

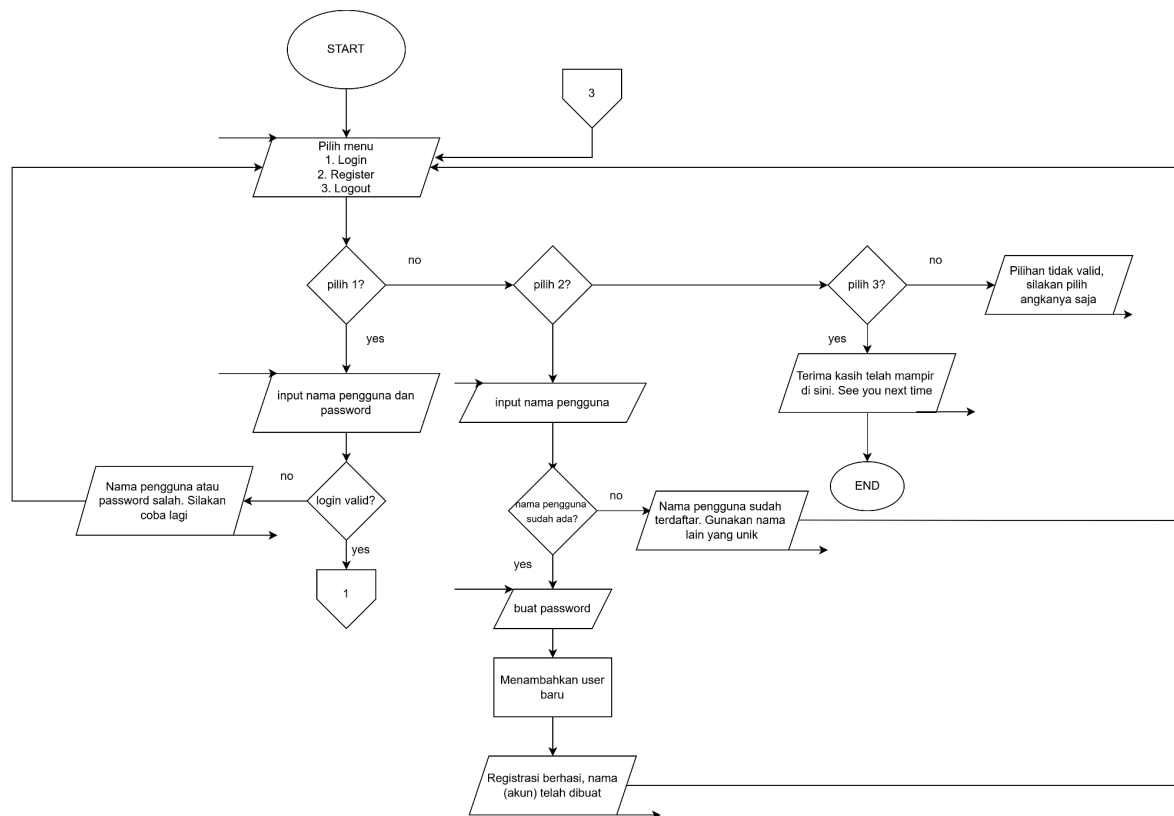
LAPORAN PRAKTIKUM
POST TEST 7
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Syarafina Nur Amalia (2509106016)
Kelas A'25

