

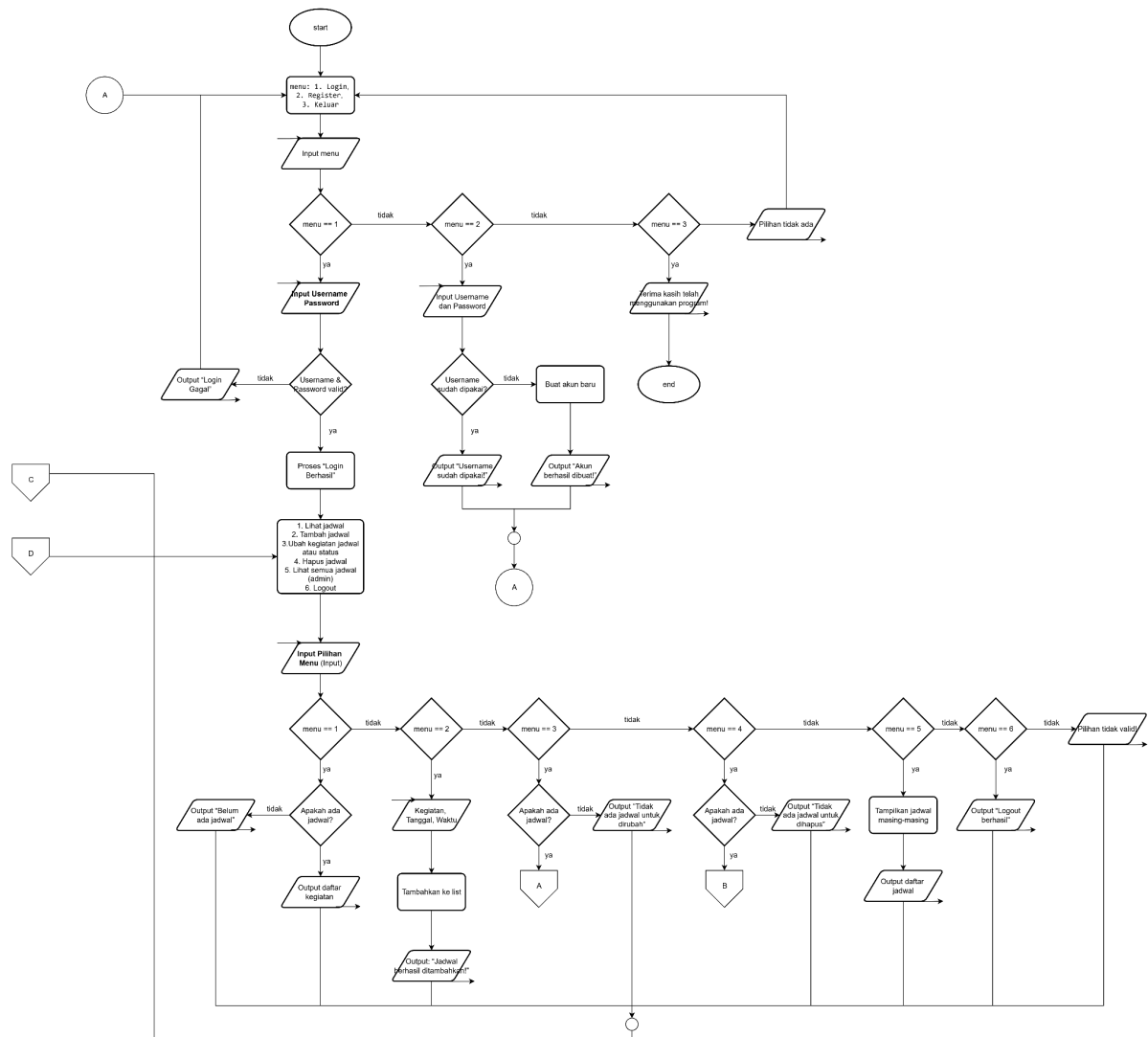
**LAPORAN PRAKTIKUM**  
**POSTTEST (7)**  
**ALGORITMA PEMROGRAMAN DASAR**



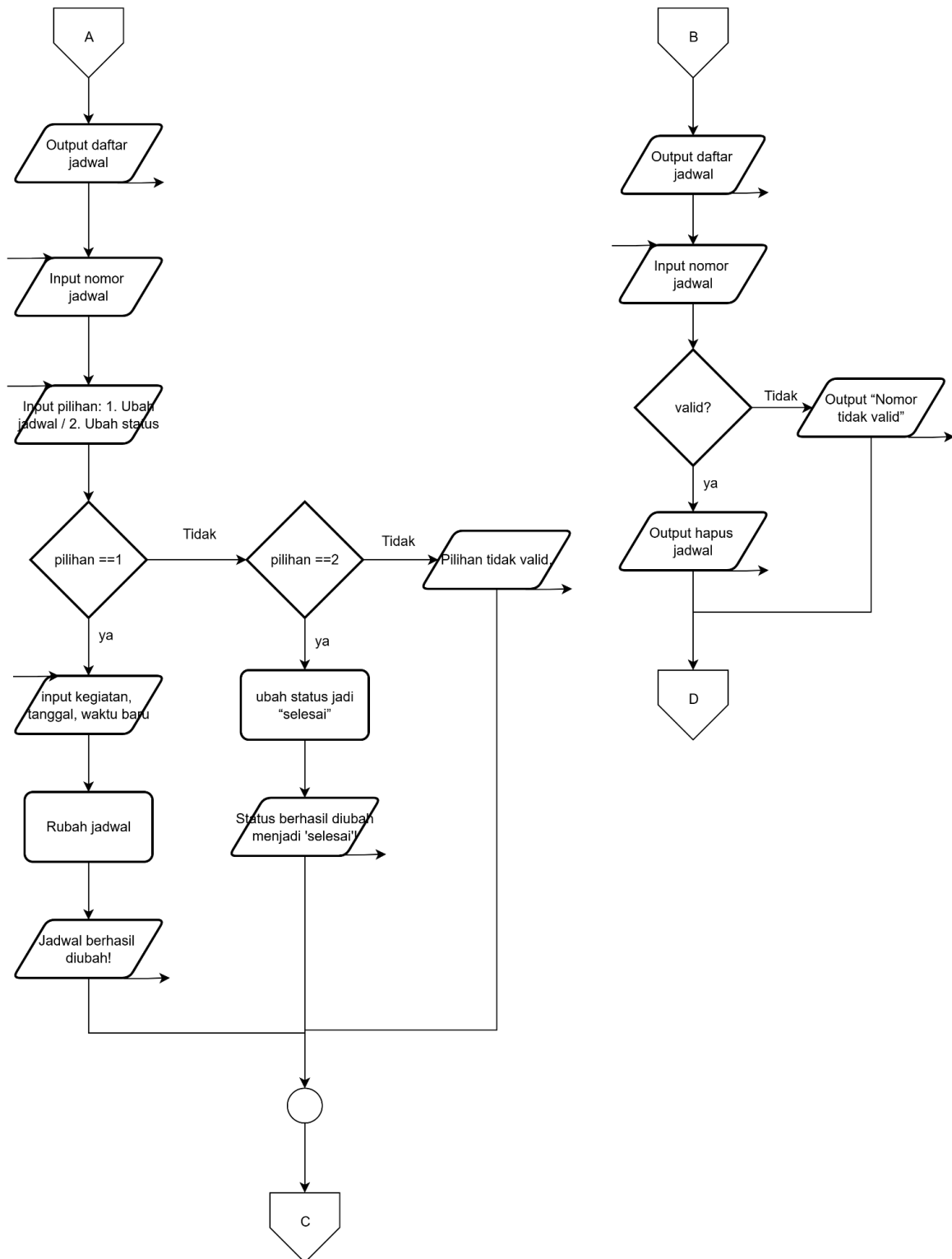
**Disusun oleh:**  
**Muhammad Zaki Fahriansyah (2509106020)**  
**Kelas (A'25)**

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

## 1. Flowchart



**Gambar 1.1 Flowchart Program Jadwal Aktivitas Digital**



**Gambar 1.2 Flowchart Program Jadwal Aktivitas Digital**

Penjelasan singkat :

Program “Jadwal Aktivitas Digital” digambarkan melalui flowchart yang menunjukkan alur login, registrasi, dan pengelolaan jadwal kegiatan. Program dimulai dari tampilan menu utama dengan tiga opsi: Login, Register, dan Keluar. Jika pengguna memilih Login, sistem memverifikasi akun admin. Bila berhasil, pengguna diarahkan ke menu utama pengguna yang berisi pilihan lihat jadwal, tambah jadwal, ubah jadwal atau status, hapus jadwal, dan logout.

