuk kembali ke menu...

```

Gambar 4.4 Lihat Daftar Barang Terurut Berdasarkan Harga Tertinggi-Terendah

```

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

===== PENGURUTAN =====
Pilih metode pengurutan :
-----
1. Urutkan berdasarkan Nama (A-Z)
2. Urutkan berdasarkan Harga (Tertinggi-Terendah)
3. Urutkan berdasarkan Stok (Terbanyak-Tersedikit)
4. Tanpa pengurutan
-----
Pilihan: 3
Diurutkan berdasarkan Stok (Terbanyak-Tersedikit)

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2      Gula Pasir       50       2000
3          Garam       50       5000
4        Masako       25       2500
5    Minyak Goreng     20      15000
6    Telur 1 piring    10      60000
=====
Tekan Enter untuk kembali ke menu...

```

Gambar 4.5 Lihat Daftar Barang Berdasarkan Stok Terbanyak-Tersedikit

```

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

===== PENGURUTAN =====
Pilih metode pengurutan :
-----
1. Urutkan berdasarkan Nama (A-Z)
2. Urutkan berdasarkan Harga (Tertinggi-Terendah)
3. Urutkan berdasarkan Stok (Terbanyak-Tersedikit)
4. Tanpa pengurutan
-----
Pilihan: 4
Menampilkan tanpa pengurutan

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2      Gula Pasir       50       2000
3    Minyak Goreng     20      15000
4        Masako       25       2500
5          Garam       50       5000
6    Telur 1 piring    10      60000
=====
Tekan Enter untuk kembali ke menu...

```

Gambar 4.6 Lihat Daftar Barang Tanpa Pengurutan

```

=====
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        Masako          25        2500
5        Garam           50        5000
6        Telur 1 piring  10        60000
=====
Masukkan nomor barang yang ingin diupdate: 2
Masukkan Nama Barang Baru: Beras 25 kg
Masukkan Stok Barang Baru: 20
Masukkan Harga Barang Baru: 150000

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

```

```

=====
DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras          100       5000
2        Beras 25 kg    20        150000
3        Minyak Goreng  20        15000
4        Masako          25        2500
5        Garam           50        5000
6        Telur 1 piring  10        60000
=====
Tekan Enter untuk kembali ke menu...

```

Gambar 4.7 Update Data Barang Yang Sudah Ada

```

=====
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        Beras 25 kg    20        150000
3        Minyak Goreng  20        15000
4        Masako          25        2500
5        Garam           50        5000
6        Telur 1 piring  10        60000
=====
Masukkan nomor barang yang ingin dihapus: 1

=====
BARANG BERHASIL DIHAPUS
=====
Barang dengan nama Beras 25 kg
Jumlah barang : 20
Harga barang : 150000
=====
Klik Enter untuk kembali ke menu...

```

```
=====
                        DAFTAR BARANG SEMBAKO
=====
```

No	Nama Barang	Stok	Harga
1	Beras 25 kg	20	150000
2	Minyak Goreng	20	15000
3	Masako	25	2500
4	Garam	50	5000
5	Telur 1 piring	10	60000

```
=====
```

Gambar 4. 8 Menghapus Data Barang Sembako Dari Daftar

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

=====
                        REGISTRASI AKUN
=====
Username: Ridwan
Password: ridwan123

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

Gambar 4. 9 Registrasi Akun Pembeli

```
=====
                        Login Program
=====
Username: Ridwan
Password: ridwan123

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

=====
                        MENU PEMBELI
=====
1. Lihat Barang
2. Beli Barang
3. Keluar
=====
Pilih menu: |
```

Gambar 4. 10 Login Berhasil Pembeli

```
=====
                        MENU PEMBELI
=====
1. Lihat Barang
2. Beli Barang
3. Keluar
=====
Pilih menu: 2

=====
                        DAFTAR BARANG SEMBAKO
=====
No      Nama Barang      Stok      Harga
1        Beras 25 kg      20      150000
2        Minyak Goreng    20      15000
3          Masako         25       2500
4          Garam          50       5000
5      Telur 1 piring      10      60000
=====
Masukkan nomor barang yang ingin dibeli: 1
Masukkan jumlah yang ingin dibeli: 2

