

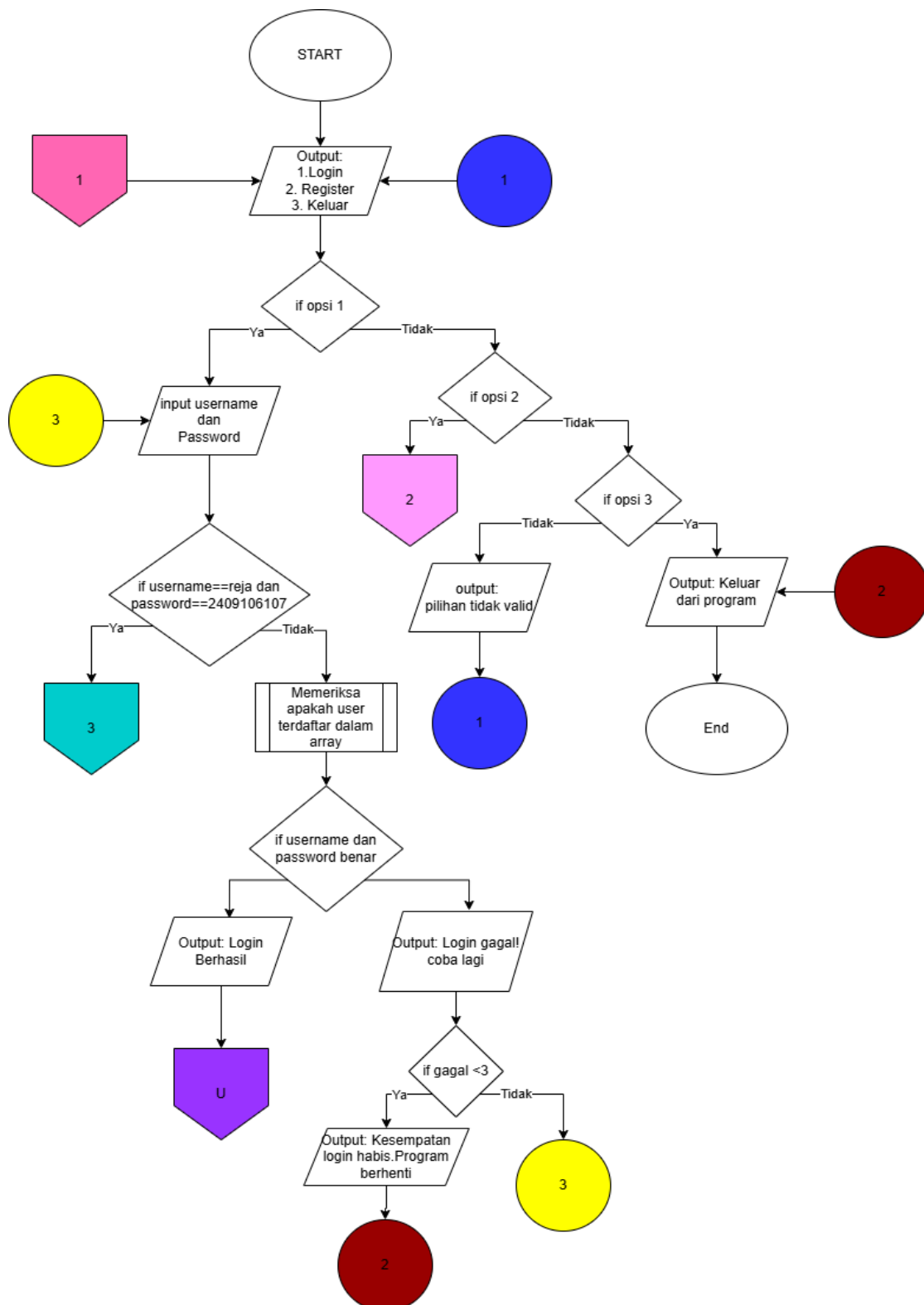
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT

