

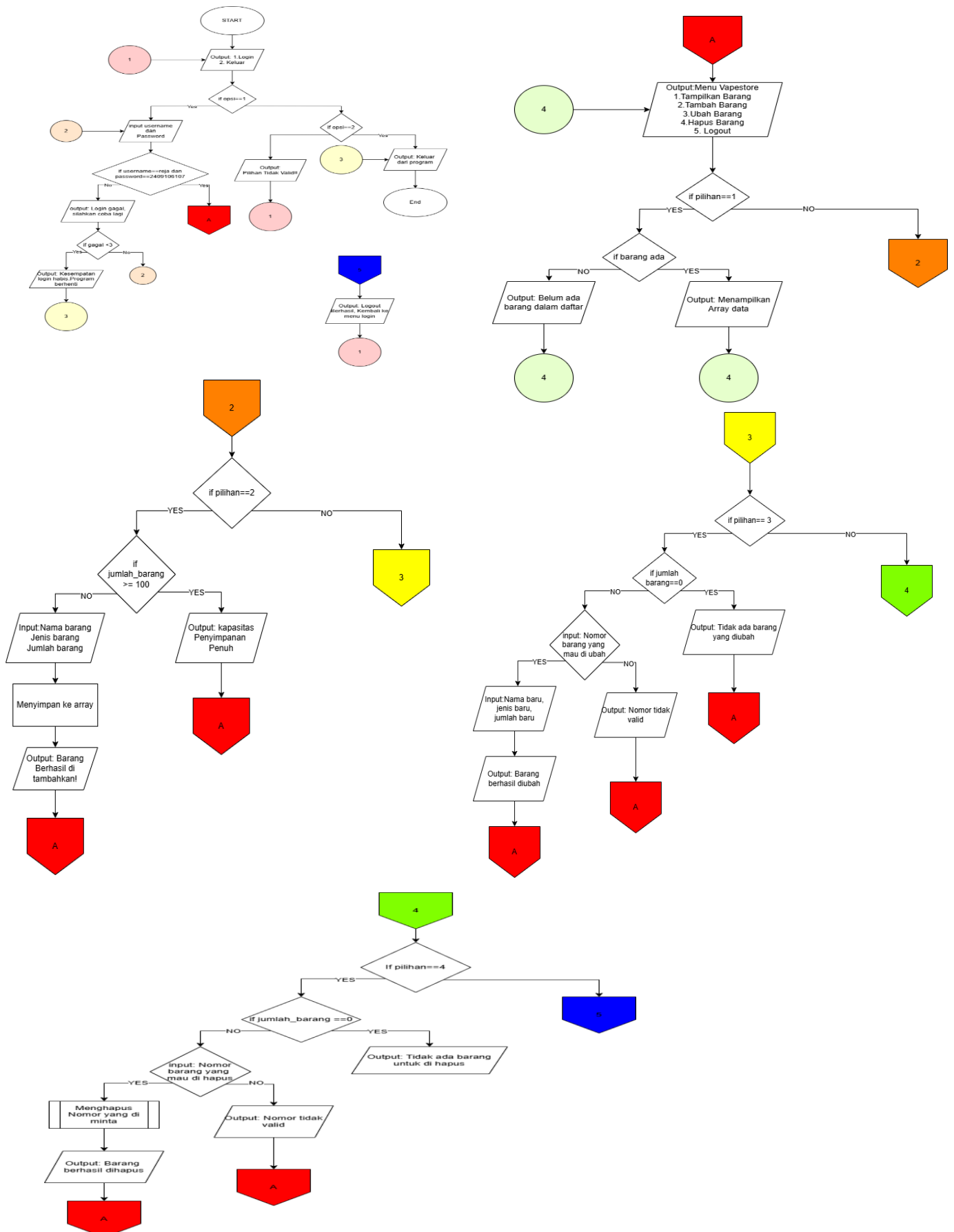
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Reza Alameka (2409106107)
Kelas (C2 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini dibuat untuk mensimulasikan sistem manajemen inventaris sederhana untuk toko vape. Pengguna dapat melakukan login, melihat daftar barang, menambah, mengubah, atau menghapus data barang yang tersedia.

