

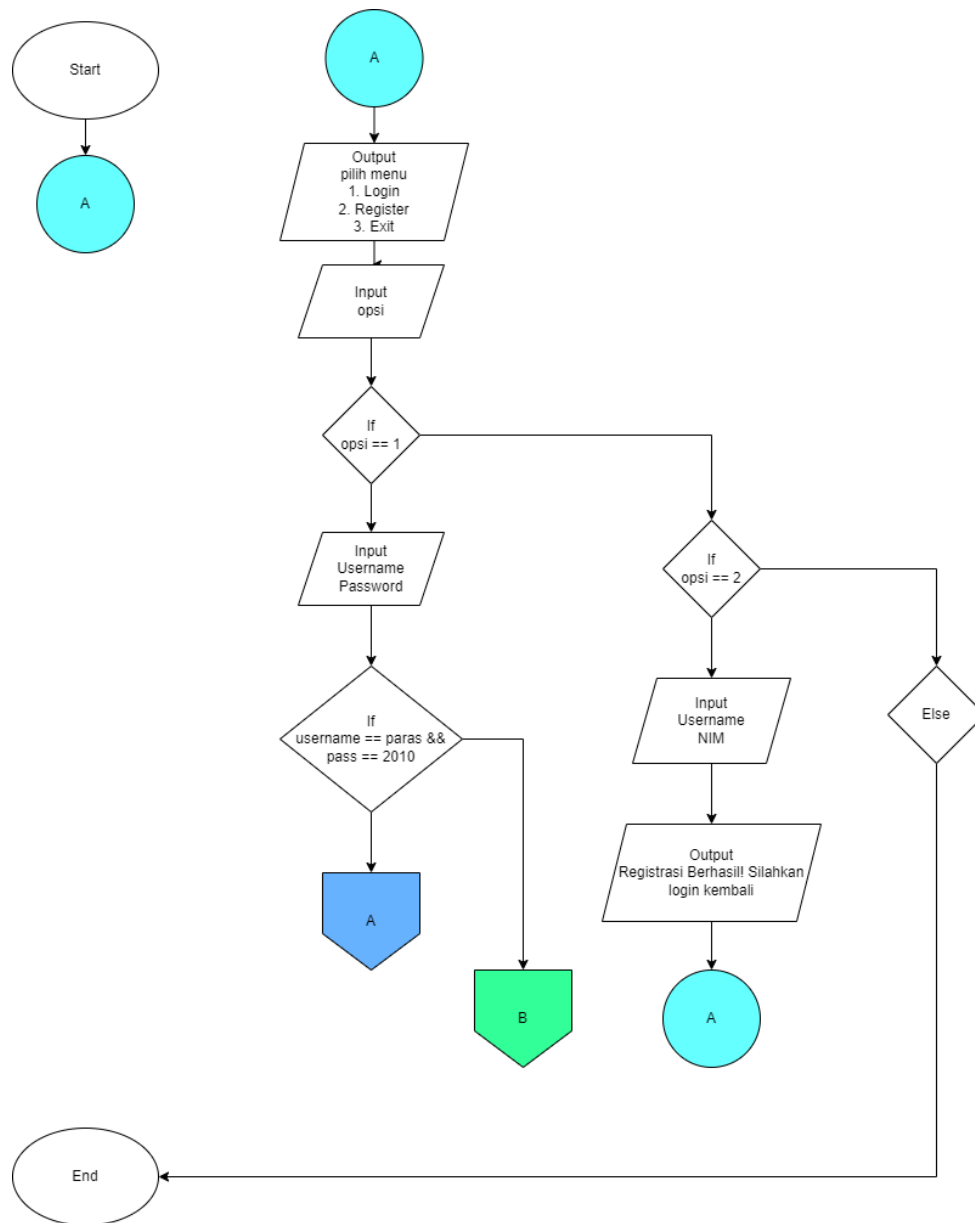
**LAPORAN PRAKTIKUM**  
**POSTTEST III**  
**ALGORITMA PEMROGRAMAN LANJUT**



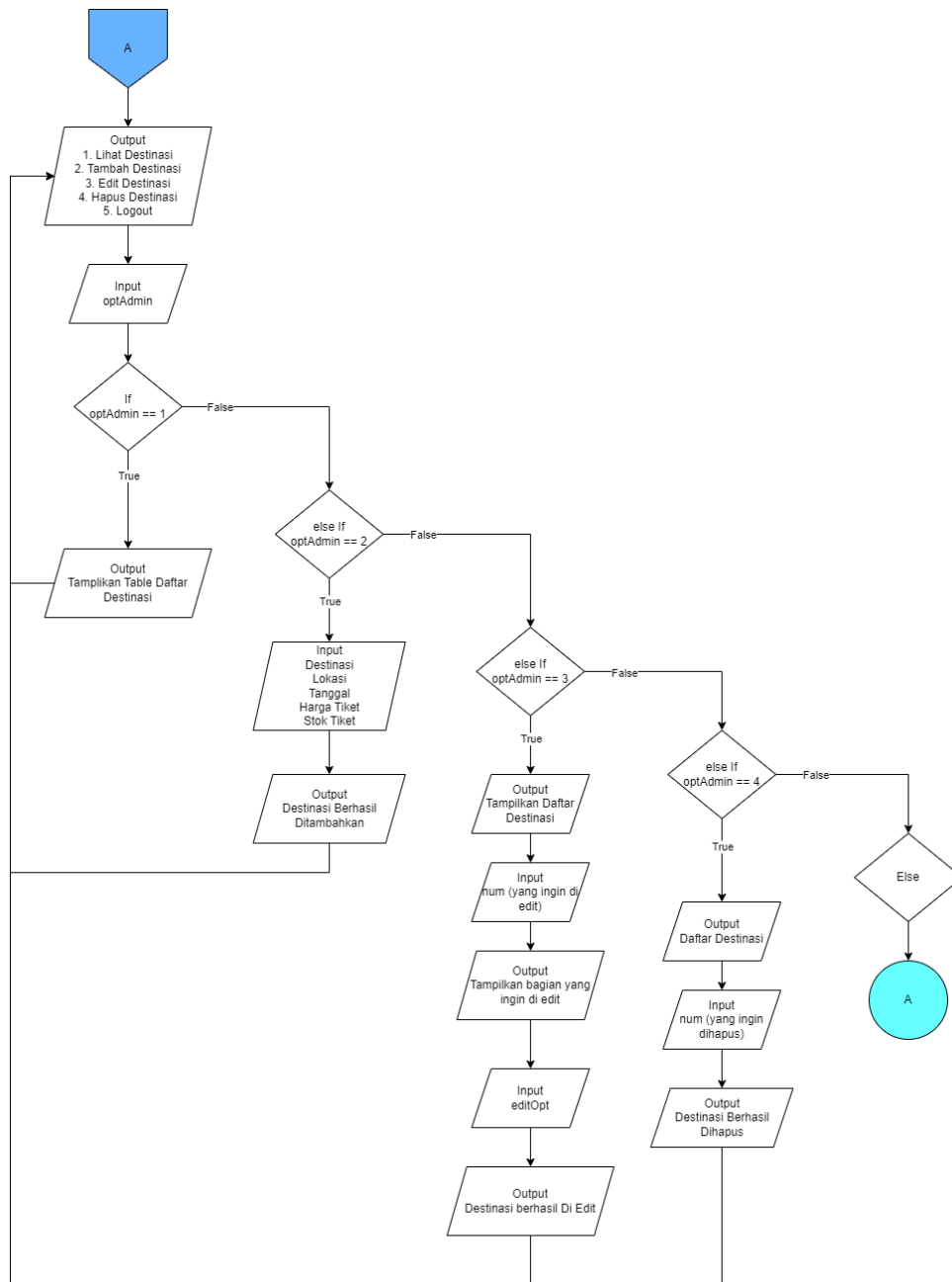
**Disusun oleh:**  
**Muhammad Farras Arhab Ince (2409106092)**  
**Kelas (C1 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