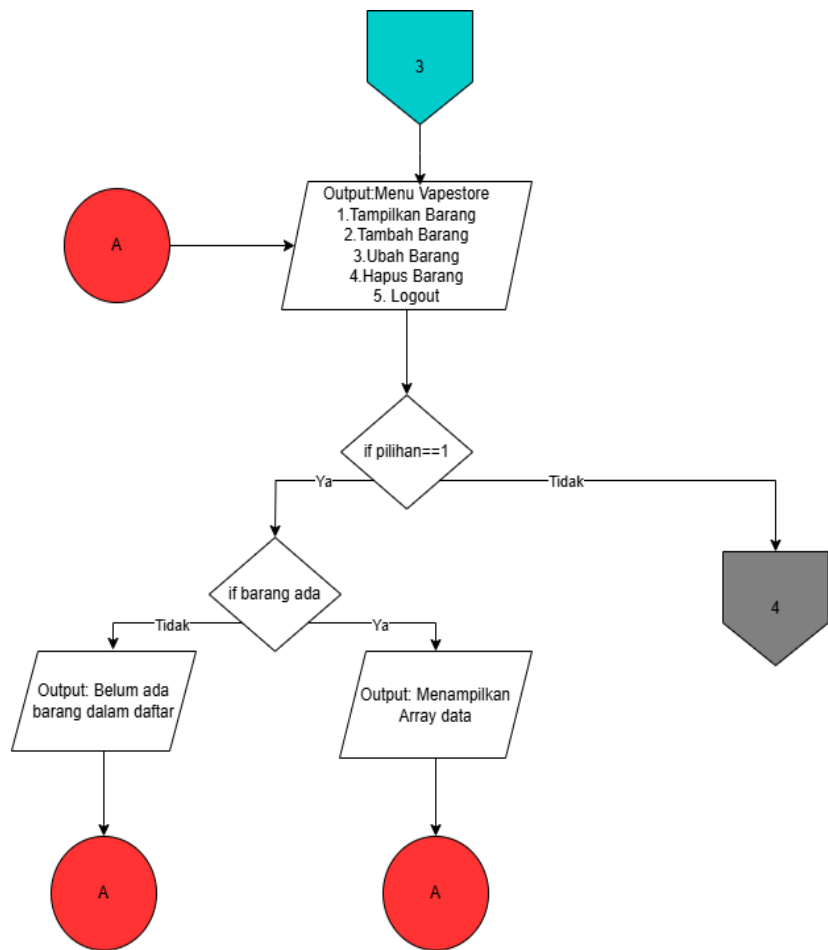
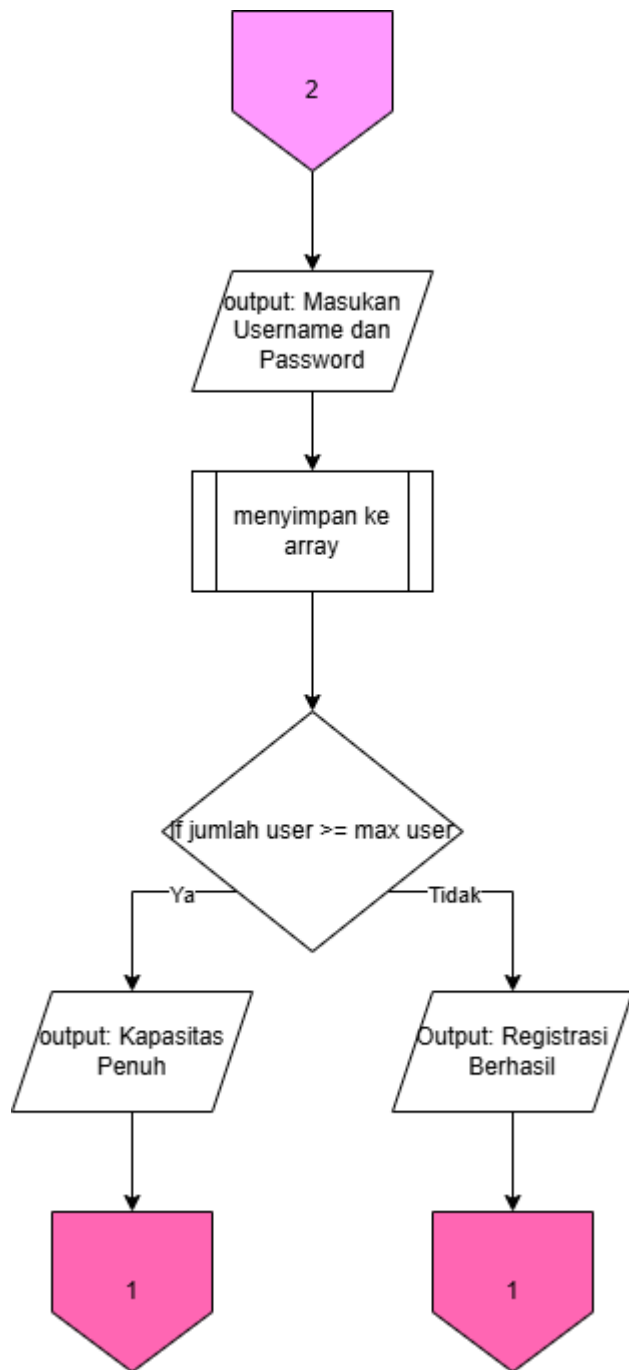


