

LAPORAN PRAKTIKUM
POSTTEST (7)
ALGORITMA PEMROGRAMAN DASAR



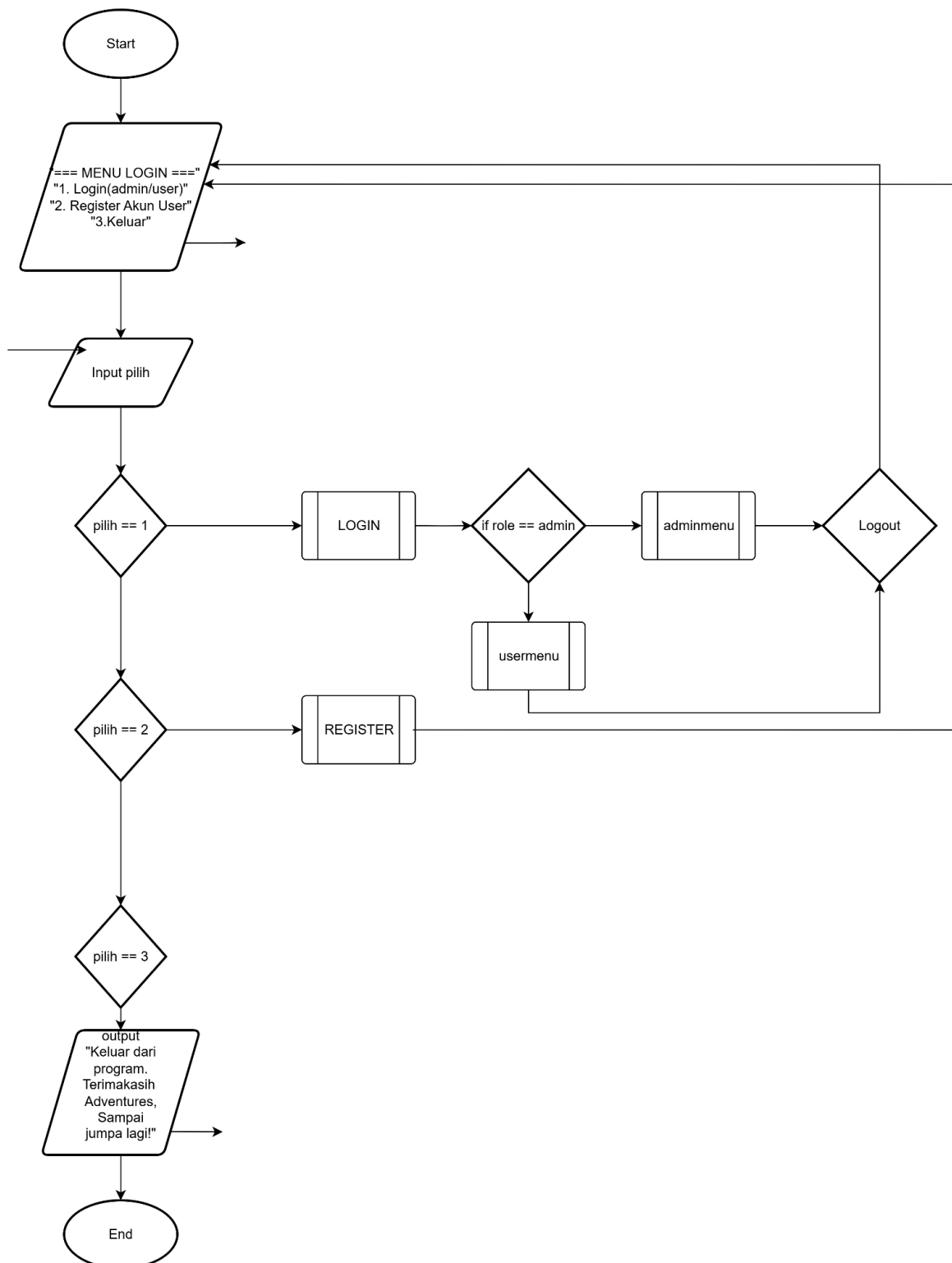
Disusun oleh:

Nama (2509106066)

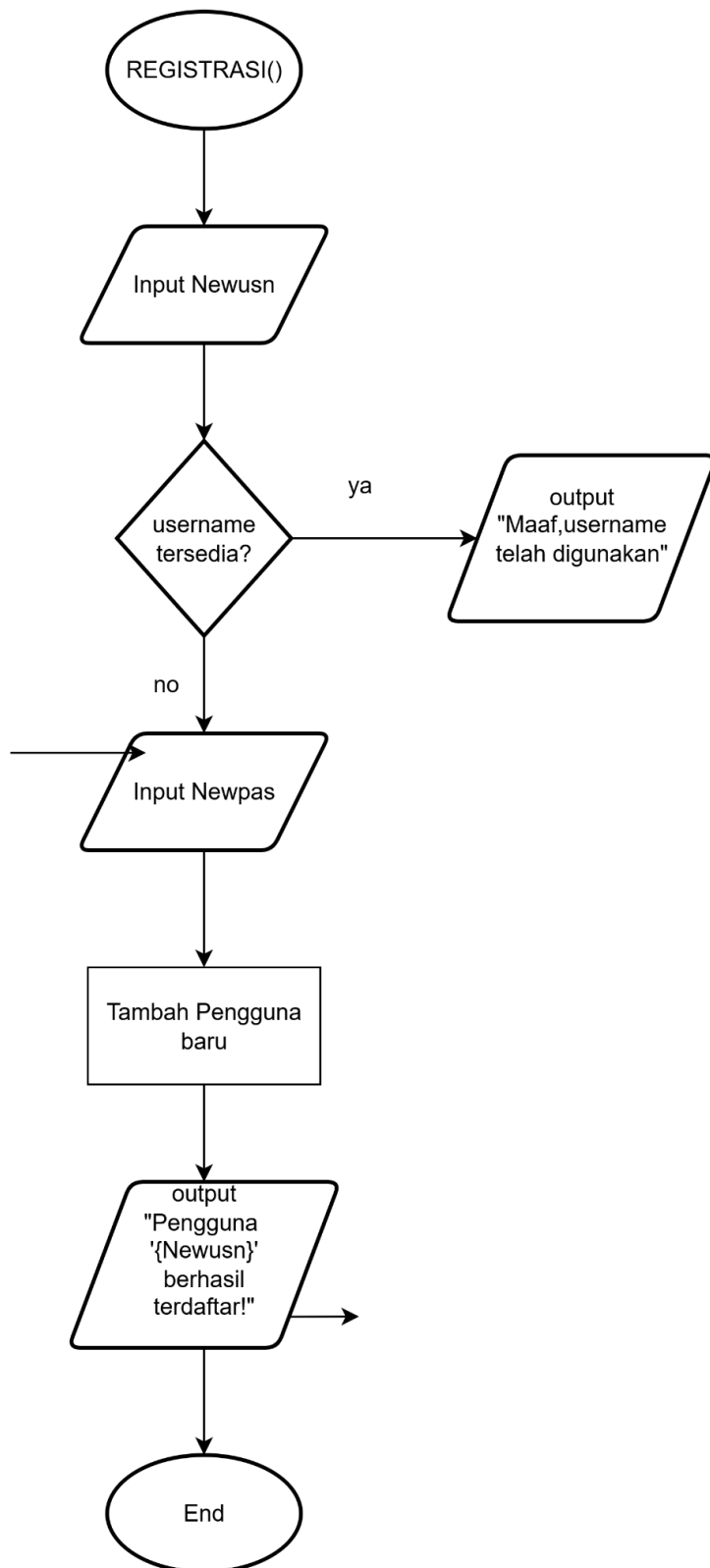
Kelas (B1 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