=====
                        PEMBELIAN BERHASIL
=====
Total harga: Rp 300000,-
Silahkan lakukan pembayaran...
=====
Klik Enter untuk kembali ke menu...|
```

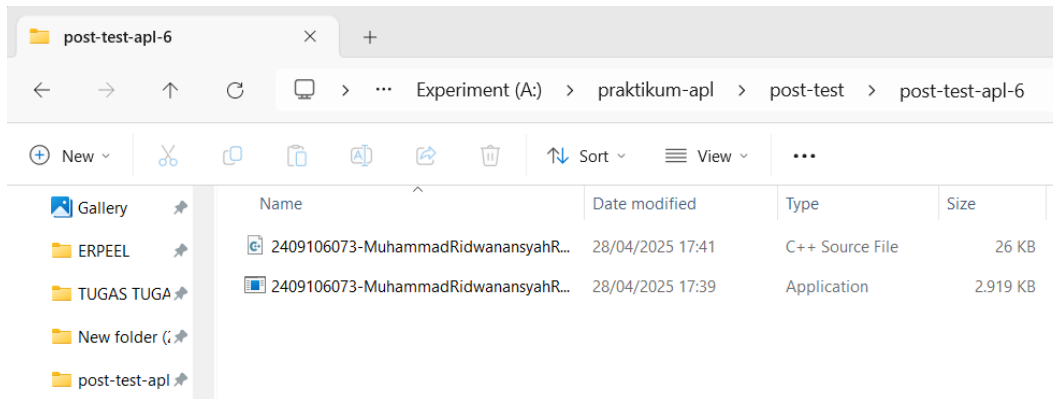
Gambar 4.11 Proses Pembelian

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

=====
                        ANDA TELAH KELUAR DARI PROGRAM
=====
PS A:\kerjakan posttest> |
```

Gambar 4.12 Keluar Program

5. Langkah-langkah Git



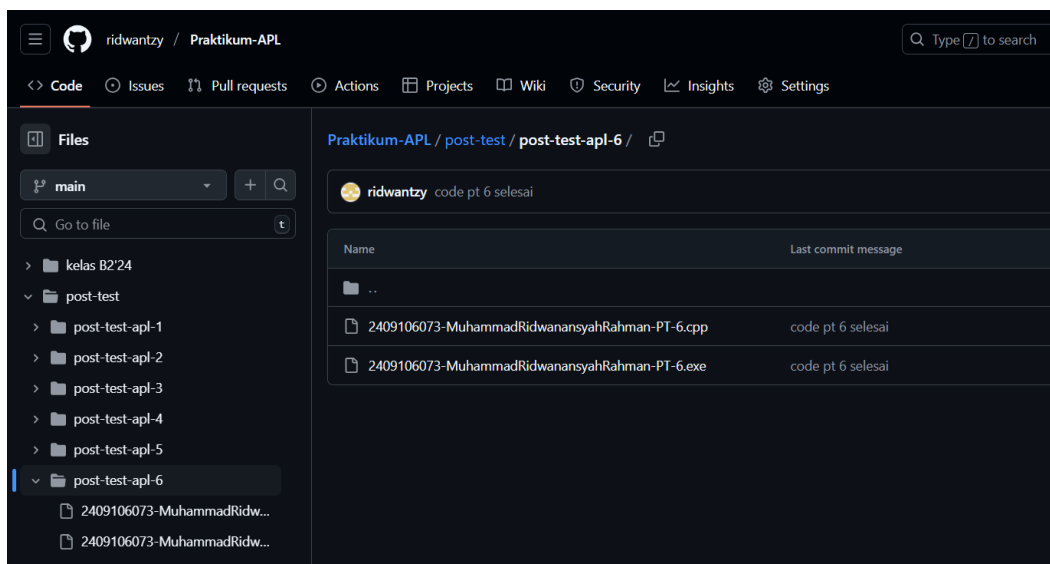
Gambar 5.1 Membuat folder `post-test-apl-6` yang berisi file `.cpp` dan `.exe` dalam file lokal `praktikum-apl`

```
PS A:\praktikum-apl> git add .
PS A:\praktikum-apl> git commit -m "code pt 6 selesai"
[main e9e8b56] code pt 6 selesai
2 files changed, 719 insertions(+)
create mode 100644 post-test/post-test-apl-6/2409106073-MuhammadRidwanansyahRahman-PT-6.cpp
create mode 100644 post-test/post-test-apl-6/2409106073-MuhammadRidwanansyahRahman-PT-6.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), 684.92 KiB | 4.15 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
   c483b4d..e9e8b56  main -> main
branch 'main' set up to track 'origin/main'.
PS A:\praktikum-apl> |
```

Gambar 5.3 Melakukan `Git Push`



Gambar 5.4 Posttest 6 berhasil ditambahkan