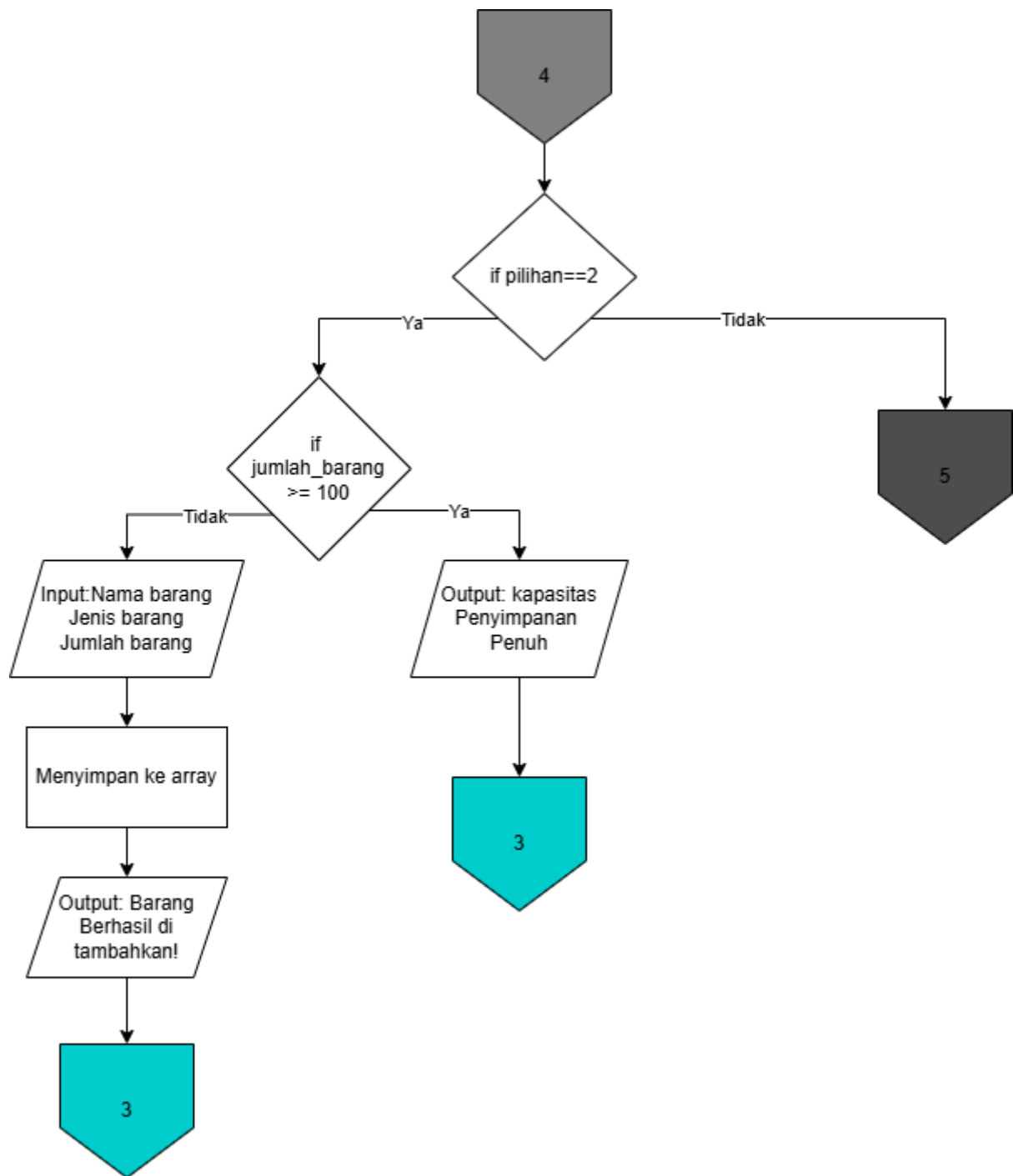
Disusun oleh:
Reza Alameka (2409106107)
Kelas (C2 '24)