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart

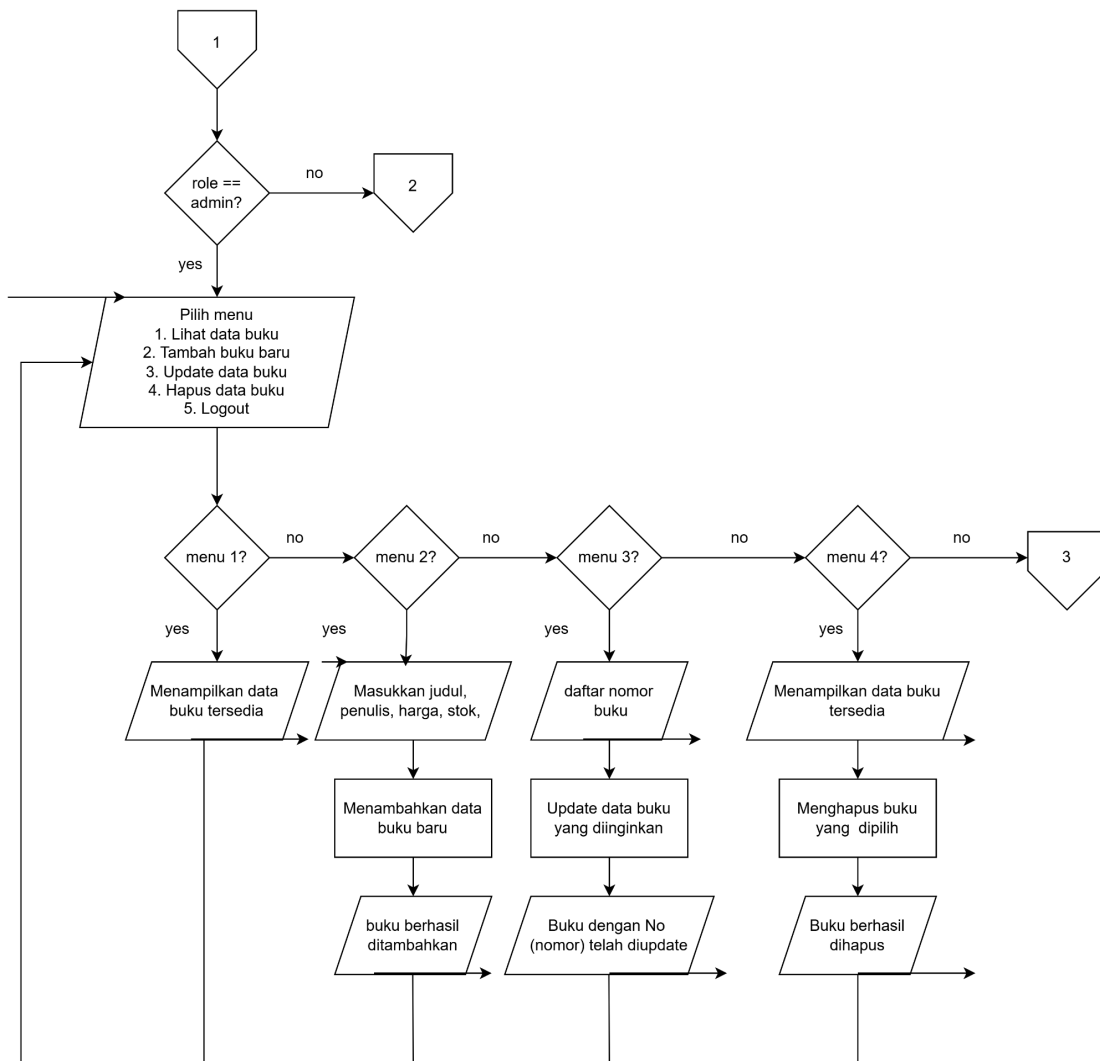


1.1 flowchart login dan register

Pada saat dimulai, program akan menampilkan 3 menu utama, yaitu *login*, *register*, dan *logout*. *User* akan diminta untuk memilih salah satu dari ketiga menu yang tersedia.

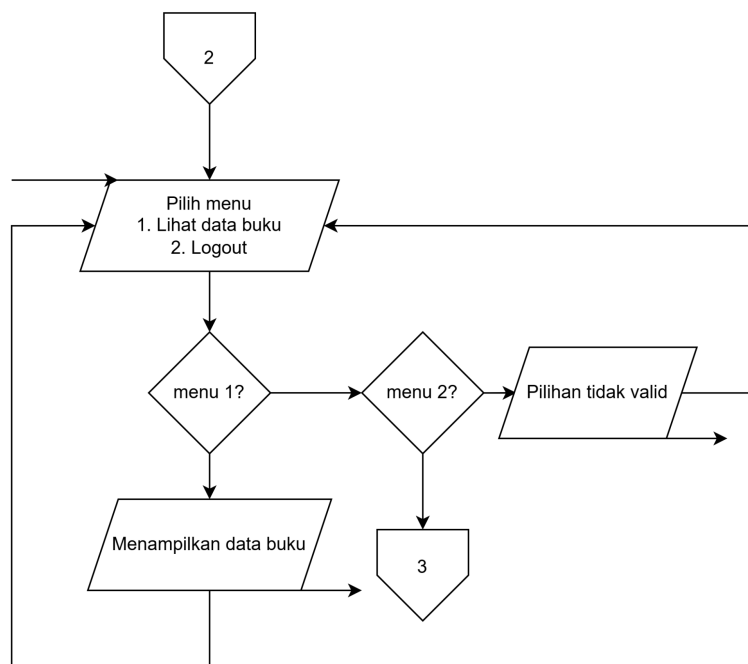
Jika *user* memilih angka 1, maka program akan meminta *user* untuk memasukkan nama pengguna dan *password* yang telah dimiliki. Program akan melakukan pengecekan terhadap nama pengguna dan *password*, jika benar, program akan lanjut ke menu selanjutnya. Jika nama pengguna dan *password* *invalid*, maka *user* akan kembali ke menu utama.

