

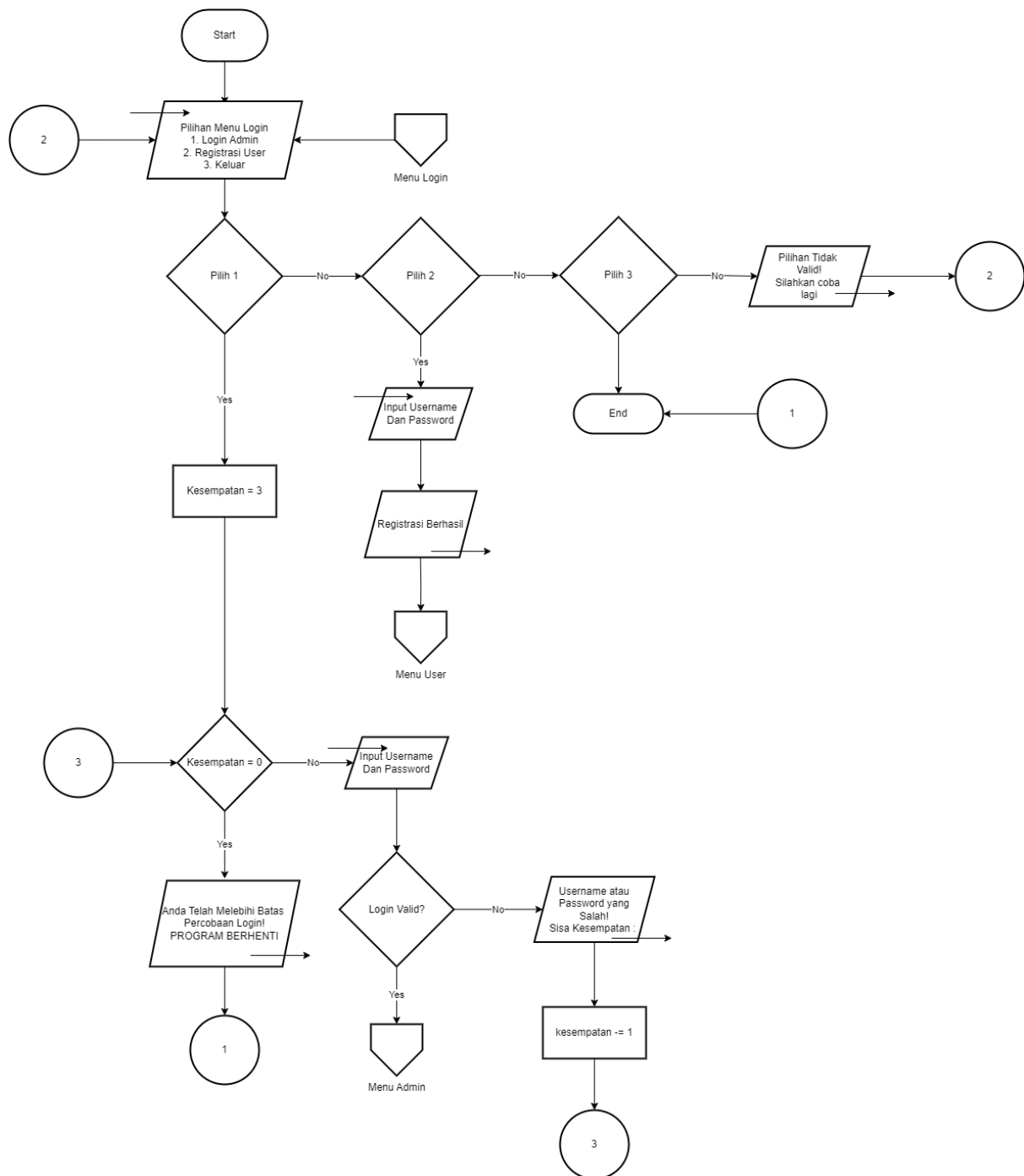
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