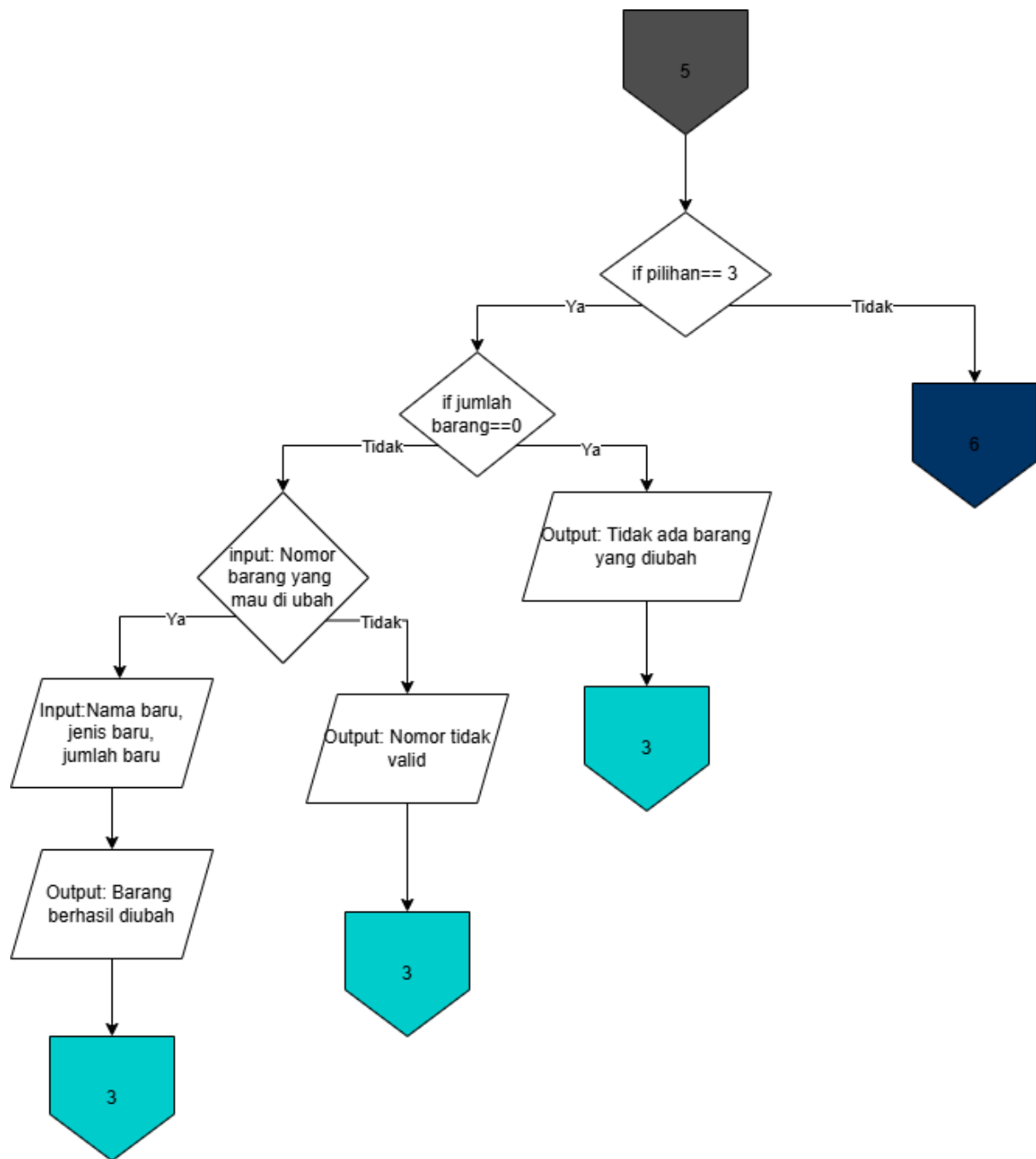
PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