Jika *user* memilih angka 2, maka program akan meminta *user* untuk memasukkan nama pengguna. Program akan melakukan pengecekan terhadap nama pengguna yang dimasukkan, apakah sudah terdaftar atau belum. Jika belum terdaftar, maka program akan meminta *user* untuk memasukkan *password*. Jika nama pengguna sudah tersedia, maka *user* akan diminta untuk mengganti nama pengguna. Setelah selesai, program akan kembali ke menu utama dan *user* yang memilih *register* dapat *login* dengan nama pengguna dan *password* yang baru saja dibuat.



1.2 Flowchart menu admin

Setelah *login*, program akan memeriksa *role* terlebih dahulu. Jika *role user* adalah admin, maka program akan menampilkan menu admin yang terdiri dari 5 menu. Pada menu 1, *user* hanya akan melihat list buku yang tersedia. Pada menu 2, *user* dapat menambahkan data buku baru pada program. Pada menu 3, *user* dapat mengubah data buku yang telah ada. Pada menu 4, *user* dapat menghapus data buku yang tersedia. Setelah memilih, *user* akan dibawa kembali ke menu utama.



1.3 Flowchart menu user

Jika pada pengecekan sebelumnya ternyata *role user* bukan admin, maka program akan memberikan hanya 2 pilihan menu saja. Jika memilih menu 1, maka program akan menampilkan data buku yang tersedia. Jika memilih 2, maka program akan kembali ke menu utama, begitu juga jika *user* telah selesai dengan menu 1.

2. Deskripsi Singkat Program

Program ini dibuat untuk mendaftarkan setiap buku yang ada di dalam sebuah toko buku yang bernama “Toko Buku Senkumedia.” Dalam program ini, kita dapat menambah, mengubah, melihat, maupun menghapus data yang ada. Program ini juga dilengkapi dengan fitur admin serta user dengan tampilan menu yang berbeda untuk setiap *role*-nya.

3. Source Code

3.1 Source Code (data.py)

```
users = {
    'admin' : {'password': 'admin123', 'role': 'admin'},
    'fina' : {'password': '016', 'role': 'admin'},
    'senku' : {'password': '100billion', 'role': 'admin'},
    'kohaku' : {'password': 'power', 'role': 'user'},
    'denji' : {'password': 'chainsaw', 'role': 'user'},
}
buku_sementara = {
    1: {
        'Judul' : 'Laut Bercerita',
        'Penulis' : 'Leila S. Chudori',
        'Harga' : 128000,
        'Stok' : 18,
        'Kategori' : 'Fiksi'
    },
    2: {
        'Judul' : 'Namaku Alam',
        'Penulis' : 'Leila S.',
        'Harga' : 157000,
        'Stok' : 10,
        'Kategori' : 'Sejarah'
    },
    3: {
        'Judul' : 'Bumi Manusia',
        'Penulis' : 'Pramoedya Ananta Toer',
        'Harga' : 231000,
        'Stok' : 12,
        'Kategori' : 'Fantasi'
    },
    4: {
        'Judul' : 'Dunia Sophie',
        'Penulis' : 'Jostein Gaarder',
        'Harga' : 267000,
        'Stok' : 16,
        'Kategori' : 'Petualangan'
    },
}
kategori_buku = ['Fiksi', 'Non-Fiksi', 'Ilmiah', 'Petualangan', 'Fantasi',
```

```
'Sejarah']
```

Ini adalah code yang menyimpan dictionary untuk berjalan di program utama.

3.2 Source code (main)

```
def login(username, password):  
    for key, value in users.items():  
        if key == username and value['password'] == password:  
            return (True, key, value['role'])  
    return (False, None, None)
```

3.2.1 Menu login

Pada *code* di atas, saya menggunakan fungsi dengan parameter untuk menu login *user*, dimana program akan melakukan pengecekan dengan menggunakan *key* dan *value* agar *user* dapat lanjut ke menu selanjutnya. *Input* yang dilakukan *user* harus sesuai dengan yang ada di dalam *dictionary* pada *file* sebelumnya.

```
def register():  
    global users  
    clear()  
    print("\n--- REGISTER PENGGUNA BARU ---")  
  
    while True:  
        pengguna_baru = input("Nama pengguna baru: ").strip()  
        if not teks_tidak_kosong(pengguna_baru):  
            print("Nama pengguna tidak boleh kosong.")  
        elif pengguna_baru in users:  
            print("Nama pengguna sudah terdaftar. Gunakan nama lain yang unik.")  
        else:  
            break  
  
        password_baru = input_valid("Password baru: ", teks_tidak_kosong, "Password  
tidak boleh kosong. Buatlah password yang kuat.")  
  
        users[pengguna_baru] = {'password': password_baru, 'role': 'user'}  
        print(f"Registrasi berhasil! Akun dengan nama '{pengguna_baru}' telah  
dibuat.")  
        input("Tekan enter untuk kembali...")
```

