LAPORAN PRAKTIKUM

POSTTEST 6

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Muhammad Naufal Adi Brata Putra Suharizman Poerwo (2409106049)

Kelas (B1 ‘24)

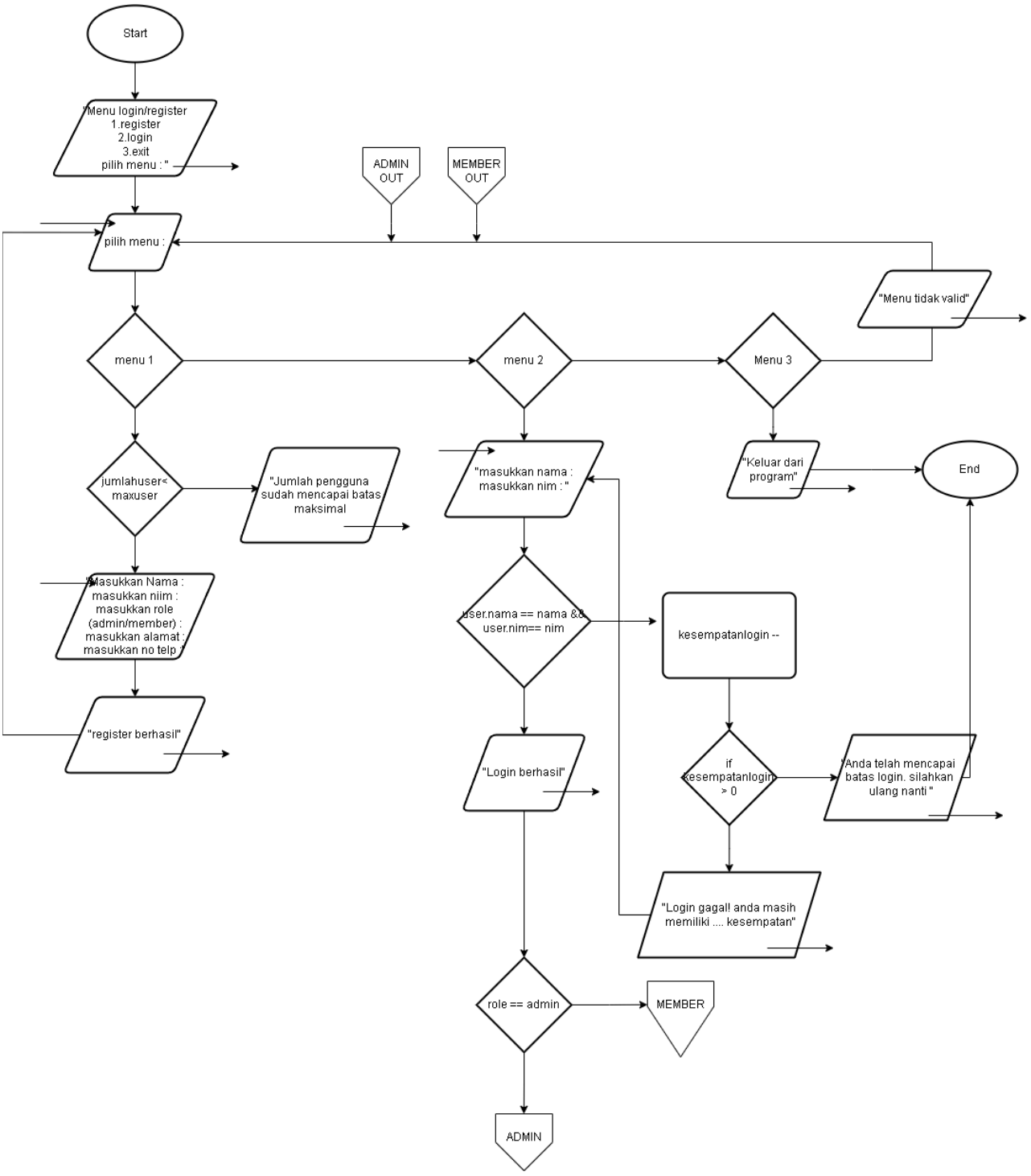
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