2.2 Penjelasan Alur & Algoritma

- Menampilkan menu login:
 1. Login (memasukkan Nama & NIM).
 2. Keluar (menghentikan program).
- Jika login gagal 3 kali, program berhenti.
- Menu Utama Vape Store
 1. Tampilkan Barang → Menampilkan daftar barang.
 2. Tambah Barang → Menambahkan barang baru ke daftar.
 3. Ubah Barang → Mengedit informasi barang tertentu.
 4. Hapus Barang → Menghapus barang dari daftar.
 5. Logout → Kembali ke menu login.
- Fungsi Utama
 1. Tampilkan Barang: Menampilkan tabel daftar barang.
 2. Tambah Barang: Memasukkan nama, jenis, dan jumlah barang.
 3. Ubah Barang: Memilih barang untuk diubah, lalu memasukkan data baru.
 4. Hapus Barang: Memilih barang untuk dihapus, lalu menggeser daftar barang ke atas.
 5. Program Berjalan dalam Looping
- Setelah login, pengguna dapat terus menggunakan fitur sampai memilih logout atau keluar.

3. Source Code

A. Login & Logout

Pada fitur ini program pengguna memilih pilihan apakah lanjut ke tahap selanjutnya atau ingin memberhentikan program.

```
int main() {
    string nama, nim;
    int menuLogin;
    while (true) {
        do {
            cout << "\nMenu Login:\n";
            cout << "1. Login\n";
            cout << "2. Keluar\n";
            cout << "Pilihan: ";
            cin >> menuLogin;

            if (menuLogin == 2) {
                cout << "Keluar dari program.\n";
                return 0;
            } else if (menuLogin != 1) {
                cout << "Pilihan tidak valid!\n";
            }
        } while (menuLogin != 1);
    }
}
```

B. Input Username dan password

Pada fitur ini program akan mevalidasi apakah pengguna memasukan username dan password yang benar.

```
while (menuLogin != 1);

int attempts = 0;
while (attempts < 3) {
    cout << "Masukkan Nama: ";
    cin >> nama;
    cout << "Masukkan NIM: ";
    cin >> nim;

    if (nama == "Reja" && nim == "2409106107") {
        break;
    } else {
        cout << "Login gagal! Coba lagi.\n";
        attempts++;
    }
}
```

```

} else if (pilihan == 2) {
    if (jumlahBarang >= MAX_BARANG) {
        cout << "Kapasitas penyimpanan penuh!\n";
    } else {
        cout << "Masukkan nama barang: ";
        cin.ignore();
        getline(cin, namaBarang[jumlahBarang]);
        cout << "Masukkan jenis (Device/Liquid): ";
        getline(cin, jenisBarang[jumlahBarang]);
        cout << "Masukkan jumlah barang: ";
        cin >> jumlahBarangArr[jumlahBarang];
        jumlahBarang++;
        cout << "Barang berhasil ditambahkan!\n";
    }
} else if (pilihan == 3) {
    if (jumlahBarang == 0) {
        cout << "Tidak ada barang untuk diubah.\n";
    } else {
        int index;
        cout << "Pilih nomor barang yang ingin diubah: ";
        cin >> index;
        if (index < 1 || index > jumlahBarang) {
            cout << "Nomor tidak valid!\n";
        } else {
            cin.ignore();
            cout << "Masukkan nama baru: ";
            getline(cin, namaBarang[index - 1]);
            cout << "Masukkan jenis baru (Device/Liquid): ";
            getline(cin, jenisBarang[index - 1]);
            cout << "Masukkan jumlah baru: ";
            cin >> jumlahBarangArr[index - 1];
            cout << "Barang berhasil diubah!\n";
        }
    }
} else if (pilihan == 4) {
    if (jumlahBarang == 0) {
        cout << "Tidak ada barang untuk dihapus.\n";
    } else {
        int index;
        cout << "Masukkan nomor barang yang akan dihapus: ";
        cin >> index;
        if (index < 1 || index > jumlahBarang) {
            cout << "Nomor tidak valid!\n";
        } else {
            for (int i = index - 1; i < jumlahBarang - 1; i++) {
                namaBarang[i] = namaBarang[i + 1];
                jenisBarang[i] = jenisBarang[i + 1];
                jumlahBarangArr[i] = jumlahBarangArr[i + 1];
            }
        }
    }
}