3.2.2 Menu Register user

Code di atas merupakan bagian *register*; *user* akan diminta untuk membuat nama pengguna dan *password* baru agar bisa *login*. Program juga akan mengecek jika nama pengguna sudah

terdaftar atau belum, jika sudah terdaftar, *user* akan diminta untuk membuat nama pengguna yang baru.

```
if __name__ == "__main__":

    clear()
    while program_berjalan:

        if not sudah_login:
            menu_utama = menu_utama_admin()

            if menu_utama == '1':
                print("\n--- LOGIN ---")
                input_nama_pengguna = input("Nama pengguna: ").strip()
                input_password = input("Password: ").strip()

                login_berhasil, user_id, user_role = login(input_nama_pengguna,
input_password)

                if login_berhasil:
                    sudah_login = True
                    user_saat_ini = user_id
                    role = user_role
                    print(f"\nSelamat datang, {user_saat_ini}")
                else:
                    print("Nama pengguna atau password salah. Silakan coba
lagi.")

            elif menu_utama == '2':
                register()

            elif menu_utama == '3':
                print("\nTerima kasih telah mampir di sini. See you next time.")
                program_berjalan = False

            else:
                print("Pilihan tidak valid. Silakan pilih angkanya saja (1, 2,
atau 3).")
```

3.2.3 Menu utama user

Code di atas merupakan menu utama user yang terdiri lagi 3 bagian, yaitu *login*, *register*, dan *logout* jika *user* belum melakukan *login*. Jika *user* memilih angka 1, maka program akan menjalankan fungsi *login* dan meminta *user* untuk memasukkan *username* serta *password*. Jika *user* memilih angka 2, maka program akan menjalankan fungsi *register*. Jika *user* memilih angka 3, maka program akan berhenti dengan menjalankan perintah (program

berjalan = *False*) dan mengeluarkan output “Terima kasih telah mampir di sini. See you next time.” yang akan langsung mengakhiri program.

```
elif sudah_login:
    if role == 'admin':
        pilihan_admin = menu_admin_login()

        if pilihan_admin == '1': #Read admin
            print("\n--- DAFTAR BUKU SEKARANG ---")
            daftar_buku(buku_sementara)
        elif pilihan_admin == '2': #Create admin
            tambah_buku()
        elif pilihan_admin == '3': #Update admin
            update_buku()
        elif pilihan_admin == '4': #Delete admin
            hapus_buku()
        elif pilihan_admin == '5': #Logout admin
            sudah_login = False
            user_saat_ini = None
            role = None
            print("Anda berhasil logout.")
        else:
            print("Pilihan tidak valid.")

    elif role == 'user':
        clear()
        tabel_user = PrettyTable()
        tabel_user.field_names = ["No", "Menu"]
        tabel_user.align = "L"
        tabel_user.add_row(["1", "Lihat daftar buku tersedia (Read)"])
        tabel_user.add_row(["2", "Logout"])
        print(f"\n=== MENU USER ===")
        print(tabel_user)
        pilihan_user = input("Pilih menu (nomor): ").strip()

        if pilihan_user == '1': # Read User
            print("\n--- DAFTAR BUKU TERSEDIA ---")
            daftar_buku(buku_sementara)
        elif pilihan_user == '2': # Logout User
            sudah_login = False
            user_saat_ini = None
            role = None
            print("Anda berhasil logout.")
        else:
            print("Pilihan tidak valid.")
```