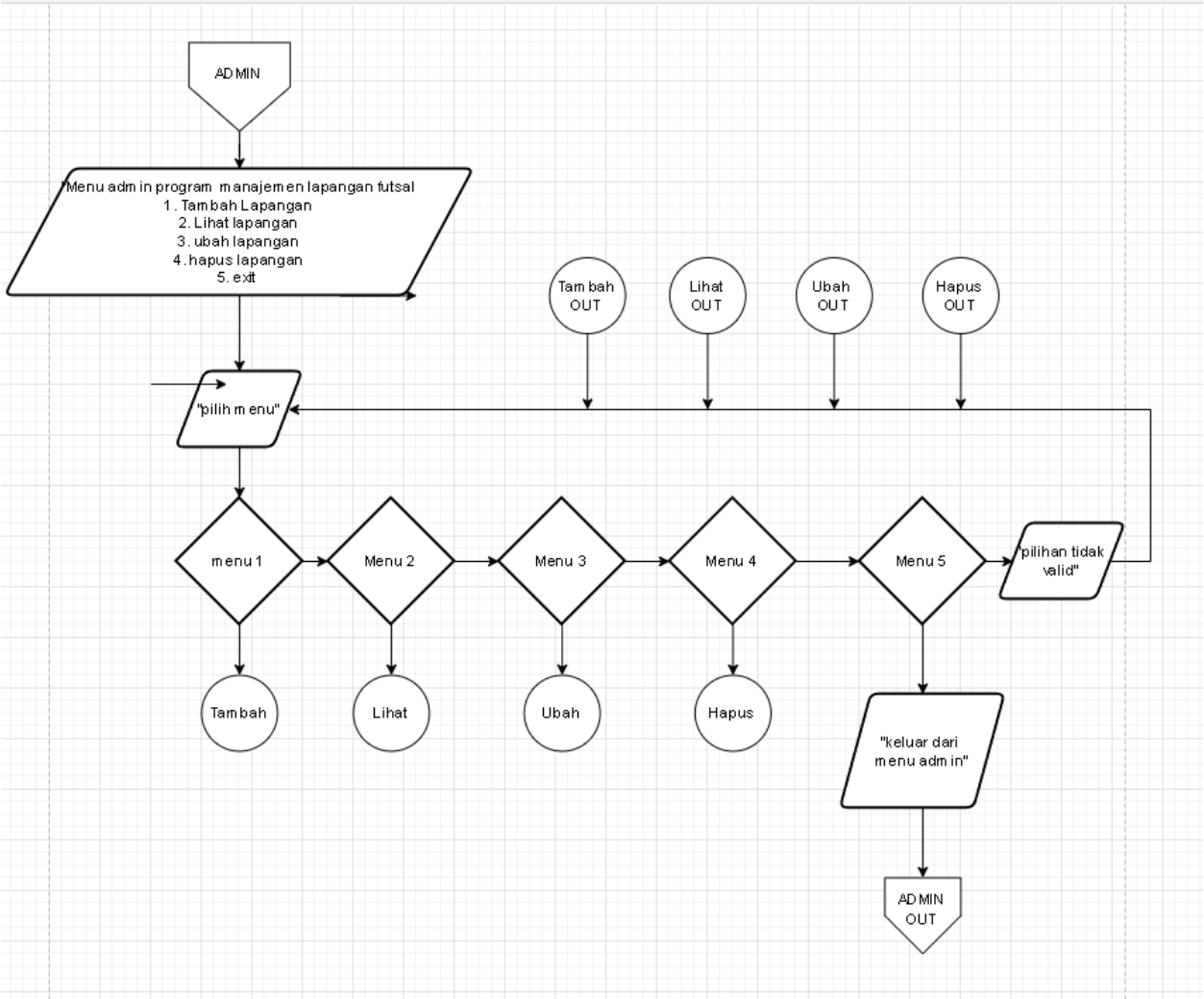
SAMARINDA

2025

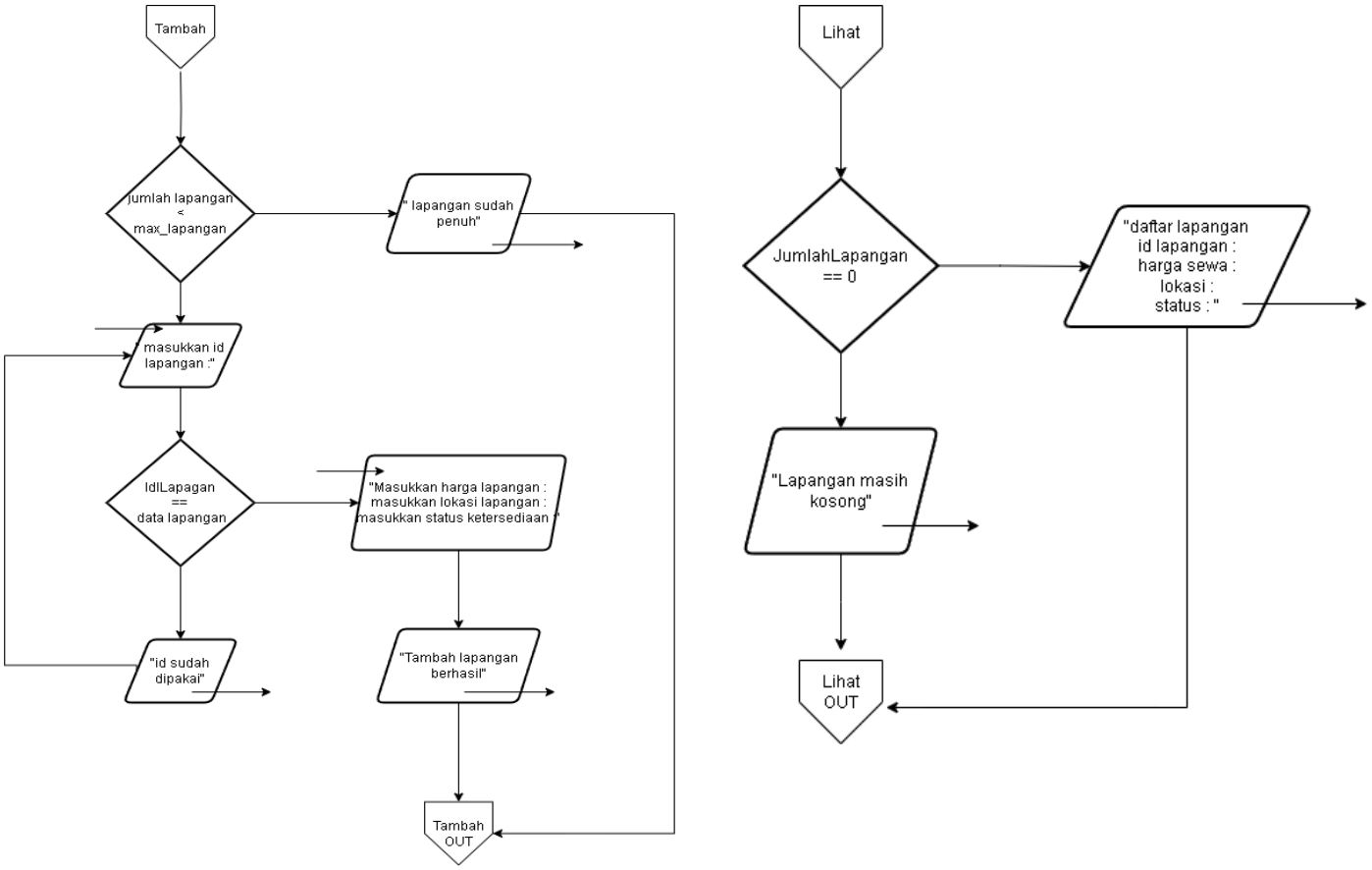
1. Flowchart



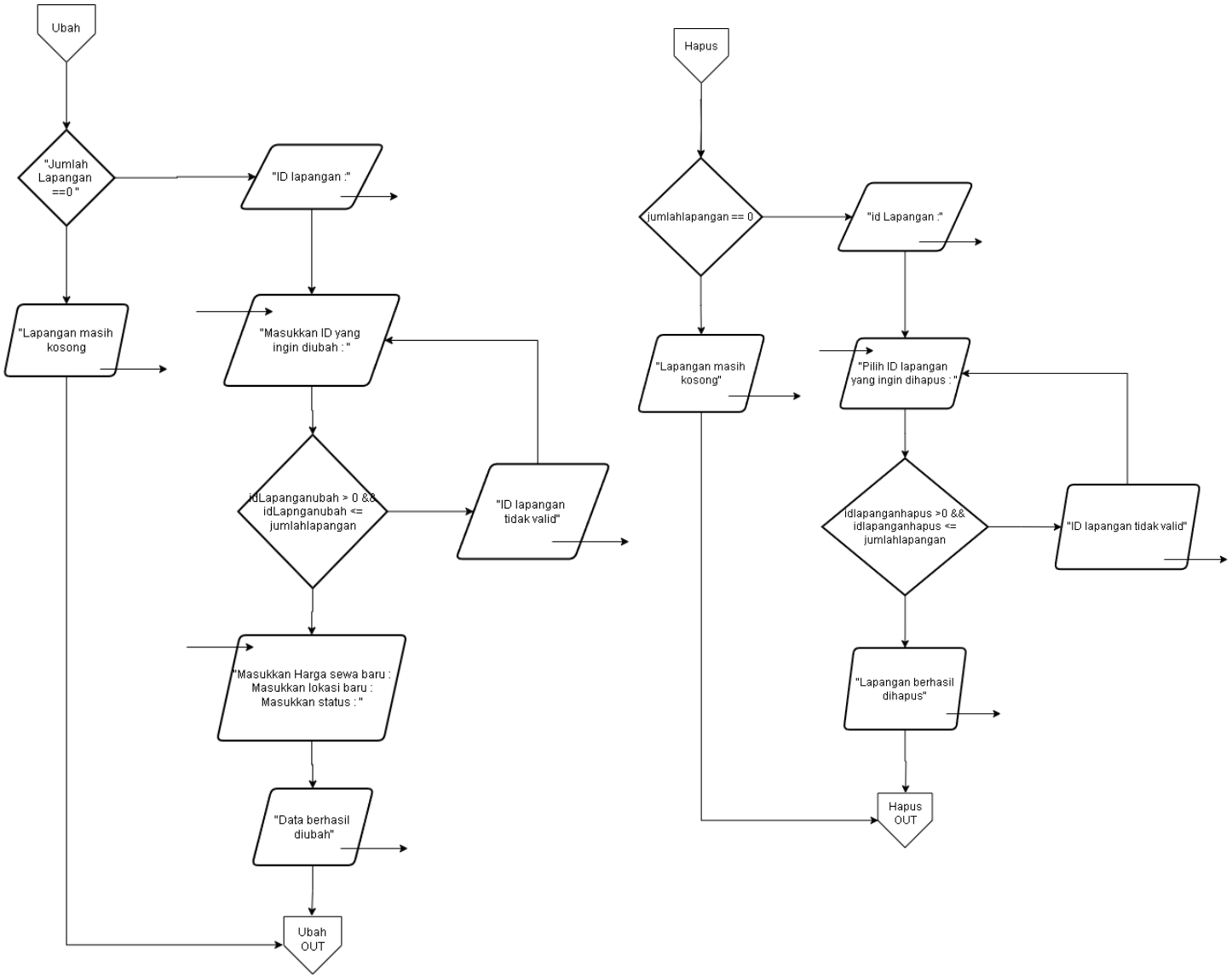
Gambar 1.1 Flow register/login



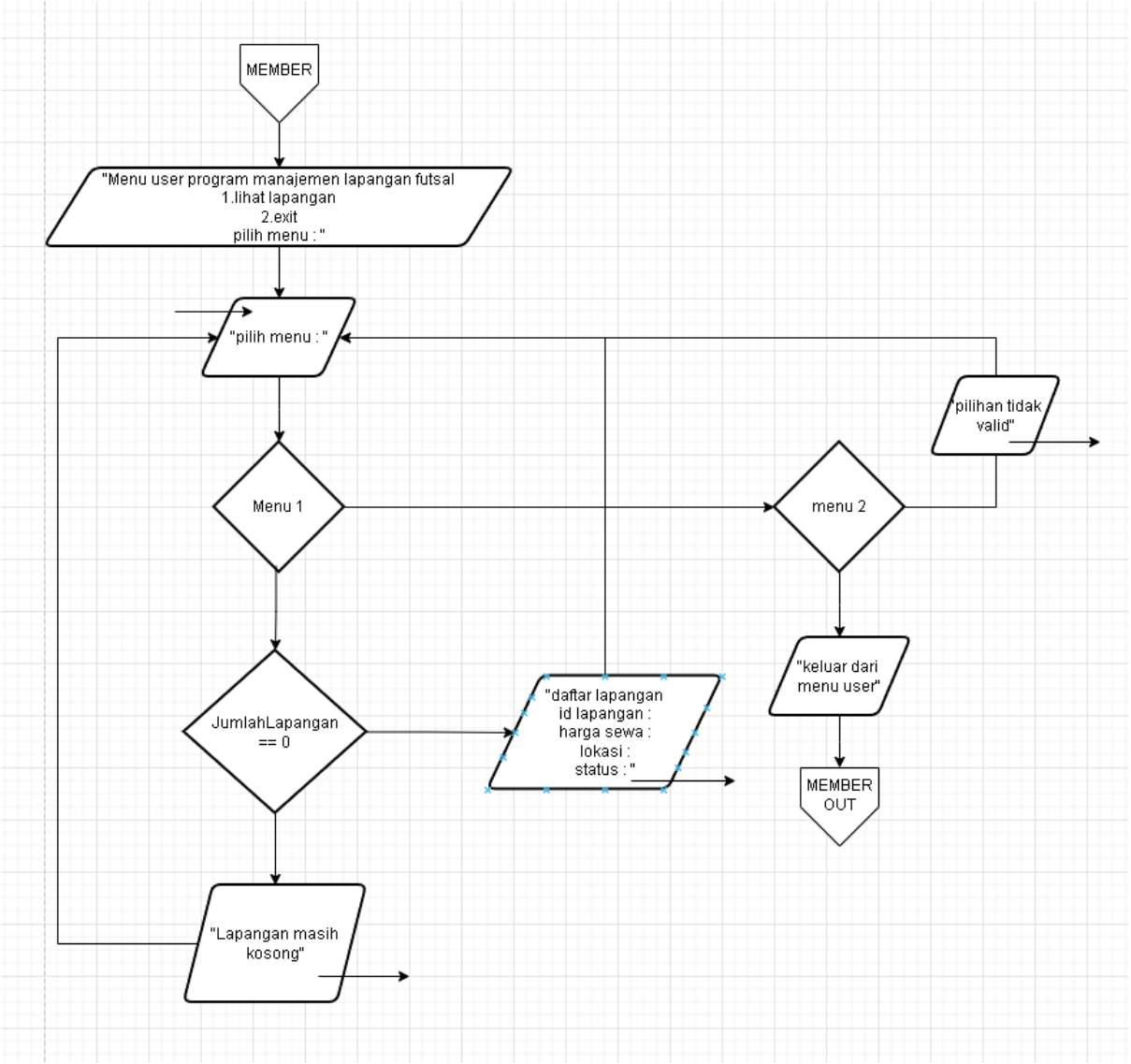
Gambar 1.2 Flow CRUD Admin



Gambar 1.3Flow CRUD (Tambah dan Lihat)



Gambar 1.4 Flow CRUD(Ubah dan Hapus)



Gambar 1.5 Menu member

2. Analisis Program

2.1 Deskripsi Singkat Program

jadi program ini merupakan program Manajemen Lapangan Futsal yang menggunakan multiuser yaitu admin dan member. admin bisa