Pada menu register, sistem memeriksa apakah username sudah ada atau belum. Jika belum, pengguna diminta membuat akun baru dan akan disimpan di dalam data. Di dalam menu pengguna, pengguna dapat menambah, melihat, mengubah, atau menghapus jadwal yang tersimpan (kurang lebih sama dengan admin). Admin memiliki tambahan fitur untuk melihat semua jadwal pengguna.

Terakhir pada menu keluar, maka program akan selesai dan menampilkan tulisan “Terima kasih telah menggunakan program!”. Sedangkan jika tidak memilih sesuai nomor pilihan akan menampilkan pilihan tidak ada.

## **2. Deskripsi Singkat Program**

Program ini adalah program Jadwal Aktivitas Digital. Program ini ada sistem login yang di mana bisa multi user (admin dan pengguna). Pertama program akan menampilkan pilihan login, register, dan keluar. Jika pengguna memilih login maka akun awal untuk login adalah akun admin dan akan menampilkan 5 pilihan, yang terdiri dari pilihan melihat jadwal, tambah jadwal, merubah jadwal atau merubah status, hapus jadwal, melihat semua jadwal akun (pengguna admin), dan terakhir log out. Pada menu melihat jadwal akan menampilkan jadwal jadwal kegiatan yang sudah kita tambahkan. Pada menu tambah jadwal, kita akan mengunggah atau menambahkan jadwal aktivitas kita yang terdiri dari nama kegiatan, tanggal, dan waktu. Lalu pada menu merubah jadwal atau merubah status untuk merubah jadwal aktivitas yang kita lakukan dr yang belum menjadi selesai. Pada menu melihat jadwal (pengguna akun admin), program akan melihatkan semua jadwal aktivitas akun akun yang di loginkan. Selanjutnya ada menu hapus jadwal yang digunakan untuk menghapus jadwal aktivitas yang tidak diinginkan. Terakhir ada menu logout yang akan menampilkan menu sebelumnya pada awal sebelum login.

Pada menu bagian awal jika pengguna memilih register, maka program akan membuatkan akun baru untuk pengguna, dan pengguna diminta untuk membuat username dan password baru. Selanjutnya setelah membuat akun program akan mengarahkan pengguna untuk login ulang dan bisa memilih beberapa menu seperti pada pengguna admin sebelumnya. Terakhir ada menu keluar maka akan keluar dari program.