3.2.4 Menu admin dan user biasa

Jika *user* sudah melakukan *login* dan mendapatkan *role admin*, maka akan muncul 5 menu yang dapat dipilih. Jika *user* mendapatkan *role* sebagai *user* biasa, maka hanya ada 2 menu yang ditampilkan.

```
def daftar_buku(data_buku):
    tabel_buku = PrettyTable()
    tabel_buku.field_names = ["NO", "Judul", "Penulis", "Harga (Rp)", "Stok",
                              "Kategori"]

    if not data_buku:
        print("Tidak ada data buku yang tersedia.")
        return

    if role == 'user':
        buku_tampil = {key: value for key, value in data_buku.items() if
                        value['Stok'] > 0}
    else:
        buku_tampil = data_buku

    if not buku_tampil and role == 'user':
        print("Maaf, buku yang Anda cari tidak tersedia.")
        return

    for key, value in buku_tampil.items():
        format_harga = f"{value['Harga']:,}".replace(",", ".")
        tabel_buku.add_row([key, value['Judul'], value['Penulis'], format_harga,
                           value['Stok'], value['Kategori']])

    tabel_buku.align["NO"] = "r"
    tabel_buku.align["Judul"] = "l"
    tabel_buku.align["Penulis"] = "l"
    tabel_buku.align["Harga (Rp)"] = "r"
    tabel_buku.align["Stok"] = "r"
    tabel_buku.align["Kategori"] = "l"

    print(tabel_buku)
    print("\nTekan enter untuk melanjutkan...")
    input()
```

3.2.5 Menu read admin dan user biasa

Pada bagian ini, program akan menampilkan menu *read*, baik pada *admin* maupun *user* biasa. Jika Anda merupakan *user* biasa, maka buku yang ditampilkan adalah buku dengan stok lebih dari 0. Tetapi, jika Anda merupakan *admin*, maka semua data akan ditampilkan.

```
def tambah_buku():
    global no_buku_selanjutnya
```

```

clear()
print("\n--- TAMBAH BUKU BARU ---")

judul_baru = input_valid("Judul: ", teks_tidak_kosong, "Judul tidak boleh kosong.")
penulis_baru = input_valid("Penulis: ", teks_tidak_kosong, "Penulis tidak boleh kosong.")

harga_teks = input_valid("Harga (contoh: 150000): ", angka_positif, "Input harga tidak valid. Isi dengan angka positif.")
harga_baru = int(harga_teks)

stok_teks = input_valid("Stok: ", angka_positif_stok, "Input stok tidak valid. Isi dengan angka positif atau nol.")
stok_baru = int(stok_teks)

#Kategori
print("\n--- PILIH KATEGORI BUKU ---")
tabel_kategori = PrettyTable()
tabel_kategori.field_names = ["NO", "Kategori"]
for i, kategori in enumerate(kategori_buku):
    tabel_kategori.add_row([i+1, kategori])
print(tabel_kategori)

#Input kategori dengan nilai positif
pilihan_kategori_teks = input_valid("Pilih kategori (nomor): ", kategori_buku_positif, "Pilihan kategori tidak valid.")
pilihan_kategori = int(pilihan_kategori_teks)
kategori_baru = kategori_buku[pilihan_kategori - 1]

buku_sementara[no_buku_selanjutnya] = {
    'Judul': judul_baru,
    'Penulis': penulis_baru,
    'Harga': harga_baru,
    'Stok': stok_baru,
    'Kategori': kategori_baru
}
print(f"\nBuku '{judul_baru}' berhasil ditambahkan dengan NO {no_buku_selanjutnya}.")
no_buku_selanjutnya += 1
input("Tekan enter untuk kembali...")

```

3.2.6 Menu admin create

Code di atas akan menjalankan menu *create* pada *role admin*, dimana judul, penulis, harga, dan stok tidak boleh kosong yang dijalankan dengan fungsi *input valid*. Sedangkan, bagian kategori akan ditampilkan dalam bentuk tabel dan akan meminta *user* untuk memilih angka yang tersedia.

```

def update_buku():
    clear()
    print("\n--- UPDATE DATA BUKU ---")
    daftar_buku(buku_sementara)

    if not buku_sementara:
        return

    buku_ditemukan = None
    while buku_ditemukan is None:
        no_update_buku = input_valid("\nMasukkan NO buku yang akan diubah: ",
angka_positif, "Input NO tidak valid. Masukkan angka positif.")
        no_update = int(no_update_buku)

        try:
            buku_ditemukan = buku_sementara[no_update]
        except KeyError:
            print(f"Buku dengan NO {no_update} tidak ditemukan.")

    print("\n(Kosongkan jika tidak ada data yang ingin diubah)")
    data_diubah = False

    #Update judul
    isi_judul_baru = input(f"Judul baru ({buku_ditemukan['Judul']}): ").strip()
    if teks_tidak_kosong(isi_judul_baru):
        buku_ditemukan['Judul'] = isi_judul_baru
        data_diubah = True

    #Update penulis
    isi_penulis_baru = input(f"Penulis baru ({buku_ditemukan['Penulis']}): ").strip()
    if teks_tidak_kosong(isi_penulis_baru):
        buku_ditemukan['Penulis'] = isi_penulis_baru
        data_diubah = True

    #Update harga
    while True:
        harga_lama = f"{buku_ditemukan['Harga']:,}".replace(",", ".")
        harga_baru_teks = input(f"Harga baru ({harga_lama}): ").strip()
        if harga_baru_teks == "":
            break
        if angka_positif(harga_baru_teks):
            buku_ditemukan['Harga'] = int(harga_baru_teks)
            data_diubah = True
            break
        print("Masukkan angka positif.")

    #Update stok
    while True:
        stok_baru_teks = input(f"Stok baru ({buku_ditemukan['Stok']}): ").strip()