```

```

        }
        jumlahBarang--;
        cout << "Barang berhasil dihapus!\n";
    }
}
} else if (pilihan == 5) {
    cout << "Logout berhasil. Kembali ke menu login.\n";
    break;
} else {
    cout << "Pilihan tidak valid!\n";
}
} while (true);
}
return 0;
}

```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Pada awal menjalankan program Pengguna akan di arahkan pada menu login dan keluar jika memilih keluar maka program akan berhenti. jika memilih login pennguna akan di arahkan untuk memasukkan username dan password. jika username dan password benar maka pengguna akan di arahkan pada menu utama yang berisi opsi 1 hingg 5 yang mana pengguna dapat melakukan CRUD. jika memilih opsi 1 program akan mengeluarkan output tampilan array. jika memilih opsi 2 pengguna dapat menambahkan barang yang ingin di masukan ke dalam data array. jika memilih opsi 3 pengguna dapat mengubah nama,jenis,dan jumlah barang pada array.jika memilih opsi 4 maka pengguna akan di suruh memilih nomor barang yang ingin di hapus pada array. jika memilih opsi 5 pengguna akan di arahkan ke menu awal .

4.2 Hasil Output

1. Pengguna akan masuk dalam program menu login dan di intruksi kan memilih opsi 1 dan 2. jika memilih opsi 1 maka pengguna akan masuk ke program selanjutnya dan jika memilih opsi 2 maka pengguna akan memberhentikan program.

```
Menu Login:  
1. Login  
2. Keluar  
Pilihan: 
```

Gambar 4.2.1 Menu awal

2. Jika memilih 1 program akan mengarahkan pengguna untuk memasukkan username dan password yang sudah di tetapkan oleh program yaitu dengan username = Reja dan password = 2409106107 jika benar akan program akan menampilkan menu utama.

```
Masukkan Nama: Reja  
Masukkan NIM: 2409106107  
  
Menu Vape Store:  
1. Tampilkan Barang  
2. Tambah Barang  
3. Ubah Barang  
4. Hapus Barang  
5. Logout  
Pilihan: 
```

Gambar 4.2.2 Menu Utama

3. Jika username != "Reja" dan password != "2409106107" maka program akan meminta inputan username dan password sebanyak 3 kali jika salah lebih dari 3 kali maka program akan berhenti sendirinya.

```
Masukkan Nama: Alameka  
Masukkan NIM: 6969  
Login gagal! Coba lagi.  
Masukkan Nama: R  
Masukkan NIM: 1  
Login gagal! Coba lagi.  
Masukkan Nama: e  
Masukkan NIM: 2  
Login gagal! Coba lagi.  
Kesempatan login habis. Program berhenti.
```

Gambar 4.2.3 Jika salah Username dan password
sebanyak 3 kali

4. Jika pengguna memilih opsi 1 maka program akan menampilkan data array yang ada.

```
Pilihan: 1
```

No	Nama Barang	Jenis	Jumlah
1	Oxva xlim G0	Device	10
2	Voopoo Drag X	Device	7
3	TRML T99	Device	5
4	Makna V2 3mg	Liquid	15
5	Bolu Lapis Talas V1 6mg	Liquid	12
6	The Orama V1 3mg	Liquid	20