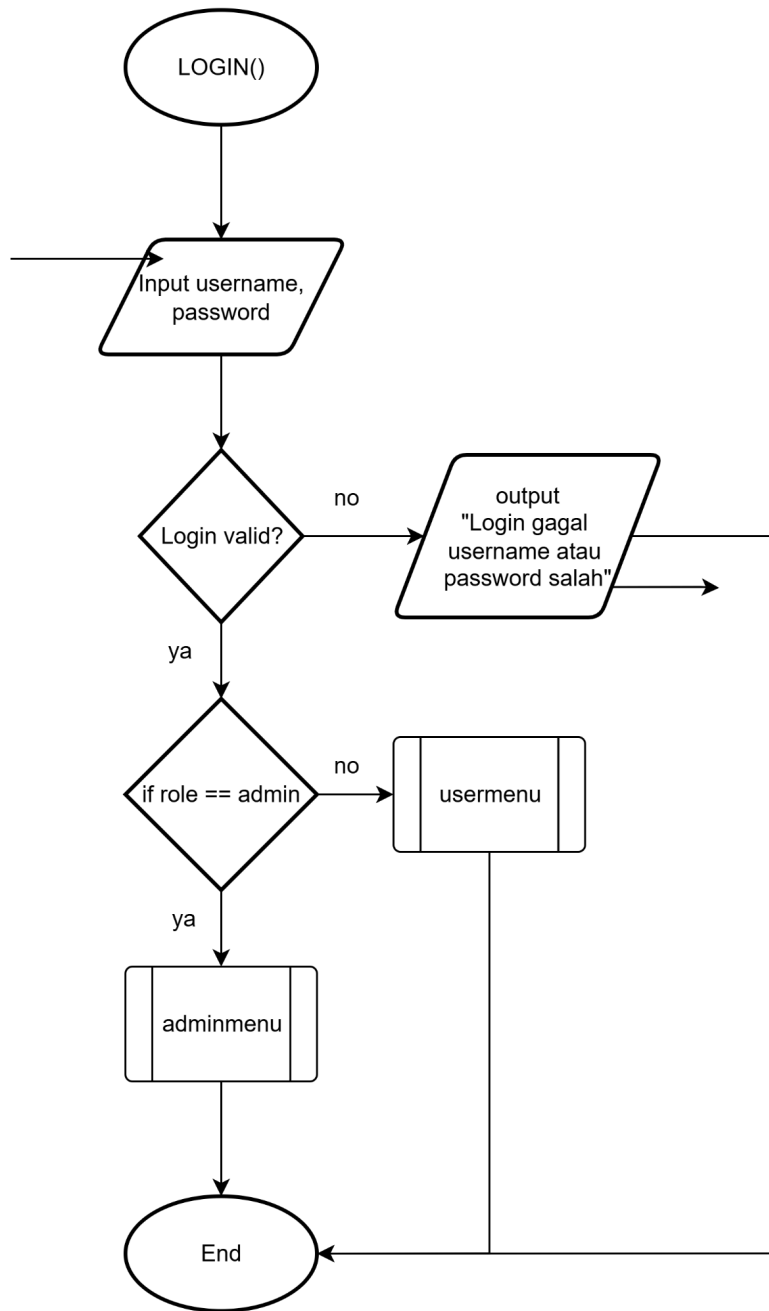
1. Flowchart



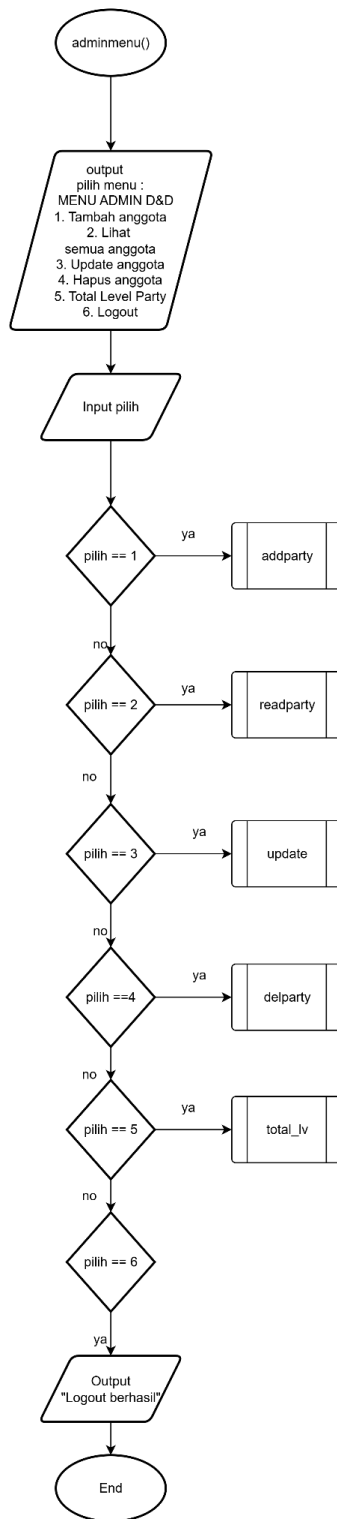
1.1 Flowchart menuawal