### 3. Source Code

```
import os

akun = {
    "admin" : {"password" : "123", "role" : "admin"}
}
jadwal = {
    "admin" : []
}

ulang = True

def menu_login():
    user = input("Masukkan Username: ")
    pw = input("Masukkan Password: ")

    if user in akun and akun[user]["password"] == pw:
        print("Login Berhasil!")
        return user
    else:
        print("Login Gagal! Username atau Password Salah!")
        return None

def menu_register():
    print("=== REGISTER ===")
    user = input("Buat username: ")
    pw = input("Buat password: ")

    if user == '' or pw == '':
        print("Username dan Password tidak boleh kosong!")
    elif user in akun:
        print("Username sudah dipakai!")
    else:
        akun[user] = {"password": pw, "role": "user"}
        jadwal[user] = []
        print("Akun berhasil dibuat! Silahkan Login Kembali.")

    if tanya_lanjut():
        menu_register()
    else:
        print("Silakan login kembali.")
        input("Tekan Enter untuk melanjutkan...")

def tambah_jadwal(user, kegiatan, tanggal, waktu):
    if kegiatan == "" or tanggal == "" or waktu == "":
        print("Data tidak boleh kosong!")
    else:
        jadwal[user].append({
            "kegiatan": kegiatan,
            "tanggal": tanggal,
            "waktu": waktu,
            "status": "belum"
        })
        print("Jadwal berhasil ditambahkan!")

    if tanya_lanjut():
        kegiatan = input("Nama Kegiatan: ")
        tanggal = input("Tanggal (YYYY/MM/DD): ")
        waktu = input("Waktu (HH:MM): ")
        tambah_jadwal(user, kegiatan, tanggal, waktu)

def ubah_kegiatan_jadwal(user):
    lihat_jadwal(user)
    try:
        nomor = int(input("Pilih nomor jadwal: "))
```

```

print("1. Ubah Jadwal")
print("2. Ubah Status")
ubah = input("Pilih: ")

if ubah == "1":
    kegiatan = input("Masukkan kegiatan baru: ")
    tanggal = input("Masukkan tanggal baru (YYYY/MM/DD): ")
    waktu = input("Masukkan waktu baru (HH:MM): ")
    jadwal[user][nomor-1].update({
        "kegiatan": kegiatan,
        "tanggal": tanggal,
        "waktu": waktu
    })
    print("Jadwal berhasil diubah!")

    elif ubah == "2":
        ubah_status(jadwal[user], nomor-1)
except (ValueError, IndexError):
    print("Input tidak valid!")
if tanya_lanjut():
    ubah_kegiatan_jadwal(user)

def ubah_status(jadwal_user, index):
    try:
        jadwal_user[index]["status"] = "selesai"
        print("Status berhasil diubah menjadi 'selesai'!")
    except IndexError:
        print("Nomor tidak valid!")

def lihat_jadwal(user):
    print("=== JADWAL KAMU ===")
    if len(jadwal[user]) == 0:
        print("Belum ada jadwal.")
    else:
        no = 1
        for j in jadwal[user]:
            print(f"{no}. {j['kegiatan']} | {j['tanggal']} | {j['waktu']} | Status: {j['status']}")
            no += 1

def hapus_jadwal(user):
    lihat_jadwal(user)
    try:
        hapus = int(input("Pilih nomor jadwal yang ingin dihapus: "))
        if 1 <= hapus <= len(jadwal[user]):
            del jadwal[user][hapus-1]
            print("Jadwal berhasil dihapus!")
        else:
            print("Nomor tidak valid!")
    except ValueError:
        print("Input harus berupa angka!")
    if tanya_lanjut():
        hapus_jadwal(user)

def lihat_semuaJadwal(user):
    print("=== SEMUA JADWAL PENGGUNA ===")
    for username, data in jadwal.items():
        print(f"\nUser: {username}")
        if not data:
            print(" (Belum ada jadwal)")
        else:
            no = 1
            for j in data:
                print(f"{no}. {j['kegiatan']} | {j['tanggal']} | {j['waktu']} | Status: {j['status']}")
                no += 1

def menu_1_login(user, role):
    lanjut = True
    while lanjut:

```

```

os.system('cls||clear')
print(f"\n=== MENU {user.upper()} ===")
print("1. Lihat Jadwal")
print("2. Tambah Jadwal")
print("3. Ubah Kegiatan Jadwal Atau Status")
print("4. Hapus Jadwal")
if role == "admin":
    print("5. Lihat Semua Jadwal")
    print("6. Logout")
else:
    print("5. Logout")
pilih = input("Pilih menu: ")

if user not in jadwal:
    jadwal[user] = []

if pilih == "1":
    os.system('cls||clear')
    lihat_jadwal(user)

elif pilih == "2":
    os.system('cls||clear')
    kegiatan = input("Nama Kegiatan: ")
    tanggal = input("Tanggal (YYYY/MM/DD): ")
    waktu = input("Waktu (HH:MM): ")
    tambah_jadwal(user, kegiatan, tanggal, waktu)

elif pilih == "3":
    os.system('cls||clear')
    ubah_kegiatan_jadwal(user)

elif pilih == "4":
    os.system('cls||clear')
    hapus_jadwal(user)

elif pilih == "5" and role == "admin":
    os.system('cls||clear')
    lihat_semuaJadwal(user)

elif (pilih == "5" and role != "admin") or (pilih == "6" and role == "admin"):
    print("Logout berhasil.")
    lanjut = False

else:
    print("Pilihan tidak valid!")
    input("\nTekan Enter untuk melanjutkan...")

def tanya_lanjut():
    pilih = input("\nApakah ingin melakukan lagi? (y/n)? ").lower()
    if pilih == 'y':
        return True
    elif pilih == 'n':
        return False
    else:
        print("Input tidak valid, coba lagi.")
        return tanya_lanjut()

while ulang:
    os.system('cls||clear')
    print("=== JADWAL AKTIVITAS DIGITAL ===")
    print("1. Login")
    print("2. Register")
    print("3. Keluar")
    menu = input("Pilih menu: ")

    if menu == "1":
        user = menu_login()
        if user is None:

```

```
        input("Tekan Enter untuk lanjut...")
        continue

    role = akun[user]["role"]
    menu_1_login(user, role)

elif menu == "2":
    menu_register()

elif menu == "3":
    ulang = False
    print("Terima kasih telah menggunakan program!")
else:
    print("Pilihan tidak ada!")
    input("Enter...")
```

*Gambar 3.1 Source Code*



#### 4. Hasil Output

1. Tampilan menu jika pengguna login sebagai admin.

```
=== MENU ADMIN ===  
1. Lihat Jadwal  
2. Tambah Jadwal  
3. Ubah Kegiatan Jadwal Atau Status  
4. Hapus Jadwal  
5. Lihat Semua Jadwal  
0. Logout  
Pilih menu: █
```

*Gambar 4.1 Hasil Output login admin*

2. Tampilan register akun dan buat akun baru/user.

```
=== JADWAL AKTIVITAS DIGITAL ===  
1. Login  
2. Register  
3. Keluar  
Pilih menu: 2  
=== REGISTER ===  
Buat username: zaki  
Buat password: 020  
Akun berhasil dibuat! Silahkan Login Kembali.  
Tekan Enter untuk melanjutkan...█
```

*Gambar 4.2 Hasil Output register akun baru/user*

3. Tampilan menu jika pengguna login sebagai user.

```
=== MENU ZAKI ===  
1. Lihat Jadwal  
2. Tambah Jadwal  
3. Ubah Kegiatan Jadwal Atau Status  
4. Hapus Jadwal  
0. Logout  
Pilih menu: █
```

*Gambar 4.3 Hasil Output login user*

4. Jika pengguna memilih menu lihat jadwal tetapi belum ada jadwal yang ditambahkan dan sudah menambahkan jadwal.

```
=== JADWAL KAMU ===  
Belum ada jadwal.  
  
Tekan Enter untuk melanjutkan...
```

*Gambar 4.4 Hasil Output melihat jadwal*

```
=== JADWAL KAMU ===  
1. Membuat akun | 2025/10/14 | 19:00 | Status: belum  
  
Tekan Enter untuk melanjutkan...
```

*Gambar 4.5 Hasil Output melihat jadwal*

5. Pengguna memilih menu tambah jadwal.

```
=== MENU ADMIN ===  
1. Lihat Jadwal  
2. Tambah Jadwal  
3. Ubah Kegiatan Jadwal Atau Status  
4. Hapus Jadwal  
5. Lihat Semua Jadwal  
0. Logout  
Pilih menu: 2  
Nama Kegiatan: membuat akun  
Tanggal (YYYY/MM/DD): 2025  
Waktu (HH:MM): 19:00  
Jadwal berhasil ditambahkan!  
  
Tekan Enter untuk melanjutkan...
```

*Gambar 4.6 Hasil Output daftar menu belanjaan*

6. Pengguna memilih menu ubah kegiatan jadwal atau status.

```
=== MENU ADMIN ===  
1. Lihat Jadwal  
2. Tambah Jadwal  
3. Ubah Kegiatan Jadwal Atau Status  
4. Hapus Jadwal  
5. Lihat Semua Jadwal  
6. Logout  
Pilih menu: 3  
1. Membuat akun | 2025 | 19:00 | Status: belum  
Pilih nomor jadwal: 1  
1. Ubah Jadwal  
2. Ubah Status  
Pilih: 2  
Status berhasil diubah menjadi 'selesai'  
  
Tekan Enter untuk melanjutkan...
```

*Gambar 4.7 Hasil output merubah status*

```

=== MENU ADMIN ===
1. Lihat Jadwal
2. Tambah Jadwal
3. Ubah Kegiatan Jadwal Atau Status
4. Hapus Jadwal
5. Lihat Semua Jadwal
6. Logout
Pilih menu: 3
1. Membuat akun | 2025 | 19:00 | Status: selesai
Pilih nomor jadwal: 1
1. Ubah Jadwal
2. Ubah Status
Pilih: 1
Masukkan kegiatan baru: Makan
Masukkan tanggal baru (YYYY/MM/DD): 2025/10/14
Masukkan waktu baru (HH:MM): 20:00
Jadwal berhasil diubah!

Tekan Enter untuk melanjutkan...