```

```

    if stok_baru_teks == "":
        break
    if angka_positif_stok(stok_baru_teks):
        buku_ditemukan['Stok'] = int(stok_baru_teks)
        data_diubah = True
        break
    print("Masukkan angka positif.")

#Update kategori
print("\n--- PILIH KATEGORI BARU ---")
tabel_kategori = PrettyTable()
tabel_kategori.field_names = ["NO", "Kategori"]
for i, kategori in enumerate(kategori_buku):
    tabel_kategori.add_row([i+1, kategori])
print(tabel_kategori)

pilihan_kategori_teks = input(f"Pilih nomor kategori baru
({buku_ditemukan['Kategori']}): ").strip()

if pilihan_kategori_teks != "":
    if kategori_buku_positif(pilihan_kategori_teks):
        pilihan_kategori = int(pilihan_kategori_teks)
        buku_ditemukan['Kategori'] = kategori_buku[pilihan_kategori - 1]
        data_diubah = True
    else:
        print("Pilihan kategori tidak valid, kategori tidak diubah.")

if data_diubah:
    print(f"\nBuku dengan NO {no_update} berhasil diperbarui.")
else:
    print(f"\nTidak ada perubahan data untuk buku NO {no_update}.")
input("Tekan enter untuk kembali...")

```

3.2.7 Menu admin update

Code di atas akan menjalankan fungsi *update* dengan memasukkan nomor buku yang akan diubah. *User* diminta untuk memasukkan data yang baru ke dalam program, jika terdapat data yang tidak ingin diubah, *user* dapat mengosongkan bagian tersebut dengan *enter*.

```

def hapus_buku():
    clear()
    print("\n--- HAPUS BUKU ---")
    daftar_buku(buku_sementara)

    if not buku_sementara:
        return

    buku_dihapus = None

```

```

indeks_hapus = -1

while buku_dihapus is None:
    nomor_hapus_teks = input_valid("\nMasukkan NO buku yang akan dihapus: ",
angka_positif, "Input NO tidak valid. Masukkan angka positif.")
    nomor_buku_dihapus = int(nomor_hapus_teks)

    try:
        buku_dihapus = buku_sementara[nomor_buku_dihapus]['Judul']
        indeks_hapus = nomor_buku_dihapus
    except KeyError:
        print(f"Buku {nomor_buku_dihapus} tidak ditemukan.")

#konfirmasi buku dihapus
konfirmasi = input_valid(f"Apakah yakin ingin menghapus buku
'{buku_dihapus}' (y/n)? : ", konfirmasi_delete_buku, "Konfirmasi hanya 'y' atau
'n'.")

if konfirmasi.lower() == 'y':
    try:
        buku_sementara.pop(indeks_hapus)
        print(f"\nBuku '{buku_dihapus}' berhasil dihapus.")
    except KeyError:
        print("Buku tidak ditemukan.")
else:
    print("Penghapusan dibatalkan.")
input("Tekan enter untuk kembali...")

```

3.2.8 Menu admin delete

Code di atas merupakan fungsi *delete* yang hanya bisa dijalankan oleh *admin*. Program akan menampilkan semua data buku lalu meminta *admin* untuk memilih data buku mana yang akan dihapus. Program juga akan melakukan konfirmasi ulang apakah *admin* yakin untuk menghapus data buku yang dipilih. Program juga menggunakan *error handling* dengan *try except* untuk memastikan angka yang diinput *admin* benar-benar ada.

