

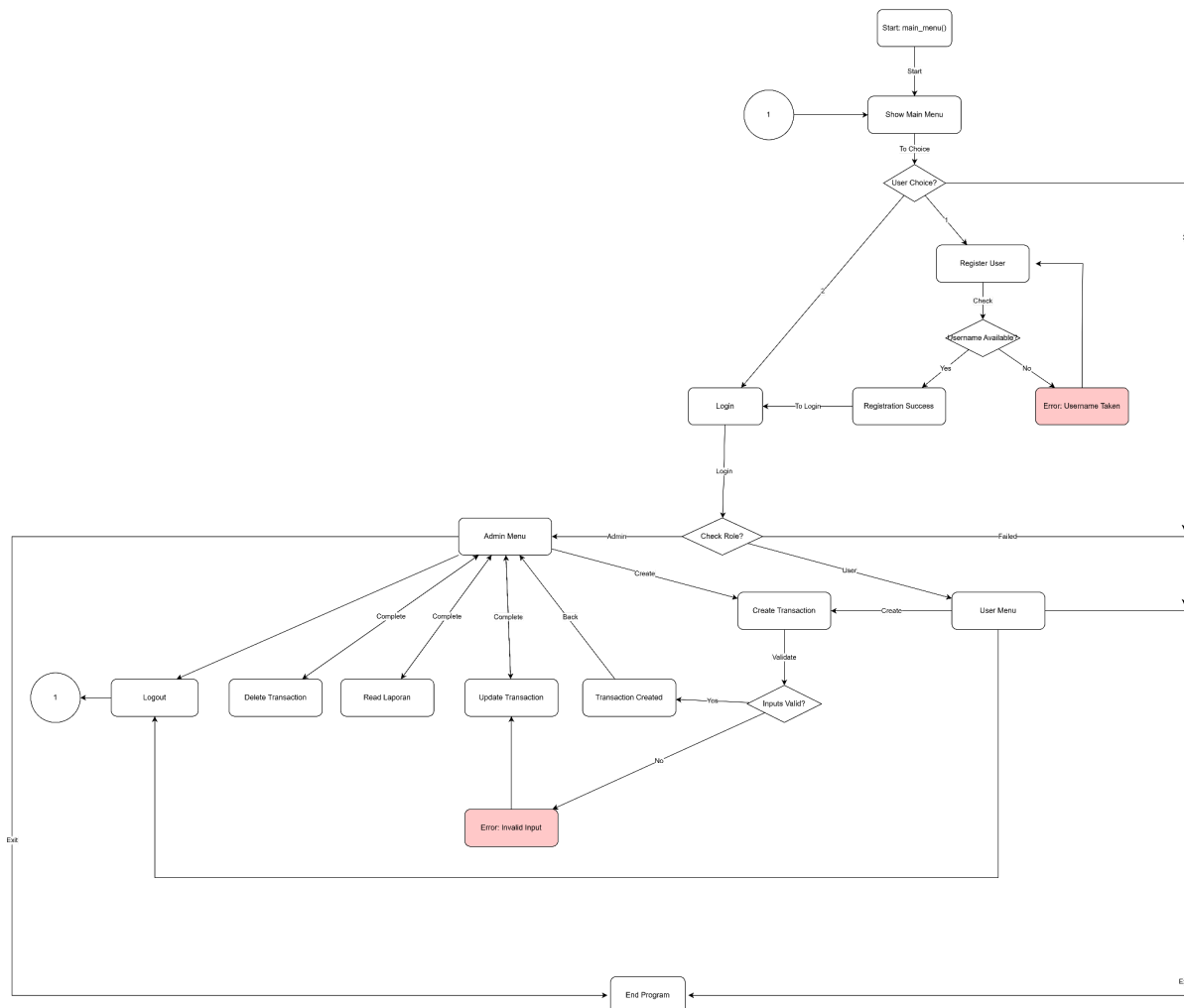
LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Arya Fickri Al Farazi (2509106022)
Kelas (A1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Deskripsi Singkat Program

Program ini adalah aplikasi berbasis terminal yang digunakan untuk mengelola penyewaan komputer di sebuah warnet. Sistem ini menggunakan struktur data berupa dictionary untuk menyimpan informasi pengguna dan transaksi selama program berjalan. Ketika setiap kali program ini dijalankan, sistem ini dibuat untuk memastikan bahwa sudah semua akun admin default dibuat dan dapat digunakan terkait untuk setiap tindakan administrator. Sistem ini dibuat untuk memastikan adanya fitur registrasi untuk pengguna baru dan sistem login yang dibuat untuk setiap peran pengguna, di mana setiap pengguna di sistem ini hanya diberikan 2 peran akses pengguna yang berbeda, yaitu admin dan user.

Sekarang, setelah login, pengguna peran user hanya bisa menyelesaikan transaksi penyewaan komputer. Sistem ini meminta untuk memasukkan waktu mulai dalam format tanggal, durasi penggunaan dalam jam, dan pemilihan nomor komputer yang diinginkan. Sistem ini akan mengulangi dan memeriksa input untuk setiap sistem, serta menghitung estimasi biaya menggunakan tarif per jam yaitu Rp 12.000, dan menyimpan setiap transaksi dengan status aktif.