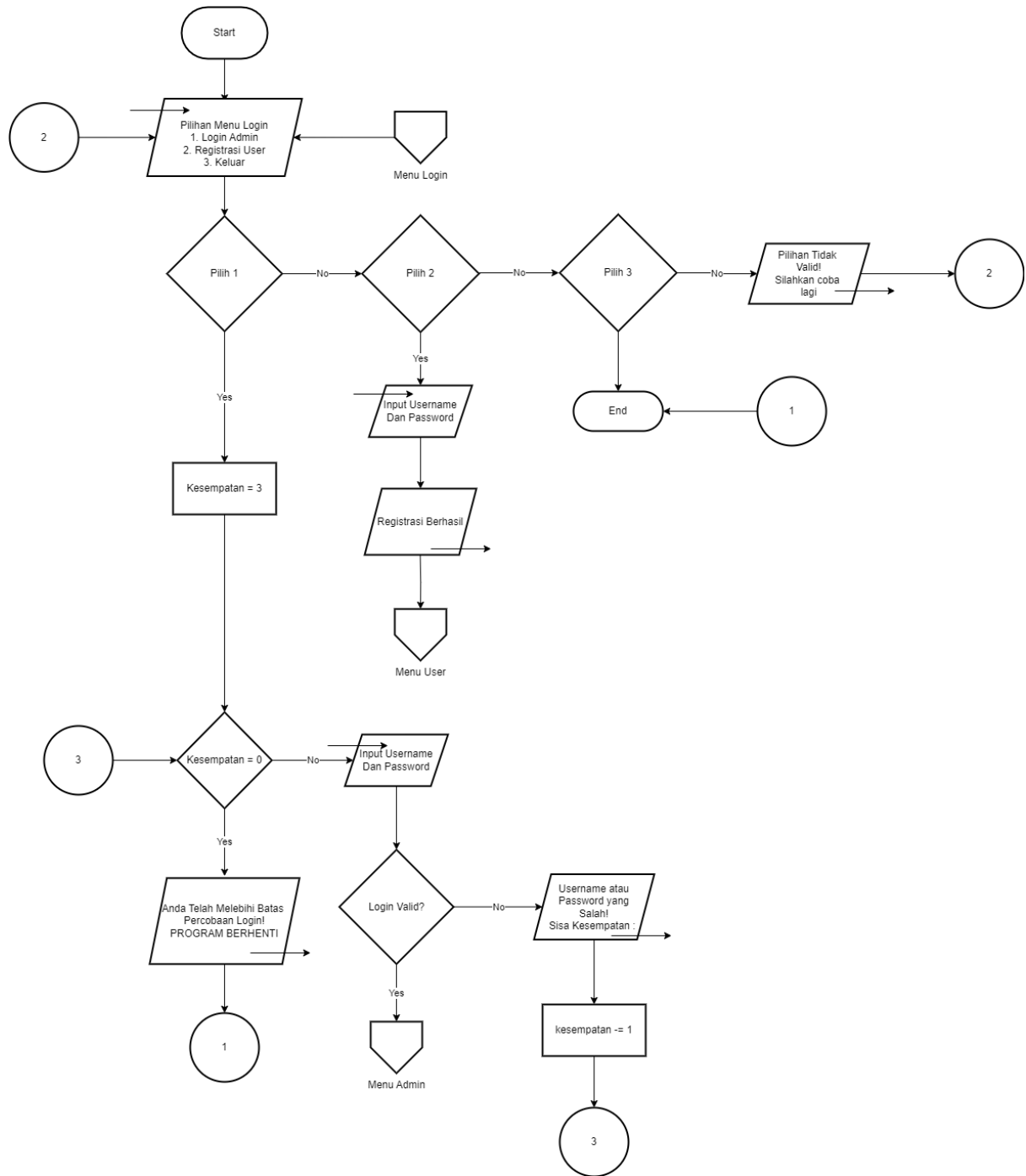
Disusun oleh:
Anindtya Puji Astari (2409106063)
Kelas B1 '24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