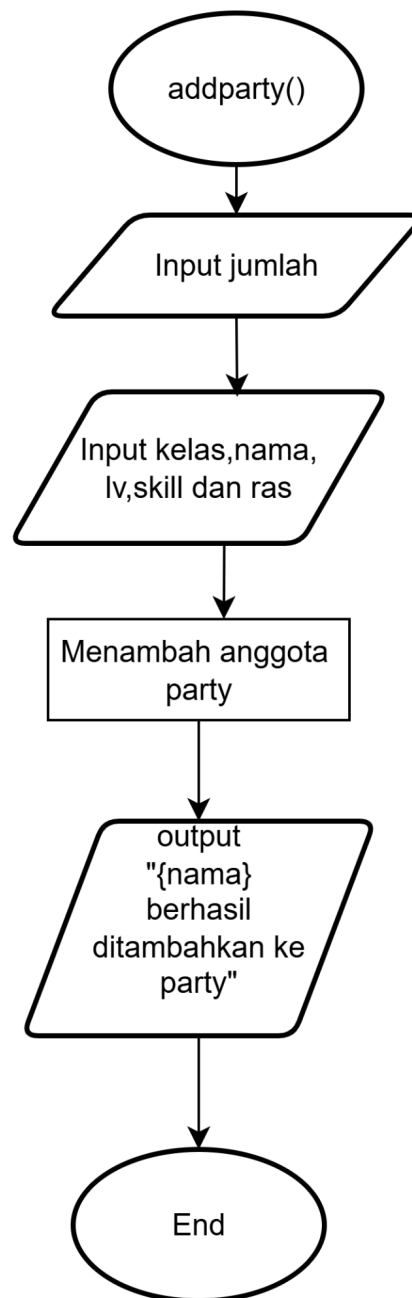
1.2 Flowchart Registrasi



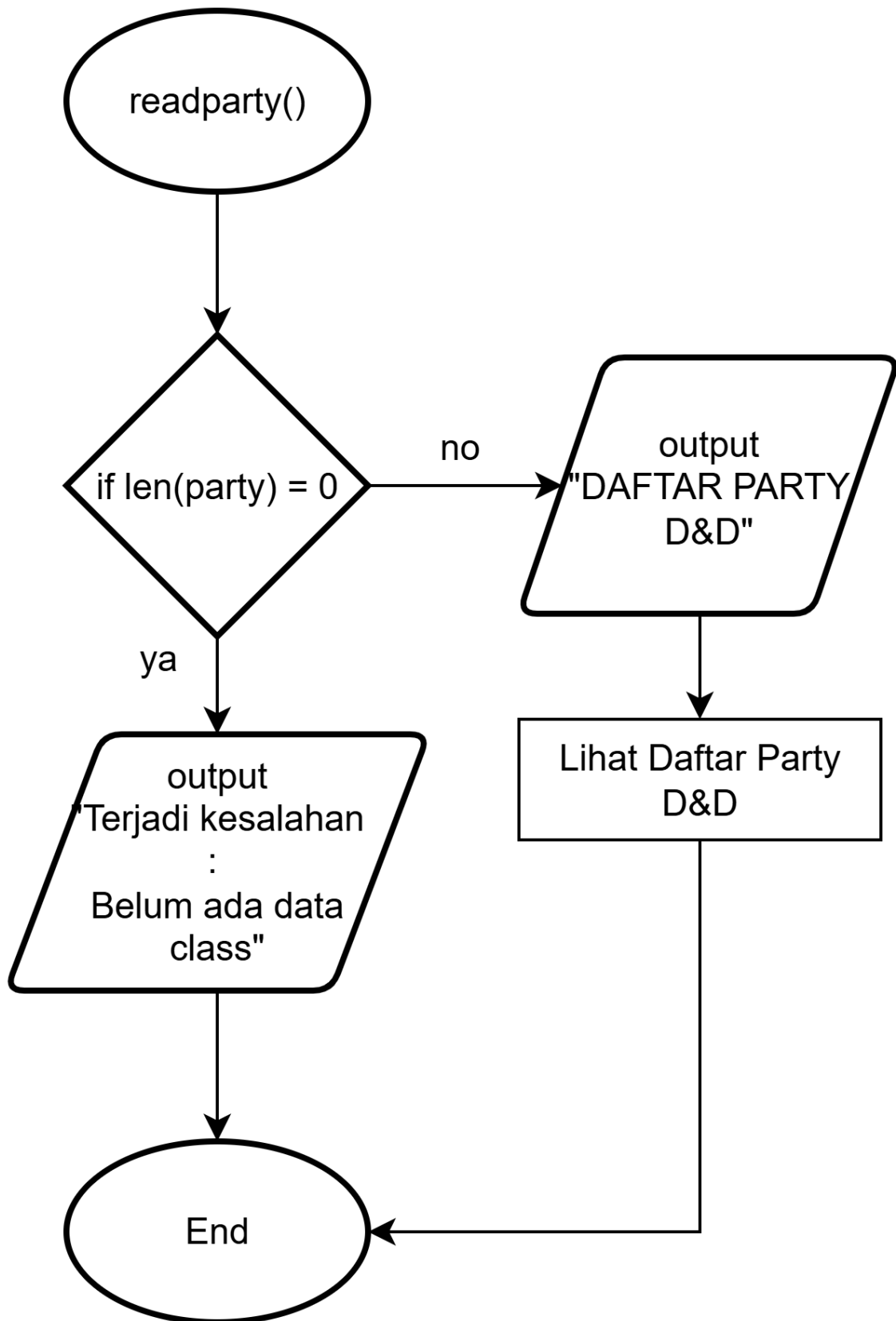
1.3 Flowchart Login