Admin memiliki menu yang lebih lengkap. Selain bisa melakukan transaksi seperti user, admin bisa membuat laporan pendapatan harian berdasarkan tanggal yang ditentukan. Laporan ini berisi daftar seluruh transaksi yang aktif pada hari tersebut, beserta total pendapatan yang diperoleh. Sistem juga memiliki fungsi rekursif untuk menghitung total seluruh transaksi aktif pada saat laporan ditampilkan.

Selain membuat laporan, admin juga bisa melakukan perubahan pada data transaksi seperti mengubah durasi sewa, dan total biaya akan diperbaharui secara otomatis. Admin juga berhak untuk “menghapus” transaksi yang tidak ingin diteruskan dengan mengubah status transaksi menjadi “dibatalkan”. Fitur ini membantu pengelolaan data dengan cara transaksi tidak dihapus secara permanen dari sistem, tetapi tetap tercatat dengan status akhir.

Program ini juga dilengkapi teknologi lain, seperti `clear_screen()` untuk menghapus tampilan terminal dan menjaga kebersihan antarmuka saat berpindah menu dan juga `Error Handling`. Seluruh operasi program berjalan dalam menu utama dalam loop, sehingga program akan tetap aktif pada pengguna hingga memilih opsi untuk keluar. Struktur ini memungkinkan program digunakan sebagai sistem manajemen warnet dasar yang sederhana dan dapat diperluas, seperti penyimpanan data ke file, autentikasi yang lebih ketat, pembatasan pada jumlah komputer yang aktif, dan antarmuka grafis.

3. Source Code

```
import os
from datetime import datetime

data = {
    'users': {},
    'transactions': {}
}

HARGA_PER_JAM = 12000
MAX_PC = 10

if 'admin' not in data['users']:
    data['users']['admin'] = {'password': '123', 'role': 'admin'}
```

```

def register_user():
    username = input("Masukkan username baru: ")
    if username in data['users']:
        print("Username sudah digunakan.")
        return None
    password = input("Masukkan password: ")
    data['users'][username] = {'password': password, 'role': 'user'}
    print(f"Pengguna {username} berhasil terdaftar sebagai user.")
    return username

def login():
    username = input("Masukkan username: ")
    password = input("Masukkan password: ")
    if username in data['users'] and data['users'][username]['password'] == password:
        print(f"Login berhasil sebagai {data['users'][username]['role']}.")
        return data['users'][username]['role']
    print("Login gagal. Username atau password salah.")
    return None

def create_transaksi(waktu_mulai, durasi, nomor_pc):
    try:
        date_part = waktu_mulai.split(' ')[0]
        if len(date_part.split('-')) != 3:
            raise ValueError("Waktu mulai harus dalam format YYYY-MM-DD.")

        if not isinstance(durasi, int) or durasi <= 0:
            raise ValueError("Durasi harus angka positif.")
        if not isinstance(nomor_pc, int) or nomor_pc <= 0 or nomor_pc >
MAX_PC:
            raise ValueError(f"Nomor PC harus angka positif dan maksimal
{MAX_PC}.")

        total_biaya = HARGA_PER_JAM * durasi
        new_transaksi = {
            'id': get_next_id(),
            'waktu_mulai': waktu_mulai,
            'durasi': durasi,
            'total_biaya/jam': total_biaya,
            'nomor_pc': nomor_pc,
            'total_biaya': total_biaya,
            'status': 'aktif'

```

```

    }
    data['transactions'][new_transaksi['id']] = new_transaksi
    print(f"Transaksi berhasil dibuat ID: {new_transaksi['id']}, Dengan
    Harga/Jam: Rp 12.000, Total Biaya Menjadi: Rp {total_biaya}")
except ValueError as e:
    print(f"Error: {e}")

def read_laporan(tanggal):
    try:
        if not isinstance(tanggal, str) or len(tanggal.split('-')) != 3:
            raise ValueError("Tanggal harus dalam format YYYY-MM-DD.")
        harian = [t for t in data['transactions'].values() if
t['waktu_mulai'].split(' ')[0] == tanggal and t['status'] == 'aktif']
        total_pendapatan = sum(t['total_biaya'] for t in harian)
        print(f"Laporan Pendapatan untuk {tanggal}: Rp {total_pendapatan}")
        for t in harian:
            print(f"ID: {t['id']}, Waktu: {t['waktu_mulai']}, Total Biaya: Rp
{t['total_biaya']}")
    except ValueError as e:
        print(f"Error: {e}")

def get_next_id():
    return len(data['transactions']) + 1

def clear_screen():
    os.system('cls' if os.name == 'nt' else 'clear')

def hitung_total_aktif(transaksi_list, index=0, count=0):
    if index >= len(transaksi_list):
        return count
    if transaksi_list[index]['status'] == 'aktif':
        count += 1
    return hitung_total_aktif(transaksi_list, index + 1, count)

def update_transaksi(id_transaksi, durasi_baru):
    try:
        if id_transaksi not in data['transactions']:
            raise ValueError("Transaksi tidak ditemukan.")
        if not isinstance(durasi_baru, int) or durasi_baru <= 0:
            raise ValueError("Durasi harus angka positif.")
        t = data['transactions'][id_transaksi]