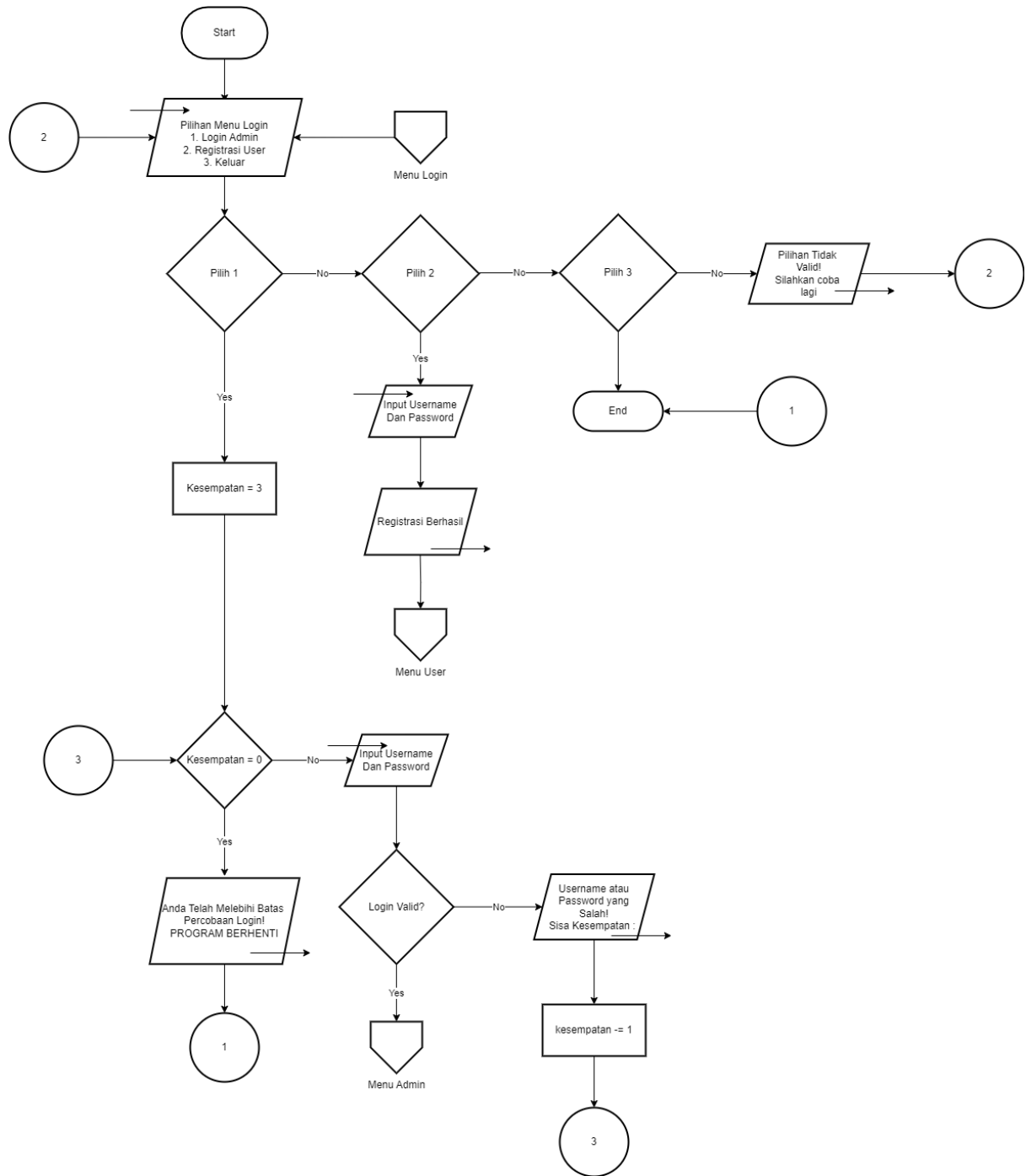
1. Flowchart



Gambar 1.1 Flowchart Menu Login



Gambar 1.2 Flowchart Menu Admin



Gambar 1.3 Flowchart Menu User

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan sistem reservasi studio musik yang memungkinkan pengguna untuk melakukan pemesanan studio musik. Program menyediakan dua jenis akses, yaitu admin dan user. Menu admin terbatas pada dapat melihat seluruh reservasi dan menghapus reservasi, sedangkan user dapat melihat daftar studio, melakukan reservasi, mengedit reservasi, serta melihat reservasi pribadi yang sudah dibuat.