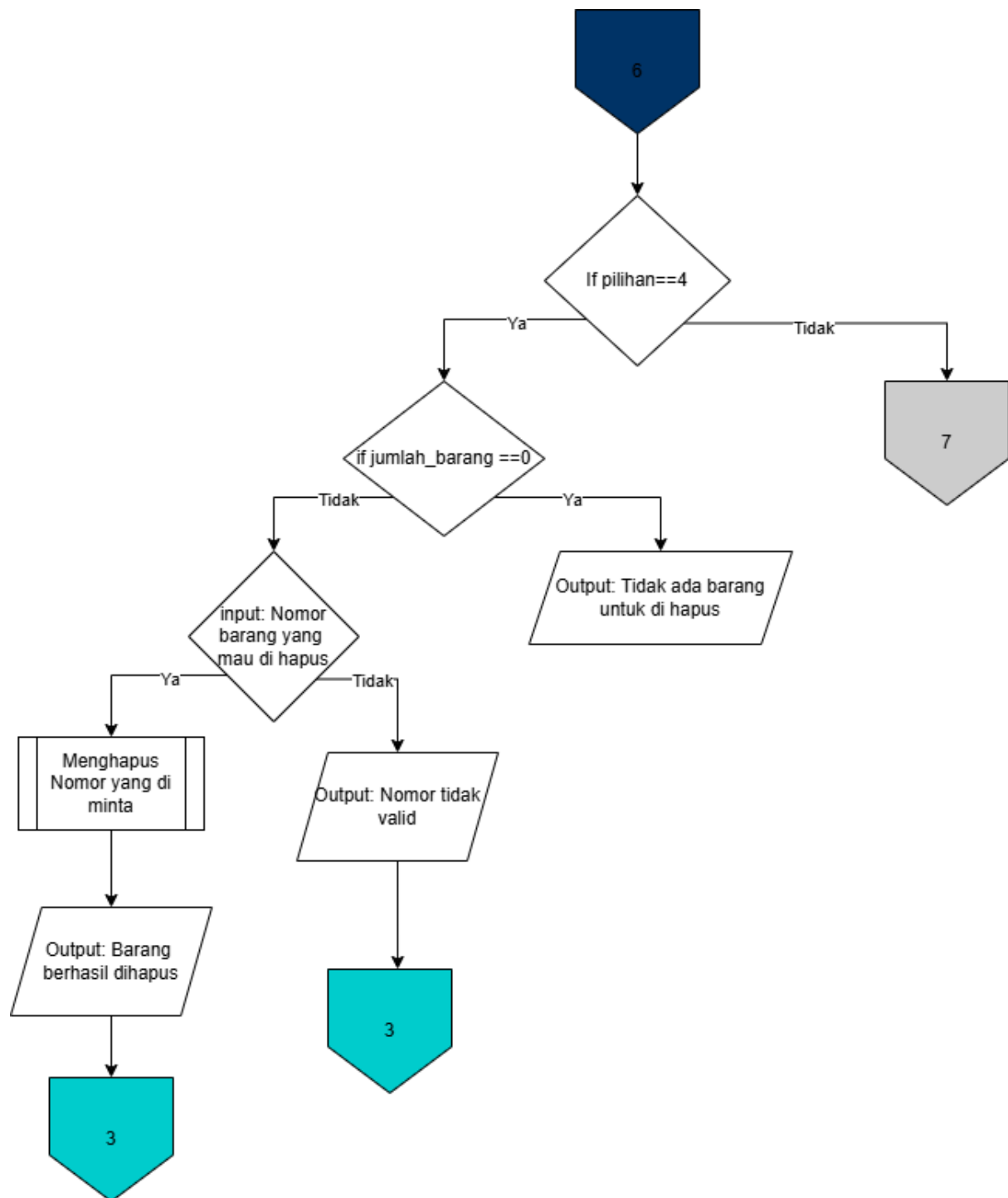
1. Flowchart

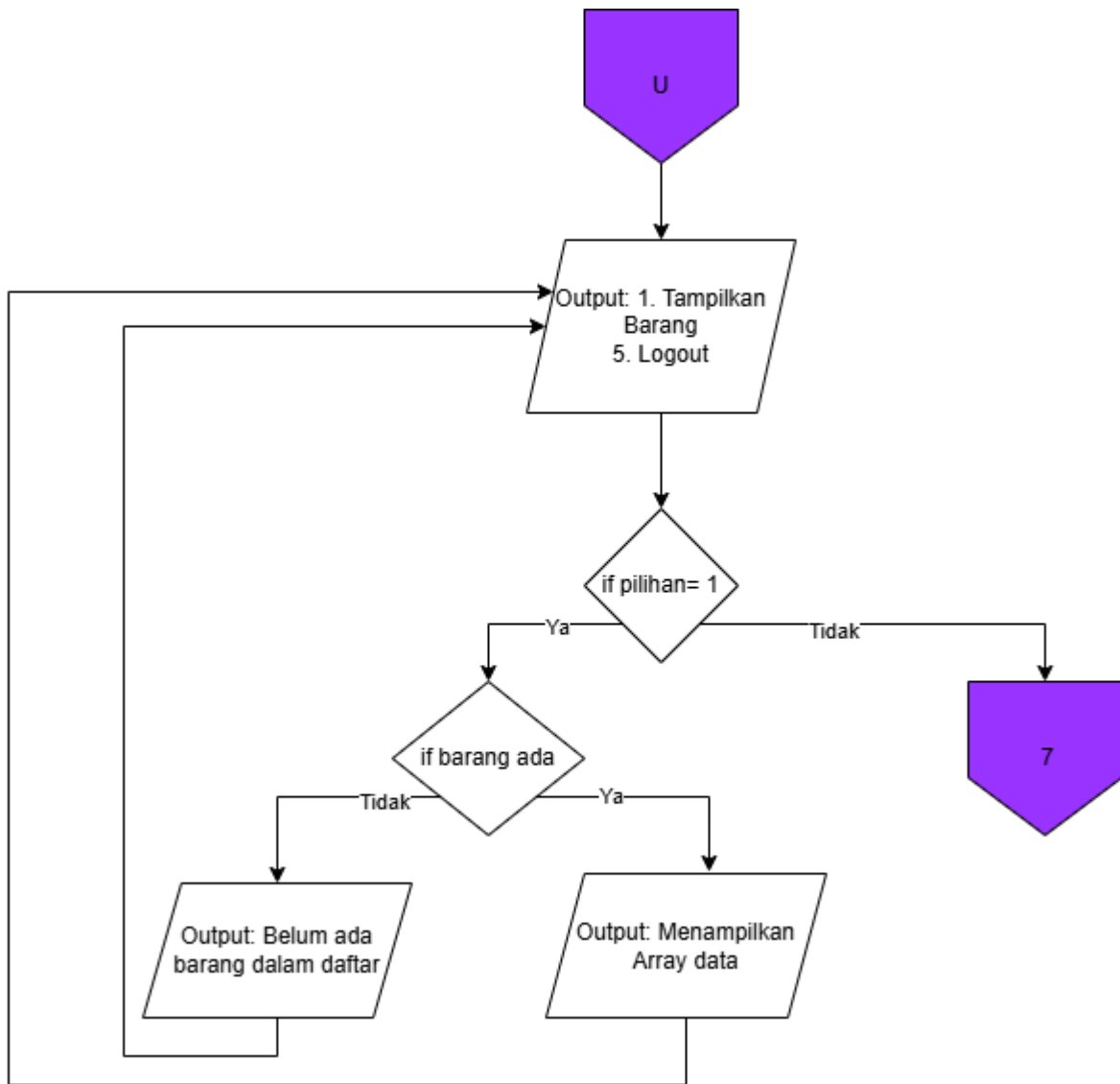


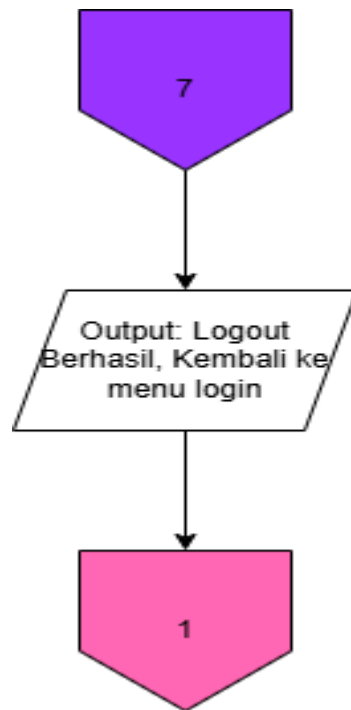












2. Analisis Program

2.2 Tujuan Dan Fungsi Program

1. Login & Registrasi:
 - Pengguna bisa mendaftar dan login sebelum mengakses sistem.
 - Admin memiliki akses khusus untuk mengelola barang.
2. Fitur Admin :
 - Menampilkan barang: Melihat daftar barang beserta jenis dan jumlah stok.
 - Menambah barang: Menambahkan produk baru ke dalam daftar.
 - Mengubah barang: Memperbarui informasi produk yang sudah ada.
 - Menghapus barang: Menghapus produk yang tidak diperlukan dari daftar.
3. Fitur User
 - Menampilkan barang: Melihat daftar barang beserta jenis dan jumlah stok.

2.3 Manfaat Utama

1. Mempermudah Manajemen Stok
 - Admin dapat menambah, mengubah, dan menghapus data barang dengan mudah.
 - Pengguna bisa melihat daftar barang yang tersedia.
2. Meningkatkan Keamanan dan Aksesibilitas
 - Sistem login memastikan hanya pengguna terdaftar yang bisa mengakses toko.
 - Admin memiliki kontrol eksklusif untuk mengelola barang.
 - Mengoptimalkan Pengelolaan Toko Vape
 - Menyediakan daftar barang yang rapi dan terstruktur.