melakukan tambah, lihat, ubah, hapus data lapangan sedangkan user hanya bisa melihat data lapangan

3. Source Code

A. Fitur Register

```
void registrasiPengguna(Pengguna* daftarPengguna, int* dataUser) {
    if (*dataUser < MAX_USER) {
        cout << "Masukkan Nama: ";
        cin >> daftarPengguna[*dataUser].nama;
        cout << "Masukkan NIM: ";
        cin >> daftarPengguna[*dataUser].nim;
        cout << "Masukkan Peran (admin/member): ";
        cin >> daftarPengguna[*dataUser].peran;

        cout << "Masukkan Alamat: ";
        cin.ignore();
        getline(cin, daftarPengguna[*dataUser].kontak.alamat);

        cout << "Masukkan Nomor Telepon: ";
        cin >> daftarPengguna[*dataUser].kontak.telepon;
```

```
        (*dataUser)++;
        cout << "Registrasi berhasil!" << endl;
    } else {
        cout << "Jumlah pengguna sudah mencapai batas maksimum!" << endl;
    }
}
```

fitur ini berfungsi untuk melakukan register sebelum login

B. Fitur Login

```
void loginPengguna(Pengguna* daftarPengguna, int dataUser, string*
peranPengguna, bool* loginBerhasil) {
    string nama, nim;
    cout << "Masukkan Nama: ";
    cin >> nama;
    cout << "Masukkan NIM: ";
    cin >> nim;

    *loginBerhasil = false;
    for (int i = 0; i < dataUser; i++) {
        if (daftarPengguna[i].nama == nama && daftarPengguna[i].nim == nim) {
            cout << "Login berhasil! Peran Anda: " << daftarPengguna[i].peran <<
endl;
            *peranPengguna = daftarPengguna[i].peran;
            *loginBerhasil = true;
            return;
        }
    }
    cout << "Nama atau NIM tidak ditemukan!" << endl;
}
```