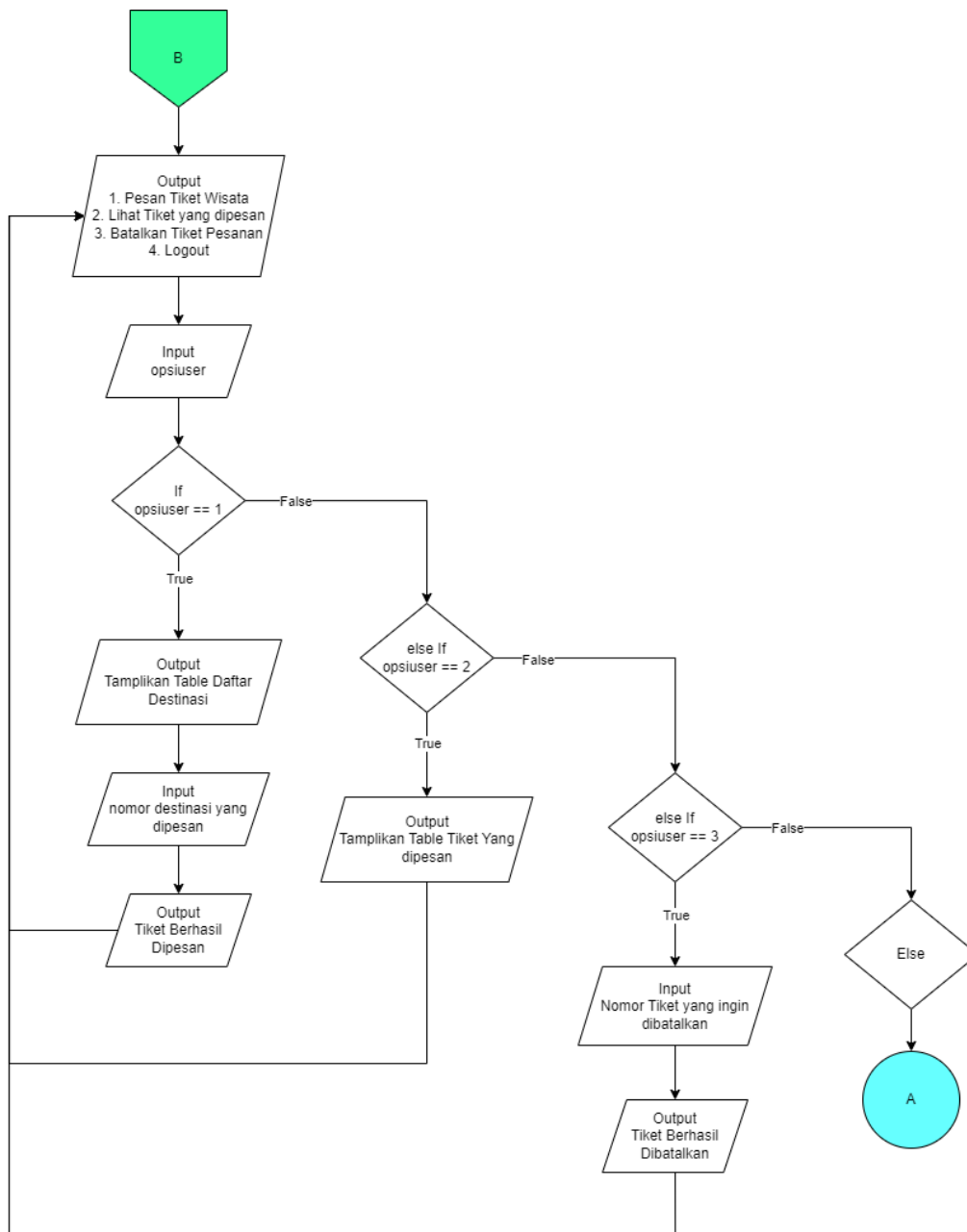
## 1. Flowchart



Gambar 1.1 Flowchart awal



Gambar 1.2 Flowchart Menu Admin



Gambar 1.3 Flowchart Menu Pengguna

## **2. Analisis Program**

Program ini dibuat untuk memudahkan proses pemesanan tiket wisata dengan menggunakan sistem multiuser. Artinya, ada dua jenis pengguna yaitu admin dan pengguna biasa. Program ini memiliki beberapa tujuan dan fungsi utama sebagai berikut:

### **1. Pembuatan Akun:**

Program menyediakan fitur register agar siapa saja bisa membuat akun baru. Saat pengguna mendaftar, data seperti nama dan NIM disimpan ke dalam array of struct bernama Akun. Akun yang dibuat melalui register secara otomatis dikategorikan sebagai pengguna biasa. Dengan demikian, setiap orang yang mendaftar tidak perlu melalui proses pembuatan akun secara terpisah karena data akun sudah tersimpan di program.