2.2 Penjelasan Alur & Algoritma

1. Inisialisasi Data

- Program memulai dengan mendeklarasikan struktur Reservasi yang menyimpan informasi nama, nim, kelas studio, jumlah jam pemakaian, dan total harga.
- Array daftarReservasi digunakan untuk menyimpan daftar reservasi dengan kapasitas maksimal 100.
- Array studioKelas menyimpan dua jenis studio yang tersedia: "Kelas A" dan "Kelas B".
- Array alatMusik menyimpan daftar alat musik yang tersedia di masing-masing kelas studio.
- Variabel hargaPerJam menyimpan harga sewa per jam untuk masing-masing kelas.
- Fungsi tambahReservasiAwal() digunakan untuk menambahkan beberapa data reservasi awal sebagai contoh. Dimana dalam program ini masih dibatasi tanpa adanya file eksternal yang dapat menyimpan data reservasi dari user, sehingga sebagai contoh digunakannya data reservasi awal.
- **Variabel input** seperti nama, nim, pilihan, dan jam digunakan untuk menyimpan input dari pengguna.

2. Login User atau Admin

- Pengguna diberikan pilihan untuk login sebagai admin atau user.
- Jika memilih admin, pengguna harus memasukkan username dan password admin yang benar dalam `loginAdmin()`. Admin memiliki maksimal 3 kali percobaan login. Jika salah menginput username ataupun password sebanyak 3 kali, maka program akan berhenti.
- Jika memilih user, pengguna cukup memasukkan nama dan nim untuk login, dan tidak ada batasan untuk percobaan login karena user selalu dapat membuat akun baru dengan username dan password baru.

3. Menu Admin

Jika login sebagai Admin berhasil, admin memiliki dua opsi utama:

- Melihat semua reservasi : Menampilkan seluruh daftar reservasi menggunakan fungsi `tampilkanReservasiAdmin()`.
- Menghapus reservasi : Admin dapat memilih nomor reservasi yang ingin dihapus menggunakan `hapusReservasiAdmin()`.

4. Menu User

Jika login sebagai User berhasil, pengguna akan diberikan beberapa pilihan:

- Melihat daftar studio musik : Menampilkan daftar studio, alat musik yang tersedia, dan harga per jam dengan `tampilkanDaftarStudio()`.
- Menambah reservasi baru : User dapat memilih studio, menentukan jumlah jam pemakaian, dan harga total dihitung secara otomatis melalui `tambahReservasiUser()`.
- Mengedit reservasi sendiri : User dapat memilih reservasi yang telah dibuat sebelumnya dan mengubah jumlah jamnya melalui `editReservasiUser()`.
- Melihat reservasi sendiri : Menampilkan daftar reservasi yang dimiliki user menggunakan `tampilkanReservasiUser()`.

5. Pengelolaan Data

- Menambah Reservasi : Data baru ditambahkan ke dalam array `daftarReservasi` dengan indeks `jumlahReservasi` yang terus bertambah.
- Mengedit Reservasi : Data jumlah jam yang ada dalam array diperbarui, kemudian total harga dihitung ulang.
- Menghapus Reservasi : Data dalam array dihapus dengan cara menggeser semua elemen setelahnya untuk menghindari celah kosong.