Fitur ini berfungsi untuk melakukan login

C. Fitur Tambah lapangan

```
void tambahLapangan(Lapangan* daftarLapangan, int* jumlahLapangan) {
    if (*jumlahLapangan < MAX_LAPANGAN) {
        cout << "Masukkan ID lapangan: ";
        cin >> daftarLapangan[*jumlahLapangan].id;

        cout << "Masukkan harga sewa lapangan: ";
        cin >> daftarLapangan[*jumlahLapangan].harga;

        cout << "Masukkan lokasi lapangan: ";
        cin.ignore();
        getline(cin, daftarLapangan[*jumlahLapangan].lokasi);
```

```
        cout << "Masukkan status ketersediaan lapangan: ";
        cin >> daftarLapangan[*jumlahLapangan].status;

        cout << "Tambah Lapangan Berhasil !!" << endl;
        (*jumlahLapangan)++;
    } else {
        cout << "Lapangan sudah penuh!" << endl;
    }
}
```

Fitur ini hanya bisa diakses oleh admin berfungsi untuk menambah data lapangan

D. Fitur Lihat Data Lapangan

```
void lihatLapangan(Lapangan* daftarLapangan, int jumlahLapangan) {
    if (jumlahLapangan == 0) {
        cout << "Lapangan masih kosong!" << endl;
    } else {
        cout << "\n=== DAFTAR LAPANGAN FUTSAL ===" << endl;
        cout << "+" << setw(10) << "ID"
            << setw(15) << "Harga"
            << setw(20) << "Lokasi"
            << setw(15) << "Status" << endl;

        for (int i = 0; i < jumlahLapangan; i++) {
            cout << "+"
                << setw(10) << daftarLapangan[i].id
                << setw(15) << daftarLapangan[i].harga
                << setw(20) << daftarLapangan[i].lokasi
                << setw(15) << daftarLapangan[i].status << endl;
        }
    }
}
```

Fitur ini berfungsi untuk melihat data lapangan

E. Fitur Ubah Data Lapangan

```
void ubahLapangan(Lapangan* daftarLapangan, int jumlahLapangan) {
    cout << "Masukkan ID lapangan yang ingin diubah: ";
    string idLapangan;
    cin >> idLapangan;
    bool found = false;
    for (int i = 0; i < jumlahLapangan; i++) {
        if (daftarLapangan[i].id == idLapangan) {
            cout << "Masukkan harga baru: ";
```

```
            cin >> daftarLapangan[i].harga;
            cout << "Masukkan lokasi baru: ";
            cin.ignore();
            getline(cin, daftarLapangan[i].lokasi);
            cout << "Masukkan status baru: ";
            cin >> daftarLapangan[i].status;
            cout << "Ubah Lapangan Berhasil !!" << endl;
            found = true;
            break;
        }
    }
    if (!found) {
        cout << "ID lapangan tidak ditemukan!" << endl;
    }
}
```

Fitur ini digunakan untuk menambah data lapangan

F. Fitur Hapus Data Lapangan

```
void hapusLapangan(Lapangan* daftarLapangan, int* jumlahLapangan) {
    cout << "Masukkan ID lapangan yang ingin dihapus: ";
    string idLapangan;
    cin >> idLapangan;
    bool found = false;
    for (int i = 0; i < *jumlahLapangan; i++) {
        if (daftarLapangan[i].id == idLapangan) {
            for (int j = i; j < *jumlahLapangan - 1; j++) {
                daftarLapangan[j] = daftarLapangan[j + 1];
            }
            (*jumlahLapangan)--;
            cout << "Hapus Lapangan Berhasil !!" << endl;
            found = true;
            break;
        }
    }
    if (!found) {
        cout << "ID lapangan tidak ditemukan!" << endl;
    }
}
```

fitur ini berfungsi untuk menghapus data lapangan

G. Fitur Sorting Bubble Sort (Berdasarkan Lokasi Lantai)

```
void bubbleSortLokasi(Lapangan* daftarLapangan, int jumlahLapangan) {
    for (int i = 0; i < jumlahLapangan - 1; i++) {
        for (int j = 0; j < jumlahLapangan - i - 1; j++) {
            if (daftarLapangan[j].lokasi > daftarLapangan[j + 1].lokasi) {
                swap(daftarLapangan[j], daftarLapangan[j + 1]);
            }
        }
    }
}
```