Gambar 4.2.4 Output Opsi 1

5. Jika pengguna memilih opsi 2 maka akan di arahkan untuk mengisi nama,jenis,dan jenis baru yang ingin di tambahkan pada array.

```
Pilihan: 2
Masukkan nama barang: Thelema nano
Masukkan jenis (Device/Liquid): Device
Masukkan jumlah barang: 15
Barang berhasil ditambahkan!
```

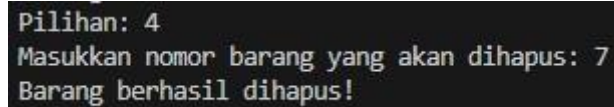
Gambar 4.2.5 Output Opsi 2

6. Jika pengguna memilih opsi 3 maka akan di suruh memilih nomor barang yang mana ingin di ubah dan setelah itu di suruh menginput nama,jeni,dan jumlah yang ingin di ubah.

```
Pilihan: 3
Pilih nomor barang yang ingin diubah: 7
Masukkan nama baru: Thelema solo
Masukkan jenis baru (Device/Liquid): Device
Masukkan jumlah baru: 10
Barang berhasil diubah!
```

Gambar 4.2.6 Output Opsi 3

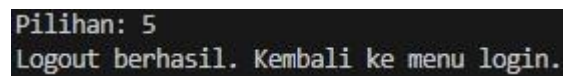
7. Jika pengguna memilih opsi 4 maka pengguna akan di suruh memasukkan inputan angka pada nomor barang yng ingin di hapus.



```
Pilihan: 4
Masukkan nomor barang yang akan dihapus: 7
Barang berhasil dihapus!
```

Gambar 4.2.7 Output Opsi 4

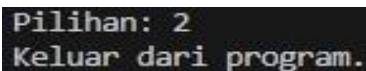
8. Jika pengguna memilih opsi 5 maka akan logout dan mengeluarkan output “Logout berhasil. Kembali kemenu Login” setelah itu program akan kembali ke menu login pada awal program.



```
Pilihan: 5
Logout berhasil. Kembali ke menu login.
```

Gambar 4.2.8 Output Opsi 5

9. Setelah itu Pengguna dapat memberhentikan program dengan memilih opsi 2 pada meu awal login.



```
Pilihan: 2
Keluar dari program.
```

Gambar 4.2.9 Output opsi 2 menu awal

6. Git

```
MINGW64:/d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL
Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL
$ git config --global user.email "rezalameka19@gmail.com"

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL
$ git init
Initialized empty Git repository in D:/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL/.git/

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (master)
$ git add .

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (master)
$ git branch -M main

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (main)
$ git remote add origin https://github.com/rezalameka/Praktikum-APL.git

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (main)
$ git commit -m "rejaaja"
[main (root-commit) 3c0dc41] rejaaja
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Posttest/Posttest_1/2409106107-RezaAlameka-Pt-1.cpp

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (main)
$ git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (5/5), 322 bytes | 322.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/rezalameka/Praktikum-APL.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

Acer@LAPTOP-HD3ENN1L MINGW64 /d/KULIAH/SEM 2/PRAKTIKUM APL/Praktikum-APL (main)
$ |
```

1. Inisialisasi Git (git init): Membuat repository Git lokal.
2. Konfigurasi Git (git config): Menetapkan email pengguna.
3. Menambahkan file (git add .): Menambahkan semua file ke staging area.
4. Membuat branch utama (git branch -M main): Mengubah nama branch ke main.
5. Menambahkan remote repository (git remote add origin <URL>): Menghubungkan repository lokal ke GitHub.
6. Commit perubahan (git commit -m "rejaaja"): Menyimpan perubahan ke repository lokal.
7. Push ke GitHub (git push -u origin main): Mengunggah perubahan ke repository di GitHub.