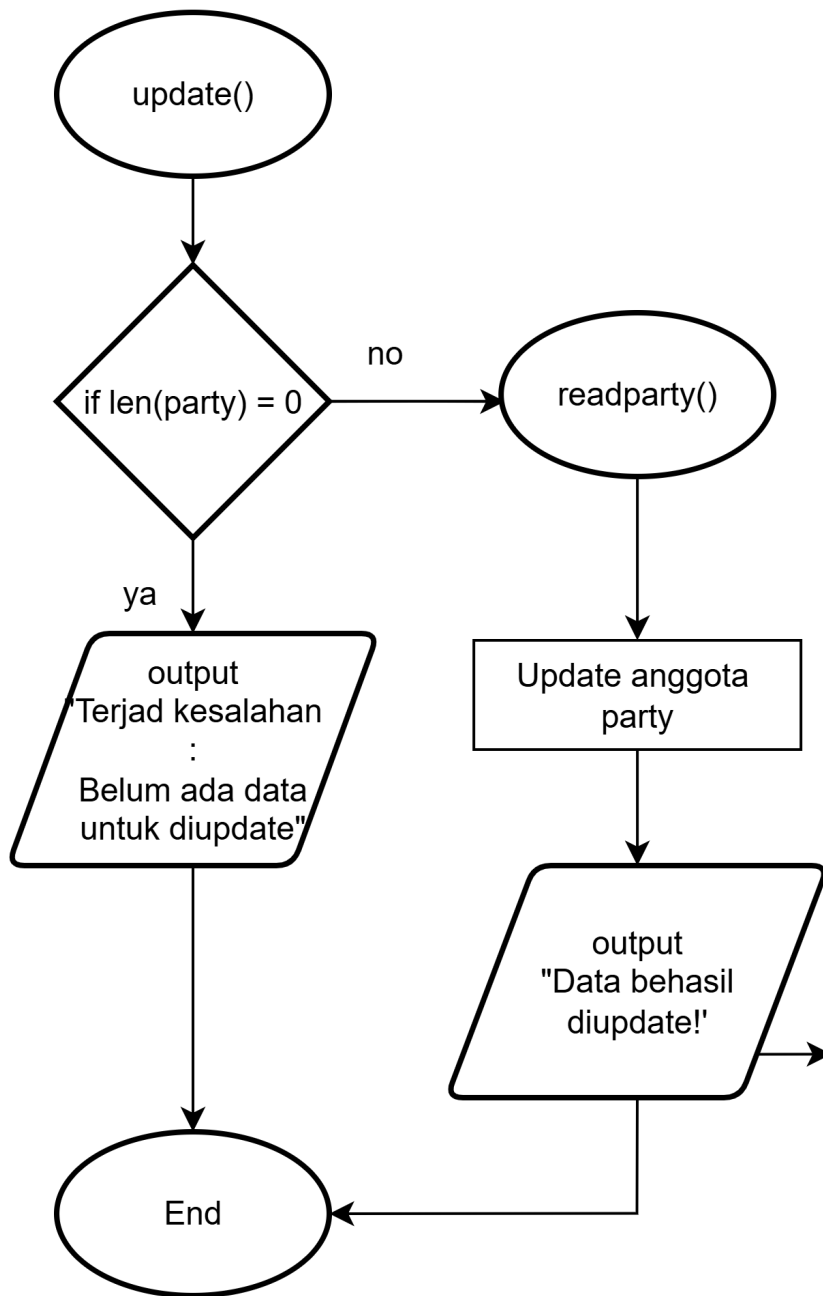
1.4 Flowchart login menu admin



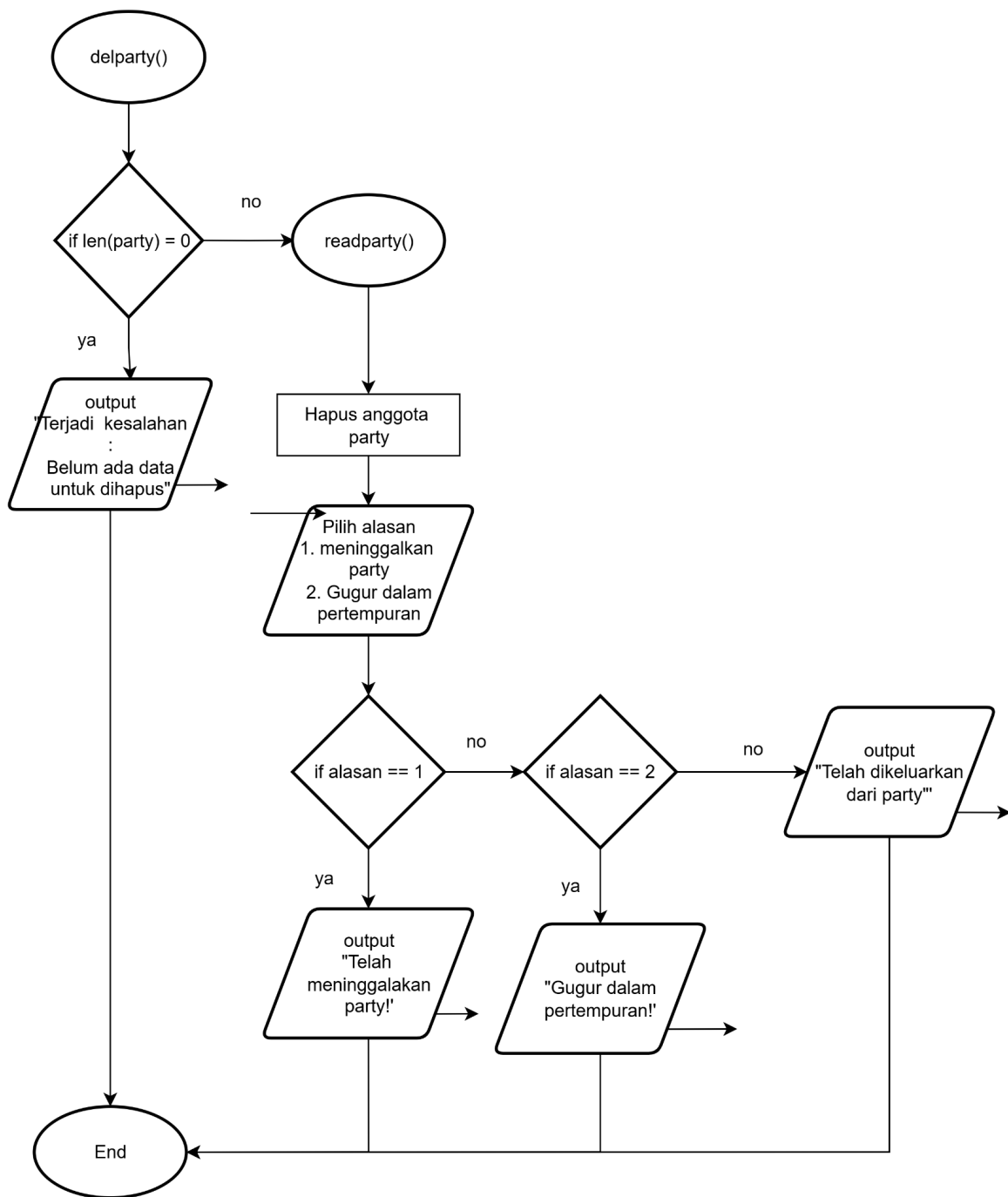
1.5 Flowchart addparty(create)