 - Menghindari kesalahan pencatatan manual dalam stok barang.
3. Mempermudah Interaksi Pengguna
 - Antarmuka sederhana berbasis menu memudahkan pengguna dalam navigasi.
 - Registrasi memungkinkan lebih banyak pengguna untuk mengakses sistem.

3. Source Code

```
#include <iostream>
#include <iomanip>
#include <string>

using namespace std;

const int MAX_BARANG = 100;
const int MAX_USER = 100;

struct Barang {
    string nama;
    string jenis;
    int jumlah;
};

struct User {
    string username;
    string password;
};

struct VapeStore {
    Barang barang[MAX_BARANG];
    User users[MAX_USER];
    int jumlahBarang = 6; // Jumlah data awal
    int jumlahUser = 0; // Jumlah user terdaftar
};

int main() {
    VapeStore store = {
        {{"Oxva xlim G0", "Device", 10}, {"Vopoo Drag X", "Device", 7},
        {"TRML T99", "Device", 5}, {"Makna V2 3mg", "Liquid", 15}, {"Bolu Lapis
        Talas V1 6mg", "Liquid", 12}, {"The Orama V1 3mg", "Liquid", 20}},
        {},
        6,
        0
    };

    while (true) {
        int menuLogin;
        cout << "\nMenu Login:\n";
        cout << "1. Login\n";
        cout << "2. Register\n";