Fitur ini berfungsi mensorting data lapangan berdasarkan lokasi lantai dari terkecil ke terbesar

H. Fitur Sorting Selection Sort (Berdasarkan Harga)

```
void selectionSortHarga(Lapangan* daftarLapangan, int jumlahLapangan) {
    for (int i = 0; i < jumlahLapangan - 1; i++) {
        int maxIdx = i;
        for (int j = i + 1; j < jumlahLapangan; j++) {
            if (daftarLapangan[j].harga > daftarLapangan[maxIdx].harga) {
                maxIdx = j;
            }
        }
        swap(daftarLapangan[i], daftarLapangan[maxIdx]);
    }
}
```

Fitur Ini berfungsi mensorting data lapangan berdasarkan Harga dari terbesar ke terkecil.

I. Fitur Sorting Insertion Sort (Berdasarkan Status)

```
void insertionSortStatus(Lapangan* daftarLapangan, int jumlahLapangan) {
    for (int i = 1; i < jumlahLapangan; i++) {
        Lapangan key = daftarLapangan[i];
        int j = i - 1;
        while (j >= 0 && daftarLapangan[j].status > key.status) {
            daftarLapangan[j + 1] = daftarLapangan[j];
            j--;
        }
        daftarLapangan[j + 1] = key;
    }
}
```

Fitur Ini berfungsi mensorting data lapangan berdasarkan Status dari terkecil ke terbesar

4. Hasil Output

4.1 Hasil Output

```
============================
Menu Registrasi/Login
1. Registrasi
2. Login
3. Exit
============================
Pilih menu: 1
Masukkan Nama: naufal
Masukkan NIM: 6049
Masukkan Peran (admin/member): admin
Masukkan Alamat: sempaja
Masukkan Nomor Telepon: 083153058333
Registrasi berhasil!
```

Gambar 4.1.1 Menu register

```
============================
Menu Registrasi/Login
1. Registrasi
2. Login
3. Exit
============================
Pilih menu: 2
Masukkan Nama: naufal
Masukkan NIM: 6049
Login berhasil! Peran Anda: admin
```

Gambar 4.1.2 Menu Login

```
============================
Menu Admin
1. Tambah lapangan
2. Lihat lapangan
3. Ubah lapangan
4. Hapus lapangan
5. Exit
============================
Pilih menu: 1
Masukkan ID lapangan: 1
Masukkan harga sewa lapangan: 15000
Masukkan lokasi lapangan: lantai-1
Masukkan status ketersediaan lapangan: tersedia
Tambah Lapangan Berhasil !!
```

Gambar 4.1.3 Menu Admin (Tambah lapanagan)

```
============================
Menu Admin
1. Tambah lapangan
2. Lihat lapangan
3. Ubah lapangan
4. Hapus lapangan
5. Exit
============================
Pilih menu: 2

=== DAFTAR LAPANGAN FUTSAL ===
+        ID          Harga             Lokasi         Status
+         1          15000           lantai-1       tersedia
```

Gambar 4.1.4 Menu Admin (Lihat Lapangan)

```
============================
Menu Admin
1. Tambah lapangan
2. Lihat lapangan
3. Ubah lapangan
4. Hapus lapangan
5. Exit
============================
Pilih menu: 3
Masukkan ID lapangan yang ingin diubah: 1
Masukkan harga baru: 14000
Masukkan lokasi baru: lantai-1
Masukkan status baru: booking
Ubah Lapangan Berhasil !!
```

Gambar 4.1.5 Menu Admin (Ubah Lapangan)

```
============================
Menu Admin
1. Tambah lapangan
2. Lihat lapangan
3. Ubah lapangan
4. Hapus lapangan
5. Exit
============================
Pilih menu: 2

=== DAFTAR LAPANGAN FUTSAL ===
+        ID          Harga              Lokasi         Status
+         1          14000            lantai-1        booking
```