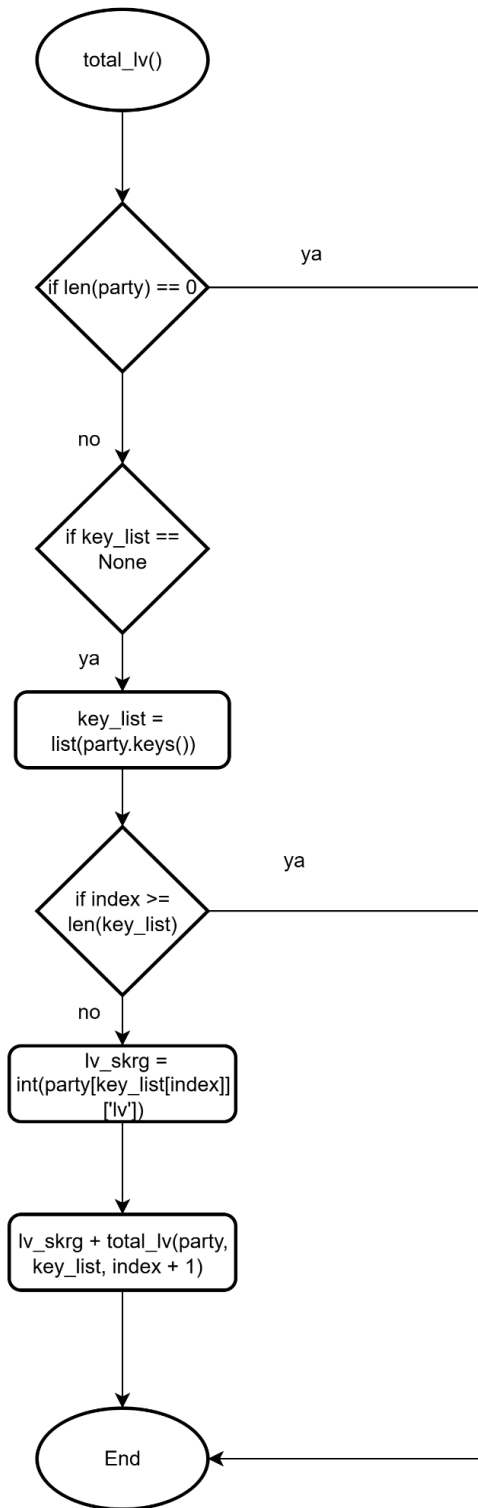
1.6 Flowchart Readparty



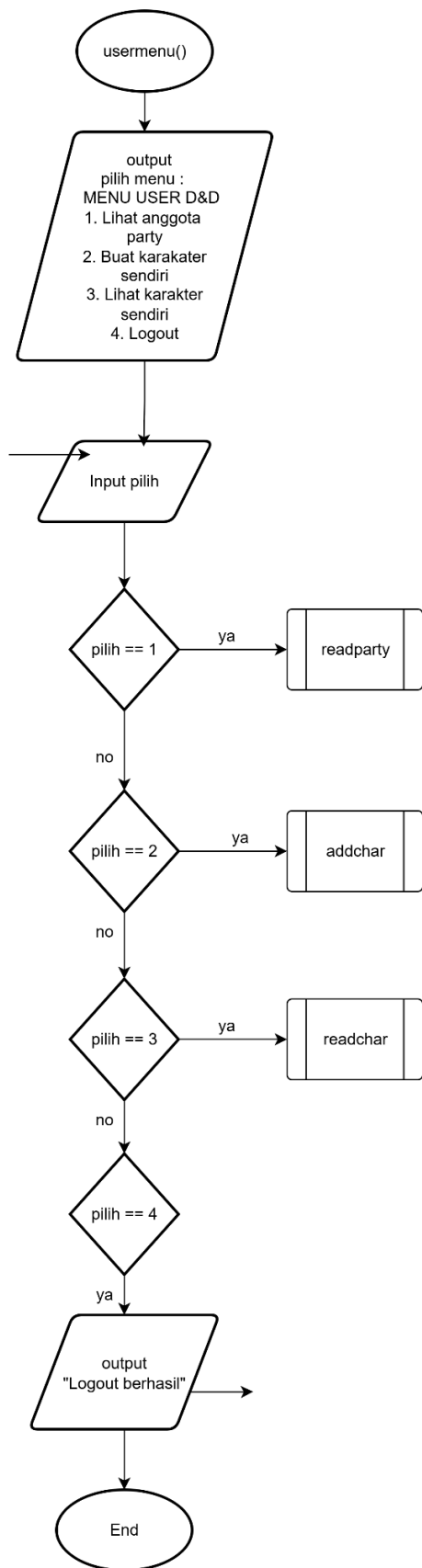
1.7 Flowchart update party



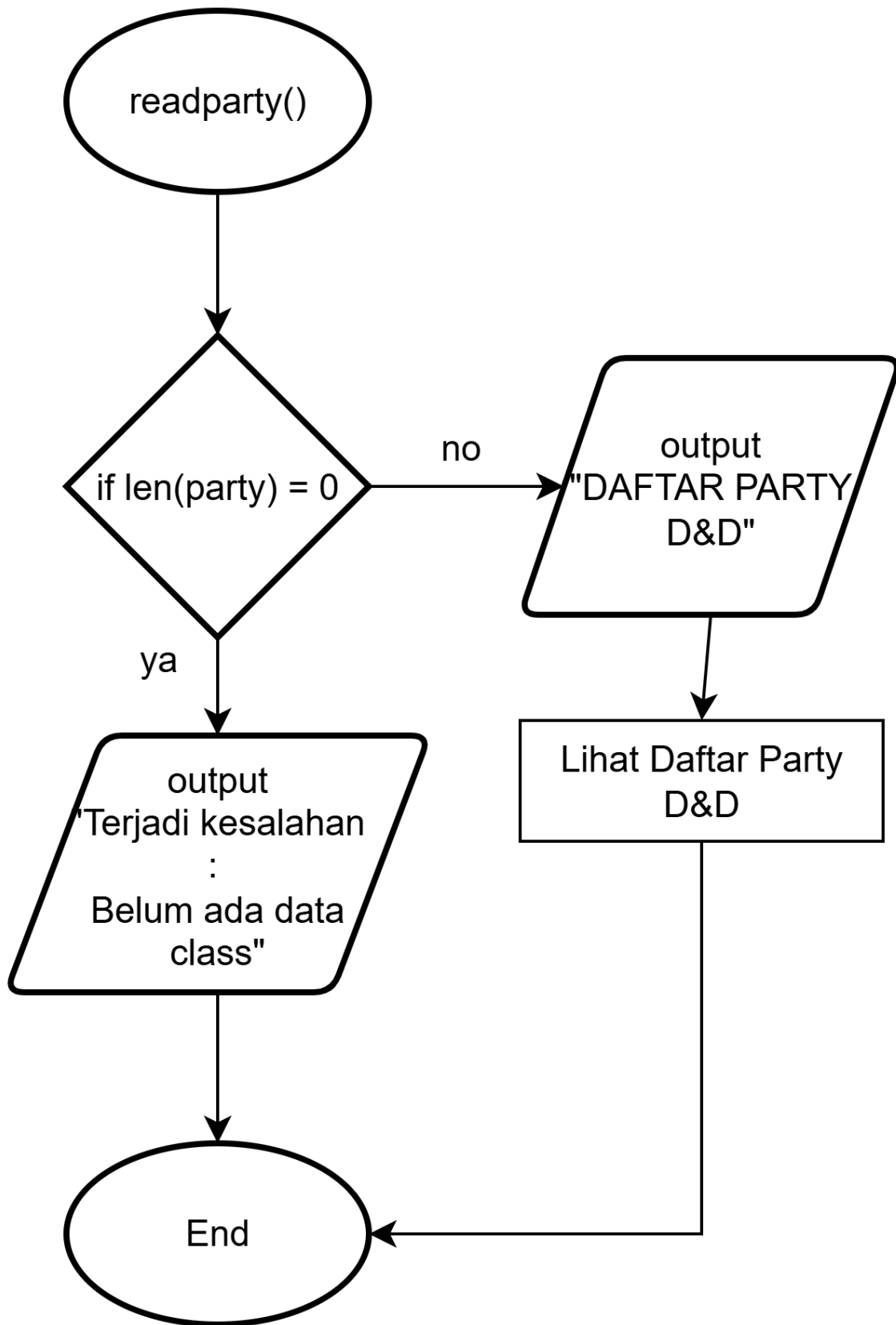
1.8 Flowchart delete party



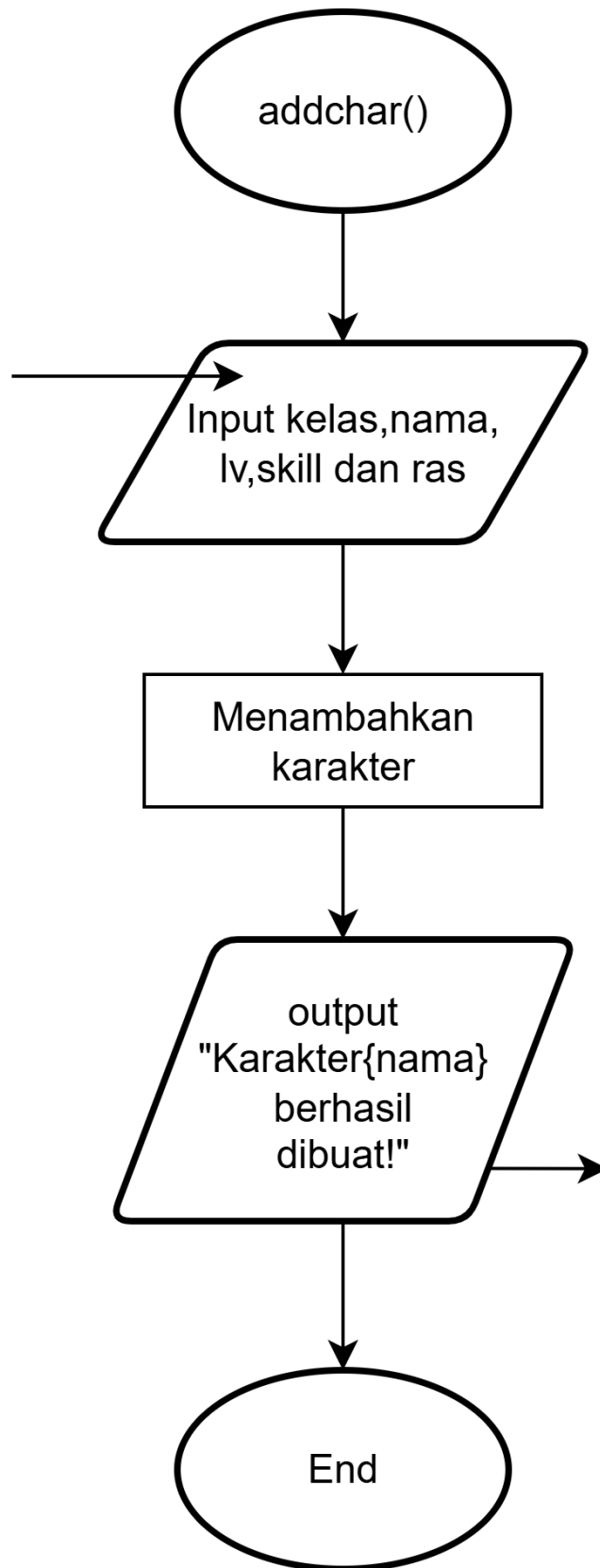
1.9 flowvhart total level



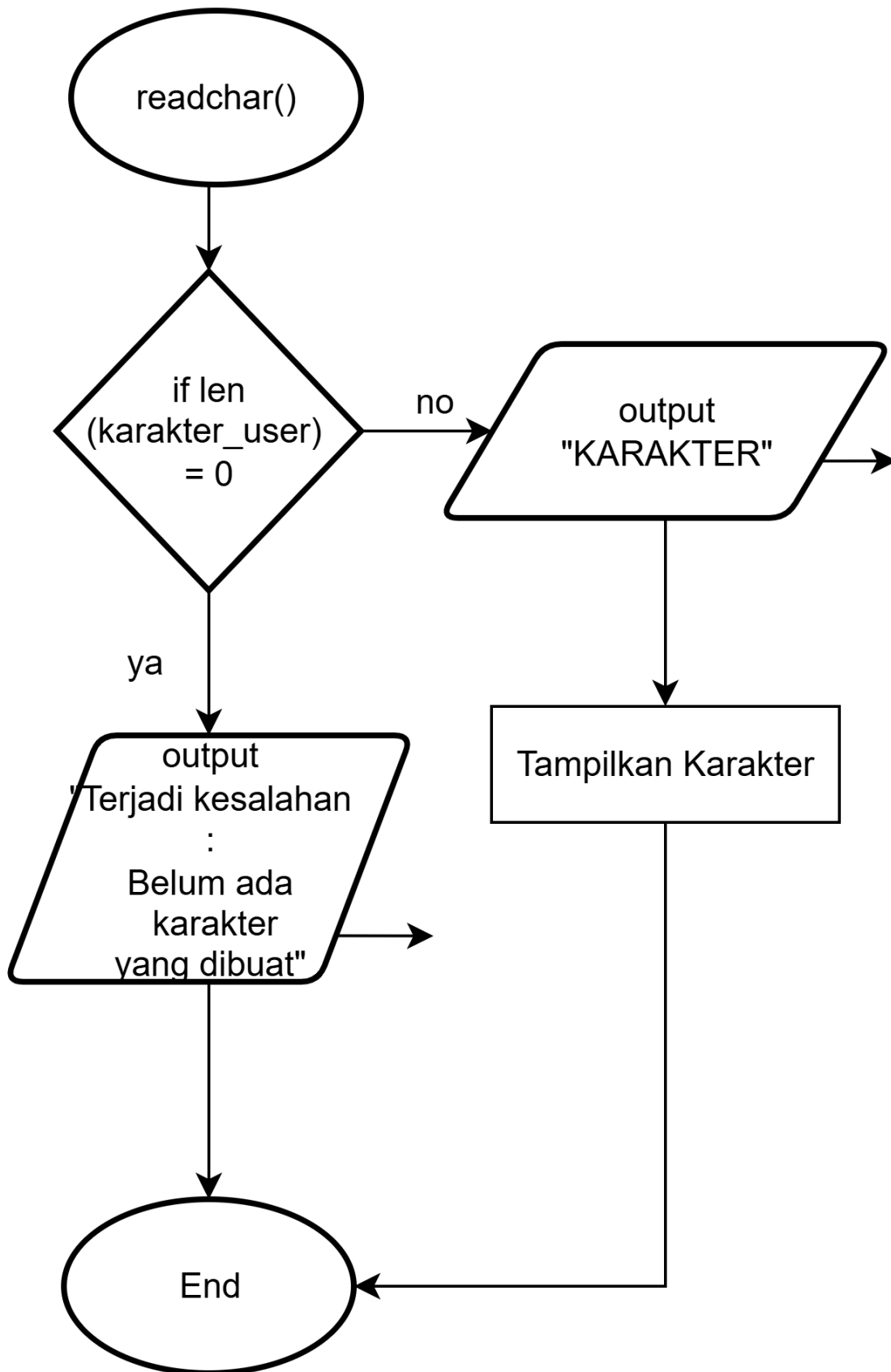
1,10 Flowchart menu user



1.11 Flowchart lihat anggota party(menu user)



1.12 Flowchart addchar(create karakter)



1.13 Flowchart readchar

2. Deskripsi Singkat Program

Program ini adalah **simulasi sistem manajemen party dan karakter** dalam permainan **Dungeon & Dragons (D&D)** berbasis teks menggunakan **Python**.

Pengguna dapat **login sebagai admin atau user**, mengelola **anggota party**, serta membuat dan melihat **karakter pribadi**.

Program ini memiliki dua jenis pengguna, yaitu admin dan user.

Admin memiliki hak akses penuh untuk menambah, melihat, memperbarui, dan menghapus anggota party (fitur CRUD), sedangkan user hanya dapat membuat karakter pribadi dan melihat data party yang telah ada.