4. Hasil Output

```
=== SELAMAT DATANG DI TOKO BUKU SENKUMEDIA ===
+-----+-----+
| No | Menu      |
+-----+-----+
| 1  | Login     |
| 2  | Register  |
| 3  | Logout    |
+-----+-----+
Silakan pilih menu yang tersedia: 1

--- LOGIN ---
Nama pengguna: fina
Password: 016
```

Gambar 4.1 Login user

```
=== MENU ADMIN ===
+-----+-----+
| NO | Menu      |
+-----+-----+
| 1  | Lihat data buku (Read) |
| 2  | Tambah buku baru (Create) |
| 3  | Update data buku (Update) |
| 4  | Hapus data buku (Delete) |
| 5  | Logout      |
+-----+-----+
Pilih menu Admin: 1

--- DAFTAR BUKU SEKARANG ---
+-----+-----+-----+-----+-----+-----+
| NO | Judul      | Penulis          | Harga (Rp) | Stok | Kategori |
+-----+-----+-----+-----+-----+-----+
| 1  | Laut Bercerita | Leila S.        | 128.000    | 18  | Novel    |
| 2  | Namaku Alam   | Leila S.        | 157.000    | 10  | Novel    |
| 3  | Bumi Manusia  | Pramoedya Ananta Toer | 231.000    | 12  | Novel    |
| 4  | Dunia Sophie  | Jostein Gaarder  | 267.000    | 16  | Novel    |
+-----+-----+-----+-----+-----+-----+
```

Gambar 4.2 Login admin menu 1

```
--- TAMBAH BUKU BARU ---
Judul: The Setting Sun
Penulis: Osamu Dazai
Harga (contoh: 150000): 145000
Stok: 14

--- PILIH KATEGORI BUKU ---
+---+-----+
| NO | Kategori |
+---+-----+
| 1 | Fiksi |
| 2 | Non-Fiksi |
| 3 | Ilmiah |
| 4 | Petualangan |
| 5 | Fantasi |
| 6 | Sejarah |
+---+-----+
Pilih kategori (nomor): 1
```

Gambar 4.3 Login admin menu 2

```
+---+-----+-----+-----+-----+-----+
| NO | Judul | Penulis | Harga (Rp) | Stok | Kategori |
+---+-----+-----+-----+-----+-----+
| 1 | Laut Bercerita | Leila S. Chudori | 128.000 | 18 | Fiksi |
| 2 | Namaku Alam | Leila S. | 157.000 | 10 | Sejarah |
| 3 | Bumi Manusia | Pramoedya Ananta Toer | 231.000 | 12 | Fantasi |
| 4 | Dunia Sophie | Jostein Gaarder | 267.000 | 16 | Petualangan |
+---+-----+-----+-----+-----+-----+

Masukkan NO buku yang akan diubah: 2

(Kosongkan jika tidak ada data yang ingin diubah)
Judul baru (Namaku Alam):
Penulis baru (Leila S.): Leila S. Chudori
Harga baru (157.000): 157000
Stok baru (10): 14

--- PILIH KATEGORI BARU ---
+---+-----+
| NO | Kategori |
+---+-----+
| 1 | Fiksi |
| 2 | Non-Fiksi |
| 3 | Ilmiah |
| 4 | Petualangan |
| 5 | Fantasi |
| 6 | Sejarah |
+---+-----+
Pilih nomor kategori baru (1-6): 2
```

Gambar 4.4 Login admin menu 3

```

--- HAPUS BUKU ---
+-----+-----+-----+-----+-----+-----+
| NO | Judul          | Penulis          | Harga (Rp) | Stok | Kategori |
+-----+-----+-----+-----+-----+-----+
| 1 | Laut Bercerita | Leila S. Chudori | 128.000    | 18  | Fiksi    |
| 2 | Namaku Alam    | Leila S.         | 157.000    | 10  | Sejarah  |
| 3 | Bumi Manusia   | Pramoedya Ananta Toer | 231.000    | 12  | Fantasi  |
| 4 | Dunia Sophie   | Jostein Gaarder  | 267.000    | 16  | Petualangan |
| 5 | The Setting Sun | Osamu Dazai      | 145.000    | 14  | Fiksi    |
+-----+-----+-----+-----+-----+-----+

Masukkan NO buku yang akan dihapus: 2
Anda yakin ingin menghapus buku 'Namaku Alam' (y/n)? : y

```

Gambar 4.5 Login admin menu 4

```

--- LOGIN ---
Nama pengguna: kohaku
Password: power

Selamat datang, kohaku

=== MENU USER ===
+-----+-----+
| No | Menu |
+-----+-----+
| 1 | Lihat daftar buku tersedia (Read) |
| 2 | Logout |
+-----+-----+

Pilih menu Pengguna: 1

--- DAFTAR BUKU TERSEDIA ---
+-----+-----+-----+-----+-----+-----+
| NO | Judul          | Penulis          | Harga (Rp) | Stok | Kategori |
+-----+-----+-----+-----+-----+-----+
| 1 | Laut Bercerita | Leila S.         | 128.000    | 18  | Novel    |
| 2 | Namaku Alam    | Leila S.         | 157.000    | 10  | Novel    |
| 4 | Dunia Sophie   | Jostein Gaarder  | 267.000    | 16  | Novel    |
+-----+-----+-----+-----+-----+-----+