### **2. Sistem Login:**

Setelah mendaftar, pengguna harus melakukan login dengan memasukkan nama dan NIM. Proses login dilakukan dengan memeriksa data yang dimasukkan apakah cocok dengan data yang ada di array akun. Jika data yang dimasukkan sesuai, maka pengguna bisa mengakses sistem. Terdapat batas maksimum tiga kali percobaan login untuk menghindari kesalahan input yang terus-menerus.

### **3. Pembagian Hak Akses (Admin vs Pengguna):**

Program ini membedakan dua jenis pengguna.

#### **a. Admin:**

Admin adalah akun default yang sudah ada sejak awal (dalam kode, admin diwakili oleh akun pertama). Admin memiliki hak akses lebih untuk mengelola data destinasi wisata, misalnya bisa menambah, mengedit, dan menghapus data destinasi.

#### **b. Pengguna:**

Pengguna biasa yang mendaftar melalui register hanya dapat melakukan pemesanan tiket, melihat tiket yang sudah dipesan, dan membatalkan tiket jika diperlukan. Dengan demikian, sistem membatasi akses berdasarkan peran masing-masing.

#### **4. Pengelolaan Data Wisata:**

Data destinasi wisata disimpan dalam sebuah array dua dimensi yang berisi informasi seperti:

- a. Nama destinasi
- b. Lokasi
- c. Tanggal pelaksanaan wisata
- d. Harga tiket
- e. Stok tiket yang tersedia

Admin memiliki kemampuan untuk mengubah data ini melalui menu khusus, sehingga jika ada destinasi baru atau ada perubahan informasi, admin dapat meng-update data dengan mudah.

#### **5. Proses Pemesanan dan Pembatalan Tiket:**

Pengguna biasa dapat melihat daftar destinasi wisata yang tersedia dan memilih destinasi yang ingin dipesan tiketnya. Saat memesan, jumlah tiket yang diinginkan akan diperiksa apakah masih tersedia berdasarkan stok yang ada. Jika cukup, maka tiket berhasil dipesan dan stok tiket akan berkurang. Begitu juga jika pengguna membatalkan tiket, maka stok tiket akan dikembalikan sesuai dengan jumlah tiket yang dibatalkan.

Secara keseluruhan, program ini membantu mengatur dan memudahkan proses pemesanan tiket wisata dengan menggunakan sistem login dan register. Program ini juga memastikan bahwa hanya pengguna yang terverifikasi yang dapat mengakses menu sesuai dengan hak aksesnya, sehingga admin dapat mengelola data destinasi dengan lebih baik dan pengguna biasa dapat melakukan pemesanan tiket dengan mudah.

### 3. Source Code

#### A. Struktur Data

Menyimpan data akun, destinasi wisata, dan booking tiket menggunakan array of struct agar data lebih terstruktur.

```
#define MAX_WISATA 100
#define MAX_BOOKING 100
#define MAX_AKUN 100

typedef struct {
    string nama;
    string nim;
} Akun;

struct Destinasi {
    string nama;
    string lokasi;
    string tanggal;
    string harga;
    int stok;
};

struct Booking {
    string destinasi;
    string lokasi;
    string tanggal;
    string harga;
    int jumlah;
};

// Inisialisasi data awal
Destinasi wisata[MAX_WISATA] = {
    {"Pulau Derawan", "Berau", "10-April-2025", "1500000", 50},
    {"Bukit Bangkirai", "Kukar", "20-Mei-2025", "750000", 30},
    {"Kepulauan Maratua", "Berau", "15-Juni-2025", "2000000", 25},
    {"Pantai Melawai", "Balikpapan", "5-Juli-2025", "500000", 40},
    {"Danau Labuan Cermin", "Biduk-biduk", "1-Agustus-2025", "1000000", 20}
};
int wisataCount = 5;

Akun akun[MAX_AKUN] = {
    {"paras", "2010"}, // Admin default
    {"Muhammad Farras Arhab Ince", "2409106092"} // User default
};
int Jumlahakun = 2;

Booking bookings[MAX_BOOKING];
int bookingCount = 0;
```