Register digunakan agar pengguna baru dapat membuat akun sendiri tanpa harus melalui admin.

Fitur login berfungsi untuk melakukan proses autentikasi terhadap pengguna sebelum masuk ke dalam sistem.

Setiap pengguna memiliki username, password, dan peran (role) masing-masing. Role tersebut bisa berupa admin atau user.

Role admin = username = admin

password = dungeon master

Pada proses login, program akan meminta pengguna memasukkan username dan password. Kemudian sistem akan mencocokkan data yang dimasukkan dengan data yang tersimpan dalam daftar pengguna. Jika username dan password sesuai, pengguna akan diarahkan ke menu sesuai dengan perannya:

Fitur Admin:

- Menambah anggota party
- Melihat semua anggota party
- Mengupdate data anggota
- Menghapus anggota (dengan alasan: meninggalkan party atau gugur dalam pertempuran)
- Menghitung total level party (menggunakan rekursi)

Fitur User:

- Melihat daftar anggota party
- Membuat karakter sendiri
- Melihat karakter yang telah dibuat

Fitur Sistem:

- Login & Register akun
- Role-based access (Admin & User)
- Logout dengan notifikasi

Jika data yang dimasukkan salah, maka sistem akan menampilkan pesan bahwa login gagal.

Sistem login ini memastikan keamanan data dengan membatasi akses hanya kepada pengguna yang memiliki akun terdaftar.

Saat proses registrasi, program akan meminta pengguna membuat username dan password baru.

Setelah itu, sistem akan menyimpan data tersebut ke dalam daftar akun dengan peran otomatis sebagai user.

Dan untuk penjelasan CRUDNYA:

Create = membuat data class party anggota beserta nama,ras,skill, lv dan classnya.

Read = menampilkan data seluruh anggota party dan classnya

Update = untuk mengubah atau memperbarui lv,ras,nama,skill dan classnya.

Delete = untuk menghapus anggota yang pergi dari party atau telah mati dalam pertempuran

Data yang Disimpan: Setiap karakter memiliki atribut: Class, Nama, Level, Skill Utama, dan Ras. Program ini menggunakan dictionary Python untuk menyimpan data dan menerapkan konsep CRUD (Create, Read, Update, Delete) dengan antarmuka yang user-friendly dan tema D&D/fantasy.