6. Validasi Input

- Login Admin : Username dan password harus sesuai dengan yang telah ditentukan (`username = "tea"`, `password = "063"`).
- Pilihan Menu : Memastikan input menu berada dalam rentang yang valid.
- Pilihan Kelas Studio : User hanya bisa memilih kelas studio dalam rentang 1 atau 2.
- Jumlah Jam : Memastikan input berupa angka dan tidak negatif.

7. Looping Menu

- Program menggunakan loop `do-while` agar menu tetap muncul hingga user memilih keluar. Jika Admin memilih keluar, maka program berhenti. Jika User memilih keluar, maka program juga akan berhenti.

8. Keluar dari Program

- Jika user memilih opsi Keluar, program akan mencetak pesan perpisahan dan berhenti.

Penjelasan Detail Blok Kode

- Loop do-while (Menu Utama) : Memastikan menu terus berjalan hingga pengguna memilih keluar.
- Loop for (Menampilkan Data) : Digunakan untuk menampilkan daftar reservasi dalam format tabel.
- Loop for (Menghapus Reservasi) : Menggeser data setelah penghapusan agar tidak ada celah kosong dalam array.
- if-else (Validasi Input) : Digunakan untuk memastikan input menu, jumlah jam, dan data lainnya valid sebelum diproses.
- Fungsi tambahReservasiUser() : Menyediakan input bagi user untuk menambahkan reservasi baru dengan data yang dihitung otomatis.
- Fungsi editReservasiUser() : Mengizinkan user untuk mengubah jumlah jam reservasi yang sudah ada.
- Fungsi hapusReservasiAdmin() : Admin dapat menghapus reservasi dengan menggeser data dalam array.
- Fungsi tampilkanReservasiUser() : Menampilkan daftar reservasi yang telah dibuat oleh user tertentu.
- Fungsi tampilkanReservasiAdmin() : Menampilkan semua reservasi yang tersedia untuk admin.

3. Source Code

A. Fitur Verifikasi Admin

Fitur ini digunakan untuk login admin dengan batas percobaan 3 kali.

Source Code:

```
bool loginAdmin()
{
    string username,
    password;    int kesempatan
    = 3;    while (kesempatan >
    0)
    {
        cout << "\n--==LOGIN ADMIN==--\nMasukkan Username:
";        cin >> username;        cout << "Masukkan Password: ";
    cin >> password;        if (username == "tea" && password == "063")
    return true;        kesempatan--;        cout << "Username atau
    Password Salah!\nSisa kesempatan: " << kesempatan << endl;
    }    return
    false;
}
```

B. Fitur Verifikasi User

Fitur ini digunakan untuk login user.

Source Code:

```
string registrasiUser(string &nama, string &nim)
{
    cout << "\n--==REGISTRASI USER==--\nMasukkan Nama:
";    cin >> nama;    cout << "Masukkan NIM: ";    cin
    >> nim;
    return "Registrasi Berhasil!\n\n"; }
}
```

C. Fitur Tampilkan Daftar Studio

Fitur ini digunakan untuk menampilkan Daftar Studio Musik mulai dari kelas, alat musik, dan harga per jamnya.

Source Code:

```
string tampilkanDaftarStudio()
{
    string
    output =
    "\n=====
=====\\n";    output
    += " | No | Kelas   | Alat Musik
    | Harga Per Jam |\\n";
    output +=
    "=====
=====\\n";
    for (int i = 0; i < 2; i++)
    {
        output += " | " + to_string(i + 1) + " | " + studioKelas[i] +
    " | ";
        for (int j = 0; j < 7; j++)
        {
            output +=
alatMusik[i][j];
            if (j <
6) output += ", ";
        }
        output += " | Rp " + to_string(hargaPerJam[i]) +
    "\\n";
    }
    output +=
    "=====
=====\\n";
    return output;
}
```

D. Fitur Tambah Reservasi

Fitur ini digunakan untuk Reservasi Studio Musik oleh User.