```

*Gambar 4.8 Hasil output merubah jadwal*

7. Tampilan pengguna memilih menu hapus jadwal.

```

=== MENU ADMIN ===
1. Lihat Jadwal
2. Tambah Jadwal
3. Ubah Kegiatan Jadwal Atau Status
4. Hapus Jadwal
5. Lihat Semua Jadwal
6. Logout
Pilih menu: 4
1. Makan | 2025 | 20:00 | Status: selesai
Pilih nomor jadwal yang ingin dihapus: 1
Jadwal berhasil dihapus!

Tekan Enter untuk melanjutkan...

```

*Gambar 4.9 Hasil Output menghapus jadwal*

8. Tampilan memilih menu logout

```

=== MENU ADMIN ===
1. Lihat Jadwal
2. Tambah Jadwal
3. Ubah Kegiatan Jadwal Atau Status
4. Hapus Jadwal
5. Lihat Semua Jadwal
6. Logout
Pilih menu: 6
Logout berhasil.

Tekan Enter untuk melanjutkan...

```

*Gambar 4.10 Hasil Output logout*

9. Tampilan memilih menu lihat semua jadwal pengguna akun.

```
=== MENU ADMIN ===
1. Lihat Jadwal
2. Tambah Jadwal
3. Ubah Kegiatan Jadwal Atau Status
4. Hapus Jadwal
5. Lihat Semua Jadwal
6. Logout
Pilih menu: 5
=== SEMUA JADWAL PENGGUNA ===

User: admin
  1. makan | 2025/10/14 | 20:00 | Status: selesai

User: zaki
  1. Mengerjakan POSTTEST | 2025/10/14 | 21:00 | Status: selesai
  2. Makan | 2025/10/14 | 21:30 | Status: selesai

Tekan Enter untuk melanjutkan...
```

*Gambar 4.11 Hasil Output Tampilan Semua Jadwal*

10. Tampilan jika pilihan pilihan angka yang dipilih tidak sesuai.

```
Pilihan tidak valid!

Tekan Enter untuk melanjutkan...
```

*Gambar 4.12 Hasil Output tidak valid*

## 5. Langkah-langkah GIT

### 5.1 GIT Add

```
PS C:\Kuliah Infor\praktikum-apd> git add .
```

*Gambar 5.1 GIT Add*

Karena sebelumnya sudah di upload jadi langsung ke GIT Add yang fungsinya untuk menyiapkan perubahan file supaya siap dicatat di commit. Lalu untuk GIT add . untuk menambahkan semua file.

### 5.2 GIT Commit

```
PS C:\Kuliah Infor\praktikum-apd> git commit -m "PT 7"
[main 5035763] PT 7
 2 files changed, 379 insertions(+)
 create mode 100644 kelas/Pertemuan 7/Pertemuan7.py
 create mode 100644 post-test/post-test-apd-7/2509106020-Muhammad Zaki Fahriansyah-PT-7.py
```

*Gambar 5.2 GIT Commit*

Selanjutnya ada GIT Commit untuk menyimpan perubahan yang sudah di-add ke dalam repository.

### 5.3 GIT Push

```
PS C:\Kuliah Infor\praktikum-apd> git push
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 2 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (8/8), 3.27 KiB | 238.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/zakifahriansyah-svg/praktikum-apd.git
 a809f2f..5035763 main -> main
```

*Gambar 5.3 GIT Push*

GIT Push dilakukan agar file-file yang telah diubah atau dimodif akan masuk kedalam repository GITHUB kita sehingga saat melakukan pull maka sudah akan terupdate menjadi file yang telah diperbarui atau dimodif, disini menggunakan git push tanpa origin main.