## B. Sistem Login dan Register

User bisa login dengan batasan maksimal 3 kali percobaan. Program juga mendukung fitur register untuk menambahkan user baru ke dalam sistem.

```
// Login User
string user, pass;
int loginAttempt = 0;
bool success = false;
while (loginAttempt < 3 && !success) {
    cout << "\nNama: ";
    getline(cin, user);
    cout << "NIM: ";
    getline(cin, pass);
    for (int i = 0; i < Jumlahakun; i++) {
        if (user == akun[i].nama && pass == akun[i].nim) {
            success = true;
            break;
        }
    }
    if (!success) {
        cout << "Login Gagal. Coba Lagi\n";
        loginAttempt++;
    }
}
if (!success) {
    cout << "Gagal login 3 kali. Program berhenti.\n";
    return 0;
}

// Register
if (Jumlahakun < MAX_AKUN) {
    cout << "\n=== Register Akun Baru ===\n";
    cout << "Nama: "; getline(cin, akun[Jumlahakun].nama);
    cout << "NIM: "; getline(cin, akun[Jumlahakun].nim);
    Jumlahakun++;
    cout << "Register berhasil! Silakan login kembali.\n";
}
```

## C. Menu Admin

Admin dapat melakukan CRUD (Create, Read, Update, Delete) untuk data destinasi wisata. Tampilan data juga diformat menggunakan tabel rapi.

```
if (user == akun[0].nama && pass == akun[0].nim) { // Admin
    while (true) {
        cout << "\n=== Menu Admin ===\n";
        cout << "1. Lihat Destinasi\n2. Tambah Destinasi\n3. Edit Destinasi\n4.
```



```

Hapus Destinasi\n5. Logout\n";
    int optAdmin;
    cin >> optAdmin;
    cin.ignore();
    if (optAdmin == 1) {
        // Menampilkan daftar destinasi wisata dengan tabel rapi
    }
    else if (optAdmin == 2) {
        // Tambah destinasi baru ke array 'wisata'
    }
    else if (optAdmin == 3) {
        // Edit data destinasi tertentu
    }
    else if (optAdmin == 4) {
        // Hapus data destinasi dari array
    }
    else if (optAdmin == 5) {
        break; // Logout
    }
}
}

```

#### D. Menu Pengguna

User biasa bisa melakukan pemesanan tiket, melihat daftar tiket yang sudah dipesan, dan membatalkan tiket yang telah dibeli.

```

else { // User Biasa
    while (true) {
        cout << "\n=== Menu Pengguna ===\n";
        cout << "1. Pesan Tiket Wisata\n2. Lihat Tiket yang Dipesan\n3. Batalkan
Tiket Pesanan\n4. Logout\n";
        int opsiuser;
        cin >> opsiuser;
        cin.ignore();
        if (opsiuser == 1) {
            // Tampilkan daftar destinasi dan proses pemesanan tiket
        }
        else if (opsiuser == 2) {
            // Menampilkan tiket yang telah dipesan user
        }
        else if (opsiuser == 3) {
            // Batalkan pesanan tiket dan kembalikan stok
        }
        else if (opsiuser == 4) {
            break; // Logout
        }
    }
}
}

```

#### 4. Uji Coba dan Hasil Output

```
=====
                HALO SELAMAT DATANG!
                Pemesanan Tiket Wisata
=====
1. Login
2. Register
3. Exit
=====
Pilih opsi: 1

Nama: paras
NIM: 2010
```

Gambar 4.1 Output Menu Utama & Login

```
=====
                HALO SELAMAT DATANG!
                Pemesanan Tiket Wisata
=====
1. Login
2. Register
3. Exit
=====
Pilih opsi: 2

=== Register Akun Baru ===
Masukkan Nama: Ikhsan
Masukkan NIM: 2409106118
Register berhasil! Silakan login kembali.

=====
                HALO SELAMAT DATANG!
                Pemesanan Tiket Wisata
=====
1. Login
2. Register
3. Exit
=====
Pilih opsi: 1

Nama: ikhsan
NIM: 2409106118
Login Gagal. Coba Lagi

Nama: Ikhsan
NIM: 2409106118
```