Source Code:

```
string tambahReservasiUser(string nama, string nim)
{
    int pilihan, jam;    cout <<
tampilkanDaftarStudio();
```

```

        cout << "Pilih Kelas Studio (1/2): ";    cin >> pilihan;    if
(pilihan < 1 || pilihan > 2) return "Pilihan Tidak Valid!\n";    cout <<
"Masukkan jumlah jam: ";    cin >> jam;
daftarReservasi[jumlahReservasi++] = {nama, nim, studioKelas[pilihan -
1], jam, jam * hargaPerJam[pilihan - 1]};
return "Reservasi Berhasil Ditambahkan!\n"; }

```

E. Fitur Edit Reservasi

Fitur ini digunakan untuk Mengedit Reservasi Studio Musik oleh User.

Source Code:

```

string editReservasiUser(string nim)
{
    int index, jam, pilihan;    cout <<
tampilkanReservasiUser(nim);    cout << "Masukkan nomor reservasi
yang akan diedit: ";    cin >> index;    if (index < 1 || index >
jumlahReservasi || daftarReservasi[index -
1].nim != nim)        return "Reservasi Tidak
Ditemukan!\n";    cout << "\nPilih Kelas Studio Baru:\n";
cout << "1. Kelas A (Rp " << hargaPerJam[0] << "/jam)\n";
cout << "2. Kelas B (Rp " << hargaPerJam[1] << "/jam)\n";
cout << "Pilihan: ";    cin >> pilihan;
    if (pilihan < 1 || pilihan > 2)        return "Pilihan Tidak
Valid!\n";    daftarReservasi[index - 1].kelasStudio =
studioKelas[pilihan - 1];    cout << "Masukkan jumlah jam baru: ";
cin >> jam;    daftarReservasi[index - 1].jam = jam;
daftarReservasi[index - 1].totalHarga = jam * hargaPerJam[pilihan - 1];
return "Reservasi Berhasil Diperbarui!\n"; }

```

F. Fitur Hapus Reservasi

Fitur ini digunakan untuk Hapus Reservasi Studio Musik oleh Admin.

Source Code:

```
string hapusReservasiAdmin()
{
    if (jumlahReservasi == 0) return "Tidak Ada Reservasi Yang Bisa Dihapus.\n";
    cout << tampilkanReservasiAdmin();    int index;
    cout << "Masukkan nomor reservasi yang akan dihapus: ";
    cin >> index;
    if (index < 1 || index > jumlahReservasi) return "Nomor Reservasi Tidak Valid!\n";
    for (int i = index - 1; i < jumlahReservasi - 1; i++)
    {
        daftarReservasi[i] =
        daftarReservasi[i + 1];
    }
    jumlahReservasi--;
    return "Reservasi Berhasil Dihapus!\n"; }

```

G. Fitur Tampilkan Reservasi User

Fitur ini digunakan untuk Menampilkan Reservasi Studio Musik oleh User.

Source Code:

```
string tampilkanReservasiUser(string nimUser)
{
    string output = "\nReservasi Anda:\n";    output +=
    "=====\n";    output
    += "| No | Kelas | Jam | Total Harga | \n";
    output += "=====\n";
    bool adaReservasi = false;    for
    (int i = 0; i < jumlahReservasi; i++)    {

```

```

        if (daftarReservasi[i].nim == nimUser)
        {
            adaReservasi = true;
            output += "|
" + to_string(i + 1) + " | " + daftarReservasi[i].kelasStudio + " |
" +
            to_string(daftarReservasi[i].jam) + " |
Rp " +
to_string(daftarReservasi[i].totalHarga) + "
|\n";
        }
        if (!adaReservasi) output += "|
Tidak ada reservasi |\n";
        output +=
"=====\n";
        return
output;
    }
}

```

H. Menu Utama

Fitur ini sebagai menu utama.