```

```

        if t['status'] == 'aktif':
            t['durasi'] = durasi_baru
            t['total_biaya'] = HARGA_PER_JAM * durasi_baru
            print(f"Transaksi ID {id_transaksi} diupdate.")
        else:
            print("Transaksi tidak aktif.")
    except ValueError as e:
        print(f"Error: {e}")

def delete_transaksi(id_transaksi):
    try:
        if id_transaksi not in data['transactions']:
            raise ValueError("Transaksi tidak ditemukan.")
        t = data['transactions'][id_transaksi]
        t['status'] = 'dibatalkan'
        print(f"Transaksi ID {id_transaksi} dibatalkan.")
    except ValueError as e:
        print(f"Error: {e}")

def main_menu():
    while True:
        print("Selamat datang! Pilih: 1. Registrasi 2. Login 3. Keluar")
        pilihan_awal = input("Masukkan pilihan: ")
        if pilihan_awal == '1':
            register_user()
            role = login()
        elif pilihan_awal == '2':
            role = login()
        elif pilihan_awal == '3':
            print("Terima kasih Telah Datang.")
            return
        else:
            print("Pilihan tidak valid.")
            continue

    while role:
        clear_screen()
        if role == 'admin':
            print("1. Buat Transaksi 2. Laporan 3. Update 4. Hapus 5. Logout")
        else:
            print("1. Buat Transaksi 2. Logout")

```

```

        pilihan = input("Pilih: ")
        if pilihan == '1':
            waktu_mulai = input("Waktu mulai (YYYY-MM-DD): ")
            try:
                durasi = int(input("Durasi: "))
                nomor_pc = int(input("Nomor PC: "))
                create_transaksi(waktu_mulai, durasi, nomor_pc)
            except ValueError:
                print("Error: Input durasi dan nomor PC harus angka.")
        elif pilihan == '2' and role == 'admin':
            tanggal = input("Tanggal: ")
            read_laporan(tanggal)
            transaksi_list = list(data['transactions'].values())
            total_aktif = hitung_total_aktif(transaksi_list)
            print(f"Total transaksi: {total_aktif}")
        elif pilihan == '2' and role != 'admin':
            break
        elif pilihan == '3' and role == 'admin':
            try:
                update_transaksi(int(input("ID: ")), int(input("Durasi
baru: ")))
            except ValueError:
                print("Error: ID dan durasi harus angka.")
        elif pilihan == '4' and role == 'admin':
            try:
                delete_transaksi(int(input("ID: ")))
            except ValueError:
                print("Error: ID harus angka.")
        elif pilihan == '5' and role == 'admin':
            print("Anda telah logout.")
            role = None
        else:
            print("Pilihan tidak valid.")

        input("Tekan Enter")

if __name__ == "__main__":
    main_menu()

```

4. Hasil Output

Contoh hasil output jika termasuk admin

```
//Login
Selamat datang! Pilih: 1. Registrasi 2. Login 3. Keluar
Masukkan pilihan: 2
Masukkan username: admin
Masukkan password: 123

//Buat transaksi
1. Buat Transaksi 2. Laporan 3. Update 4. Hapus 5. Logout
Pilih: 1
Waktu mulai (YYYY-MM-DD): 2025-12-21
Durasi: 3
Nomor PC: 1
Transaksi berhasil dibuat ID: 1, Dengan Harga/Jam: Rp 12.00, Total Biaya
Menjadi: Rp 36000
Tekan Enter

//Laporan
1. Buat Transaksi 2. Laporan 3. Update 4. Hapus 5. Logout
Pilih: 2
Tanggal: 2025-12-21
Laporan Pendapatan untuk 2025-12-21: Rp 36000
ID: 1, Waktu: 2025-12-21, Total Biaya: Rp 36000
Total transaksi: 1
Tekan Enter

//Update
1. Buat Transaksi 2. Laporan 3. Update 4. Hapus 5. Keluar
Pilih: 3
ID: 1
Durasi baru: 6
Transaksi ID 1 diupdate.
Tekan Enter

//Delete
1. Buat Transaksi 2. Laporan 3. Update 4. Hapus 5. Keluar
Pilih: 4
ID: 1
Transaksi ID 1 dibatalkan.
Tekan Enter
```


Contoh hasil output jika termasuk user

```
//Register
Selamat datang! Pilih: 1. Registrasi 2. Login 3. Keluar
Masukkan pilihan: 1
Masukkan username baru: arya
Masukkan password: 123
Pengguna arya berhasil terdaftar sebagai user.

Masukkan username: arya
Masukkan password: 123

//Buat Transaksi
1. Buat Transaksi 2. Logout
Pilih: 1
Waktu mulai (YYYY-MM-DD): 2025-12-21
Durasi: 4
Nomor PC: 1
Transaksi berhasil dibuat ID: 1, Dengan Harga/Jam: Rp 12.00, Total Biaya
Menjadi: Rp 48000
Tekan Enter
```