        cout << "3. Keluar\n";
        cout << "Pilihan: ";
        cin >> menuLogin;
```

```

if (menuLogin == 3) {
    cout << "Keluar dari program.\n";
    break;
}

if (menuLogin == 2) {
    // Registrasi user baru
    if (store.jumlahUser >= MAX_USER) {
        cout << "Kapasitas user penuh!\n";
        continue;
    }

    cout << "Masukkan Nama: ";
    cin >> store.users[store.jumlahUser].username;
    cout << "Masukkan Password: ";
    cin >> store.users[store.jumlahUser].password;
    store.jumlahUser++;
    cout << "Registrasi berhasil!\n";
    continue;
}

if (menuLogin != 1) {
    cout << "Pilihan tidak valid!\n";
    continue;
}

// Proses Login
string username, password;
int attempts = 0;
bool loginBerhasil = false;
bool Admin = false;

while (attempts < 3) {
    cout << "Masukkan Username: ";
    cin >> username;
    cout << "Masukkan Password: ";
    cin >> password;

    // Cek apakah admin
    if (username == "reja" && password == "2409106107") {
        loginBerhasil = true;
        Admin = true;
        break;
    }

    // Cek apakah user terdaftar
    for (int i = 0; i < store.jumlahUser; i++) {
        if (store.users[i].username == username &&