Gambar 4.2 Output Registrasi

```

===== Menu Admin =====
1. Lihat Destinasi
2. Tambah Destinasi
3. Edit Destinasi
4. Hapus Destinasi
5. Logout
=====
Pilih opsi: 1

---- Daftar Destinasi Wisata ----
+---+-----+-----+-----+-----+-----+
| NO | Destinasi | Lokasi | Tanggal | Harga | Tiket Tersedia |
+---+-----+-----+-----+-----+-----+
| 1 | Pulau Derawan | Berau | 10-April-2025 | 1500000 | 50 |
| 2 | Bukit Bangkirai | Kukar | 20-Mei-2025 | 750000 | 30 |
| 3 | Kepulauan Maratua | Berau | 15-Juni-2025 | 2000000 | 25 |
| 4 | Pantai Melawai | Balikpapan | 5-Juli-2025 | 500000 | 40 |
| 5 | Danau Labuan Cermin | Biduk-biduk | 1-Agustus-2025 | 1000000 | 20 |
+---+-----+-----+-----+-----+-----+

```

Gambar 4.3 Output Menu Admin

```

===== Menu Pengguna =====
1. Pesan Tiket Wisata
2. Lihat Tiket yang Dipesan
3. Batalkan Tiket Pesanan
4. Logout
=====
Pilih opsi: 1

---- Daftar Destinasi Wisata ----
+---+-----+-----+-----+-----+-----+
| NO | Destinasi | Lokasi | Tanggal | Harga | Tiket Tersedia |
+---+-----+-----+-----+-----+-----+
| 1 | Pulau Derawan | Berau | 10-April-2025 | 1500000 | 50 |
| 2 | Bukit Bangkirai | Kukar | 20-Mei-2025 | 750000 | 30 |
| 3 | Kepulauan Maratua | Berau | 15-Juni-2025 | 2000000 | 25 |
| 4 | Pantai Melawai | Balikpapan | 5-Juli-2025 | 500000 | 40 |
| 5 | Danau Labuan Cermin | Biduk-biduk | 1-Agustus-2025 | 1000000 | 20 |
+---+-----+-----+-----+-----+-----+

Nomor destinasi yang ingin dipesan: 1
Jumlah tiket yang ingin dipesan: 2
Tiket berhasil dipesan!

```

Gambar 4.4 Output Menu Pengguna

## 5. Langkah-Langkah Git pada VSCode

```
farra@LAPTOP-SQ4287S4 MINGW64 /d/kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git add .

farra@LAPTOP-SQ4287S4 MINGW64 /d/kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git commit -m "Posttest 3"
[main be8fb1a] Posttest 3
1 file changed, 0 insertions(+), 0 deletions(-)
delete mode 100644 Post-Test/Post-Test-2/2409106092-MuhammadFarrasArhabInce-PT-
2.docx

farra@LAPTOP-SQ4287S4 MINGW64 /d/kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$ git push origin main
Enumerating objects: 19, done.
Counting objects: 100% (19/19), done.
Delta compression using up to 12 threads
Compressing objects: 100% (15/15), done.
Writing objects: 100% (15/15), 2.00 MiB | 896.00 KiB/s, done.
Total 15 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/parashrb/praktikum-apl.git
81a5b31..be8fb1a main -> main

farra@LAPTOP-SQ4287S4 MINGW64 /d/kuliah/SEMESTER 2/ALGORITMA PEMROGRAMAN LANJUT
(APL)/PRAKTIKUM APL/praktikum-apl (main)
$
```

Gambar 5 Terminal Git Bash

1. `git add .` digunakan untuk menambahkan semua perubahan yang telah dibuat ke dalam staging area agar siap dikomit.
2. Perintah `git commit -m` digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area ke dalam repository Git, sekaligus memberikan pesan singkat yang menjelaskan tujuan atau isi dari commit tersebut.
3. Perintah `git push origin main` digunakan untuk mengirim (push) commit yang ada di branch main pada repository lokal ke repository remote (seperti GitHub) agar perubahan dapat tersimpan dan diakses secara online.