```

Gambar 4.6 Login user menu 1


```

=== SELAMAT DATANG DI TOKO BUKU SENKUMEDIA ===
+-----+-----+
| No | Menu      |
+-----+-----+
| 1  | Login     |
| 2  | Register  |
| 3  | Logout    |
+-----+-----+
Silakan pilih menu yang tersedia: 2

--- REGISTER PENGGUNA BARU ---
Nama pengguna baru: Johnny Suh
Password baru: NCT127
Registrasi berhasil! Akun dengan nama 'Johnny Suh' telah dibuat.

```

Gambar 4.7 Register sebagai user

```

--- LOGIN ---
Nama pengguna: Johnny Suh
Password: NCT127

Selamat datang, Johnny Suh

=== MENU USER ===
+-----+-----+-----+
| No | Menu                                     |
+-----+-----+-----+
| 1  | Lihat daftar buku tersedia (Read)    |
| 2  | Logout                                |
+-----+-----+-----+
Pilih menu Pengguna: █

```

Gambar 4.8 Login menggunakan akun yang baru registrasi

```

--- REGISTER PENGGUNA BARU ---
Nama pengguna baru: kohaku
Nama pengguna sudah terdaftar. Gunakan nama lain yang unik.
Nama pengguna baru: Mark
Password baru: NCT
Registrasi berhasil! Akun dengan nama 'Mark' telah dibuat.

```

Gambar 4.9 Registrasi dengan nama pengguna yang sudah ada

```

=== SELAMAT DATANG DI TOKO BUKU SENKUMEDIA ===
+---+-----+
| No | Menu      |
+---+-----+
| 1  | Login     |
| 2  | Register  |
| 3  | Logout    |
+---+-----+
Silakan pilih menu yang tersedia: 3

Terima kasih telah mampir di sini. See you next time.

```

Gambar 4.10 Logout sebagai admin maupun user

```

--- DAFTAR BUKU SEKARANG ---
+---+-----+-----+-----+-----+-----+
| NO | Judul          | Penulis              | Harga (Rp) | Stok | Kategori |
+---+-----+-----+-----+-----+-----+
| 1  | Laut Berceita  | Leila S. Chudori     | 128.000    | 18  | Fiksi    |
| 2  | Namaku Alam    | Leila S.             | 157.000    | 10  | Sejarah  |
| 3  | Bumi Manusia  | Pramoedya Ananta Toer | 231.000    | 12  | Fantasi  |
| 4  | Dunia Sophie   | Jostein Gaarder      | 267.000    | 16  | Petualangan |
| 5  | The Setting Sun | Osamu Dazai          | 145.000    | 14  | Fiksi    |
+---+-----+-----+-----+-----+-----+
Tekan enter untuk kembali ke menu Admin...

```

Gambar 4.11 Menu read setelah data buku ditambah

5. Langkah-langkah GIT

5.1 GIT Add

```

C:\Users\w180w\OneDrive\Documents\praktikum-apd>git add .
C:\Users\w180w\OneDrive\Documents\praktikum-apd>

```

Fungsi dari git add . adalah untuk menambahkan semua file yang ada di dalam directory.

5.2 GIT Commit

```

C:\Users\w180w\OneDrive\Documents\praktikum-apd>git commit -m "PT7"
[master 98c4749] PT7
5 files changed, 577 insertions(+)
create mode 100644 kelas/pertemuan-7/.pertemuan-7.py
create mode 100644 post-test/post-test-apd-7/2509106016_Syarafina Nur Amalia.py
create mode 100644 post-test/post-test-apd-7/__pycache__/data.cpython-313.pyc
create mode 100644 post-test/post-test-apd-7/data.py

```

Fungsi dari git commit adalah untuk konfirmasi perubahan yang terjadi di repository, jadi bisa melihat riwayat perubahan yang dilakukan sebelumnya.

5.3 GIT Push

```
C:\Users\w180w\OneDrive\Documents\praktikum-apd>git push -u origin master
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (14/14), 5.89 KiB | 464.00 KiB/s, done.
Total 14 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 3 local objects.
To https://github.com/haroldie666/praktikum-apd
   ad27ef1..98c4749  master -> master
branch 'master' set up to track 'origin/master'.
```

Fungsi dari git push adalah untuk mengupload file yang berada di lokal ke github.