```



```
cout << "+-----+-----+-----+-----+";  
----+\n";  
}  
} else if (pilihan == 2 && Admin) {  
    if (store.jumlahBarang >= MAX_BARANG) {  
        cout << "Kapasitas penyimpanan penuh!\n";  
    } else {  
        cout << "Masukkan nama barang: ";  
        cin.ignore();  
        getline(cin, store.barang[store.jumlahBarang].nama);  
        cout << "Masukkan jenis (Device/Liquid): ";  
        getline(cin, store.barang[store.jumlahBarang].jenis);  
        cout << "Masukkan jumlah barang: ";  
        cin >> store.barang[store.jumlahBarang].jumlah;  
        store.jumlahBarang++;  
        cout << "Barang berhasil ditambahkan!\n";  
    }  
} else if (pilihan == 3 && Admin) {  
    if (store.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 > store.jumlahBarang) {  
            cout << "Nomor tidak valid!\n";  
        } else {  
            cin.ignore();  
            cout << "Masukkan nama baru: ";  
            getline(cin, store.barang[index - 1].nama);  
            cout << "Masukkan jenis baru (Device/Liquid): ";  
            getline(cin, store.barang[index - 1].jenis);  
            cout << "Masukkan jumlah baru: ";  
            cin >> store.barang[index - 1].jumlah;  
            cout << "Barang berhasil diubah!\n";  
        }  
    }  
} else if (pilihan == 4 && Admin) {  
    if (store.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 > store.jumlahBarang) {  
            cout << "Nomor tidak valid!\n";
```

```

        } else {
            for (int i = index - 1; i < store.jumlahBarang - 1;
i++) {
                store.barang[i] = store.barang[i + 1];
            }
            store.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";
}
}
}

return 0;
}

```

4. Uji Coba Dan hasil Output

```

Menu Login:
1. Login
2. Register
3. Keluar
Pilihan: 2
Masukkan Username: Alameka
Masukkan Password: 1
Registrasi berhasil!

```

Gambar 4.1 Output Register

```

Menu Vape Store:
1. Tampilkan Barang
5. Logout
Pilihan: 5
Logout berhasil. Kembali ke menu login.

```

Gambar 4.2 Menu User

```

Menu Login:
1. Login
2. Register
3. Keluar
Pilihan: 1
Masukkan Username: reja
Masukkan Password: 2409106107

Menu Vape Store:
1. Tampilkan Barang
2. Tambah Barang
3. Ubah Barang
4. Hapus Barang
5. Logout
Pilihan: 

```

Gambar 4.3 Login dan Menu Admin

Pilihan: 1

No	Nama Barang	Jenis	Jumlah
1	Oxva xlim GO	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 4.4 Output Pilihan 1 Admin

```

Pilihan: 2
Masukkan nama barang: Makna V3 9 mg
Masukkan jenis (Device/Liquid): Liquid
Masukkan jumlah barang: 99
Barang berhasil ditambahkan!

```

Gambar 4.4 Output Pilihan 2

```

Pilihan: 3
Pilih nomor barang yang ingin diubah: 1
Masukkan nama baru: Oxva Sq Pro
Masukkan jenis baru (Device/Liquid): Device
Masukkan jumlah baru: 59
Barang berhasil diubah!

```

Gambar 4.5 Output Pilihan 3

```
Pilihan: 4
Masukkan nomor barang yang akan dihapus: 5
Barang berhasil dihapus!
```

Gambar 4.6 Output Pilihan 4

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

Menu Login:
1. Login
2. Register
3. Keluar
Pilihan: 3
Keluar dari program.
```

Gambar 4.7 Ouput Pilihan 5

```
Pilihan: 1
```

No	Nama Barang	Jenis	Jumlah
1	Oxva xlim GO	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 4.8 Output Pilihan 6

5. Langkah-Langkah Git pada VSCode

```
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