Gambar 4.1.6 Menu Admin (lihat lapangan setelah diubah)

```
============================
Menu Admin
1. Tambah lapangan
2. Lihat lapangan
3. Ubah lapangan
4. Hapus lapangan
5. Exit
============================
Pilih menu: 4
Masukkan ID lapangan yang ingin dihapus: 1
Hapus Lapangan Berhasil !!
```

Gambar 4.1.7 Menu Admin (lihat Lapangan)

```
=== Daftar Lapangan ===
ID Lapangan    Harga     Lantai         Status
1              15000     2              Tersedia
3              20000     3              Tidak Tersedia
2              10000     4              Tidak Tersedia
```

Gambar 4.1.8 Data Lapangan (Sebelum di Sorting)

```
=== Daftar Lapangan ===
ID Lapangan    Harga     Lantai         Status
1              15000     2              Tersedia
3              20000     3              Tidak Tersedia
2              10000     4              Tidak Tersedia
```

Gambar 4.1.9 Data Lapangan (Sorting berdasarkan Lokasi lantai)

```
=== Daftar Lapangan ===
ID Lapangan     Harga      Lantai          Status
3               20000      3               Tidak Tersedia
1               15000      2               Tersedia
2               10000      4               Tidak Tersedia
```

Gambar 4.1.10 Data Lapangan (Sorting berdasarkan Harga)

```
=== Daftar Lapangan ===
ID Lapangan     Harga      Lantai          Status
1               15000      2               Tersedia
3               20000      3               Tidak Tersedia
2               10000      4               Tidak Tersedia
```

Gambar 4.1.11 Data Lapangan (Sorting berdasarkan Status)

5. Langkah - Langkah Git

A. Git Add

```
USER@DESKTOP-3005TS7 MINGW64 /d/Praktikum APL (main)
$ git add .
```

Gambar 5.1 Git add

Digunakan untuk menambahkan file apa saja yang ingin kita commit selajutnya.

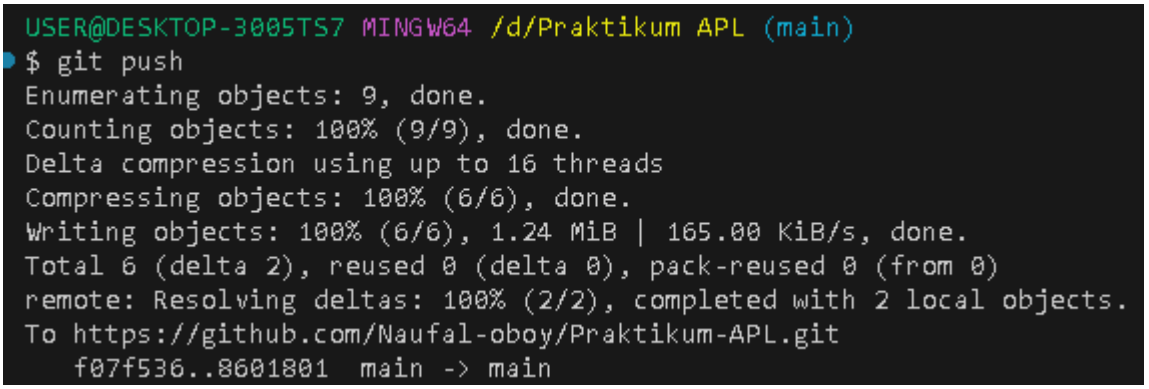
B. Git Commit

```
$ git commit -m "posttest 5 selesai"
[main 8601801] posttest 5 selesai
 3 files changed, 0 insertions(+), 0 deletions(-)
 rename Posttest/posttest-5/{main.cpp => 2409106049-MuhammadNaufalAdiBrata-PT-5.cpp} (100%)
 rename Posttest/posttest-5/{main.exe => 2409106049-MuhammadNaufalAdiBrata-PT-5.exe} (58%)
 create mode 100644 Posttest/posttest-5/2409106049-MuhammadNaufalAdiBrata-PT-5.pdf
```

Gambar 5.2 Git commit

Digunakan untuk membuat checkpoint pada file

C. Git Push



Gambar 5.3 Git push

Digunakan untuk mengupload semua hal yang ada pada repository lokal kita ke Github.