Source Code:

```

int main()
{
    int pilihan;
    bool
isAdmin = false;
    string
namaUser, nimUser;

    cout << tambahReservasiAwal();
    while
(true)
    {
        cout << "Selamat Datang di Sistem Reservasi Studio Musik\n";
        cout << "1. Login Admin\n2. Registrasi User\n3. Keluar\nPilih: ";
        cin
>> pilihan;

        if
(pilihan == 1)
        {
            isAdmin = loginAdmin();
            if (!isAdmin)
            {
                cout << "Anda Telah Melebihi Batas Percobaan Login!\nPROGRAM
BERHENTI.\n";
            }
        }
    }
}

```

```

        return 0;
    }
}
else if (pilihan == 2)
{
    cout << registrasiUser(namaUser,
nimUser);
    isAdmin = false;
}
else if
(pilihan == 3)
{
    cout << "\nTerima kasih telah menggunakan sistem
ini.\n";
    return 0;
}
else {
    cout << "\nPilihan Tidak
Valid! Silakan coba lagi.\n\n";
    continue;
}
while (true)
{
    cout <<
"\nMENU:\n";
    if
(isAdmin)
    {
        cout << "1. Lihat Semua Reservasi\n2. Hapus
Reservasi\n3.
Logout\n";
    }
    else
    {
        cout << "1. Lihat Daftar Studio Musik\n2. Tambah
Reservasi\n3. Edit Reservasi\n4. Lihat Reservasi Saya\n5. Logout\n";
    }
    cout << "Pilih menu: ";
    cin >> pilihan;

    if (isAdmin)
    {
        if (pilihan == 1) cout <<
tampilkanReservasiAdmin();
        else if (pilihan == 2) cout
<< hapusReservasiAdmin();
        {
            cout << "\nAnda
Berhasil Logout\n\n";
            break;
        }
        else
        {
            cout << "\nPilihan Tidak Valid!\nSilakan coba lagi.\n\n";

```

```

    }
} else {
if (pilihan == 1) cout << tampilkanDaftarStudio();           else if
(pilihan == 2) cout << tambahReservasiUser(namaUser, nimUser);
else if (pilihan == 3) cout << editReservasiUser(nimUser);
else if (pilihan == 4) cout << tampilkanReservasiUser(nimUser);
        else if (pilihan == 5)
        {
            cout << "\nAnda
Berhasil Logout\n\n";                                break;
        }
        else {
cout << "\nPilihan Tidak Valid!\nSilakan coba lagi.\n\n";
        }
    }
}
}
}
```

4. Uji Coba dan Hasil Output

4.1 Uji Coba

1. Skenario 1
Login sebagai admin dan lihat semua menu.
2. Skenario 2
Login sebagai user, tambah reservasi baru, lalu lihat apakah reservasi berhasil ditambahkan.
3. Skenario 3
Edit reservasi pribadi user yang telah ditambahkan.
4. Skenario 4
Hapus reservasi sebagai admin.

4.2 Hasil Output

```
HELLO
Selamat Datang di Sistem Reservasi Studio Musik
1. Login Admin
2. Registrasi User
3. Keluar
Pilih: 1

====LOGIN ADMIN====
Masukkan Username: tea
Masukkan Password: 063

MENU:
1. Lihat Semua Reservasi
2. Hapus Reservasi
3. Logout
Pilih menu: 1

=====
| No | Nama          | NIM      | Kelas | Jam | Total Harga |
=====
| 1  | ipeh          | 202101   | Kelas A | 3  | Rp 240000 |
| 2  | aryo          | 202102   | Kelas B | 2  | Rp 200000 |
| 3  | araa          | 202103   | Kelas A | 5  | Rp 400000 |
| 4  | inun          | 202104   | Kelas B | 4  | Rp 400000 |
=====
```

Gambar 4.1 lihat semua reservasi

```
HELLO
Selamat Datang di Sistem Reservasi Studio Musik
1. Login Admin
2. Registrasi User
3. Keluar
Pilih: 2

====REGISTRASI USER====
Masukkan Nama: tya
Masukkan NIM: 1
Registrasi Berhasil!

MENU:
1. Lihat Daftar Studio Musik
2. Tambah Reservasi
3. Edit Reservasi
4. Lihat Reservasi Saya
5. Logout
Pilih menu: 2

=====
| No | Kelas | Alat Musik                                     | Harga Per Jam |
=====
| 1  | Kelas A | Drum, Keyboard, Gitar Akustik, Gitar Elektrik, Bass, Cajon, Mic | Rp 80000 |
| 2  | Kelas B | Piano, Violin, Cello, Flute, Terompet, Saxofon, Harpa | Rp 100000 |
=====

Pilih Kelas Studio (1/2): 1
Masukkan jumlah jam: 5
Reservasi Berhasil Ditambahkan!
```

Gambar 4.2 tambah reservasi

```
Pilih menu: 3

Reservasi Anda:
=====
| No | Kelas  | Jam | Total Harga      |
=====
| 5  | Kelas A | 5   | Rp 400000        |
=====
Masukkan nomor reservasi yang akan diedit: 5

Pilih Kelas Studio Baru:
1. Kelas A (Rp 80000/jam)
2. Kelas B (Rp 100000/jam)
Pilihan: 2
Masukkan jumlah jam baru: 8
Reservasi Berhasil Diperbarui!
```

Gambar 4.3 edit reservasi

```
Pilih menu: 4

Reservasi Anda:
=====
| No | Kelas  | Jam | Total Harga      |
=====
| 5  | Kelas B | 8   | Rp 800000        |
=====
```

Gambar 4.4 reservasi pribadi


```
Pilih menu: 2

=====
| No | Nama          | NIM      | Kelas  | Jam | Total Harga |
=====
| 1  | ipeh          | 202101   | Kelas A | 3  | Rp 240000   |
| 2  | aryo          | 202102   | Kelas B | 2  | Rp 200000   |
| 3  | araa          | 202103   | Kelas A | 5  | Rp 400000   |
| 4  | inun          | 202104   | Kelas B | 4  | Rp 400000   |
=====

Masukkan nomor reservasi yang akan dihapus: 3
Reservasi Berhasil Dihapus!

MENU:
1. Lihat Semua Reservasi
2. Hapus Reservasi
3. Logout
Pilih menu: 1

=====
| No | Nama          | NIM      | Kelas  | Jam | Total Harga |
=====
| 1  | ipeh          | 202101   | Kelas A | 3  | Rp 240000   |
| 2  | aryo          | 202102   | Kelas B | 2  | Rp 200000   |
| 3  | inun          | 202104   | Kelas B | 4  | Rp 400000   |
=====
```

Gambar 4.6 hapus reservasi

5. Git

5.1 Git Add

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL (main)
$ git add .
```

Git add berfungsi untuk Memasukkan file ke staging area agar siap untuk commit.

5.2 Git Commit

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL (main)
$ git commit -m "tes upload"
[main 650ca4f] tes upload
5 files changed, 222 insertions(+)
create mode 100644 Kelas/Pertemuan-2/crud.cpp
create mode 100644 Kelas/Pertemuan-2/main.cpp
create mode 100644 Post-test/Post-test-APL-1/2409106063-AnindtyaPujiAstari-PT-1.cpp
create mode 100644 Post-test/Post-test-APL-1/2409106063-AnindtyaPujiAstari-PT-1.exe
create mode 100644 Post-test/Post-test-APL-1/2409106063-AnindtyaPujiAstari-PT-1.pdf
```

Git commit berfungsi menyimpan perubahan di repository lokal.

Pesan di -m "..." menjelaskan perubahan yang dilakukan.

5.3 Git push origin main

```
Lenovo@LAPTOP-2R0GRAM4 MINGW64 ~/Desktop/Praktikum-APL (main)
$ git pull origin main --rebase
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 7 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (7/7), 1.21 MiB | 1.23 MiB/s, done.
From https://github.com/Yyak135/Praktikum-APL
```

Git push origin main Mengunggah commit yang ada di branch main ke repository GitHub.

Jika branch belum ada di GitHub, jalankan git push -u origin main agar branch main menjadi default untuk push berikutnya.