3. Source Code

Source Code:

```
import os

users = {
    "admin": {'password': 'dungeon master', 'role': 'admin'}
}

party = {}
party_id = 1
karakter_user = {}
karakterid = 1
```

3.1 Data default

```
def clear():
    os.system('cls || clear')

def LOGIN():
    clear()

    print(line1)

    print("=== LOGIN ===".center(50))

    print(line1)

    username = input("Masukkan username : ")

    password = input("Masukkan password : ")
```

```

if username in users and users[username]['password'] == password:

    clear()

    print(line1)

    print(f"Login berhasil! Selamat datang, {username}
({users[username]['role'].capitalize()})")

    print(line1)

    input("Tekan Enter untuk melanjutkan ke menu...")

    return {

        "username": username,

        "password": users[username]['password'],

        "role": users[username]['role']

    }

print("Login gagal. Username atau password salah.")

input("Tekan Enter untuk melanjutkan...")

return None

```

3.2 def clear dan login

```

def REGISTER():

    try:

        clear()

```

```

print(line1)

print("=== REGISTER AKUN USER ===".center(50))

print(line1)

Newusn = input("Buat username : ")

if Newusn in users:

    print(f"\nMaaf, username '{Newusn}' telah digunakan.")

    input("Tekan Enter untuk melanjutkan...")

    return

Newpas = input("Buat password : ")

users[Newusn] = {'password': Newpas, 'role': 'user'}

print(f"\nPengguna '{Newusn}' berhasil terdaftar!")

input("\nTekan Enter untuk kembali ke menu...")

except Exception as e:

    print(f"Terjadi kesalahan saat registrasi : {str(e)}")

    input("Tekan Enter untuk melanjutkan...")

```

3.3 def register

```

def addparty():

    global party_id

    try:

        clear()

```

```

print(line1)

print("=== Tambah Anggota Party ===".center(50))

print(line1)

jumlah = int(input("Berapa anggota yang ingin ditambahkan? "))

for i in range(jumlah):

    clear()

    print(f"\nAnggota ke-{i+1}")

    kelas = input("Class : ")

    nama = input("Nama : ")

    lv = input("Lv : ")

    skill = input("Skill utama : ")

    ras = input("Ras : ")

    party[party_id] = {

        'kelas': kelas,

        'nama': nama,

        'lv': lv,

        'skill': skill,

        'ras': ras

    }

    print(f"{nama} berhasil ditambahkan ke party!")

    party_id += 1

    input("\nTekan Enter untuk kembali ke menu...")

except ValueError:

```

```

        print("Input harus berupa angka!")

        input("Tekan Enter untuk melanjutkan...")

def readparty():

    try:

        clear()

        if len(party) == 0:

            raise IndexError("Belum ada data class.")

        else:

            print(line1)

            print("=== DAFTAR PARTY D&D ===".center(50))

            print(line1)

            for key, member in party.items():

                print(f"{key}. Class : {member['kelas']}")

                print(f"    Nama   : {member['nama']}")

                print(f"    Lv    : {member['lv']}")

                print(f"    Skill : {member['skill']}")

                print(f"    Ras   : {member['ras']}")

                print(line1)

    except IndexError as e:

        print(f"Terjadi kesalahan : {e}")

```

```

input("\nTekan Enter untuk melanjutkan...")

clear()

def update():

    clear()

    if len(party) == 0:

        print("Belum ada data untuk diupdate.")

    else:

        clear()

        readparty()

        try:

            print(line1)

            print("=== Update Anggota Party ===".center(50))

            print(line1)

            ubah = int(input("\nPilih nomor yang ingin diupdate: "))

            if ubah in party:

                print("\nMasukkan Data Baru : ")

                party[ubah]['kelas'] = input("Class baru : ")

                party[ubah]['nama'] = input("Nama baru : ")

                party[ubah]['lv'] = input("Lv baru : ")

                party[ubah]['skill'] = input("Skill baru : ")

                party[ubah]['ras'] = input("Ras baru : ")

                print("\nData berhasil diupdate!")

```

```

        clear()

    else:

        print("Nomor tidak valid!")

    except ValueError:

        print("Input harus berupa angka!")

input("Tekan Enter untuk kembali ke menu...")

clear()

def delparty():

    clear()

    if len(party) == 0:

        print("Belum ada data untuk dihapus.")

    else:

        readparty()

        try:

            print(line1)

            print("\n=== Daftar Anggota Untuk Dihapus ===".center(50))

            print(line1)

            hapus = int(input("Pilih nomor yang ingin dihapus: "))

            if hapus in party:

                nama_hapus = party[hapus]['nama']

                print("\nAlasan dihapus : ")

```



```

        print("1. Meninggalkan Party")

        print("2. Gugur Dalam Pertempuran")

        alasan = input("Pilih alasan (1/2): ")

        if alasan == '1':

            print(f"{nama_hapus} telah meninggalkan party.")

        elif alasan == '2':

            print(f"{nama_hapus} telah gugur dalam pertempuran.")

        else:

            print(f"{nama_hapus} telah dikeluarkan dari party.")

        del party[hapus]

        clear()

    else:

        print("Nomor tidak valid.")

except ValueError:

    print("Input harus berupa angka!")

input("Tekan Enter untuk kembali ke menu...")

clear()

```

3.4 def CRUD

```

def total_lv(party, key_list=None, index=0):

    clear()

    print(line1)

```

```

print("=== Total Level Party ===".center(50))

print(line1)

if len(party) == 0:

    return 0

if key_list is None:

    key_list = list(party.keys())

if index >= len(key_list):

    return 0

try:

    lv_skrng = int(party[key_list[index]]['lv'])

except ValueError:

    print(f"Level {party[key_list[index]]['nama']} tidak valid, dilewati!")

    lv_skrng = 0

return lv_skrng + total_lv(party, key_list, index + 1)

```

3.5 de total_lv

```

def addchar():

    global karakterid

    try:

        clear()

```

```

print(line1)

print("=== Buat Karakter Sendiri ===".center(50))

print(line1)

kelas = input("Class : ")

nama = input("Nama : ")

lv = input("Lv : ")

skill = input("Skill utama : ")

ras = input("Ras : ")

if not all([kelas, nama, lv, skill, ras]):

    raise Exception("Semua field harus diisi.")

karakter_user[karakterid] = {

    'kelas': kelas,

    'nama': nama,

    'lv': lv,

    'skill': skill,

    'ras': ras

}

print(f"\nKarakter {nama} berhasil dibuat!")

karakterid += 1

except Exception as e:

```

```

        print(f"\nTerjadi kesalahan: {e}")

    input("\nTekan Enter untuk kembali ke menu...")

def readchar():

    try:

        clear()

        if len(karakter_user) == 0:

            raise IndexError("Belum ada karakter yang dibuat.")

        else:

            print(line1)

            print("=== KARAKTER ===".center(50))

            print(line1)

            for key, karakter in karakter_user.items():

                print(f"{key}. Class : {karakter['kelas']}")

                print(f"    Nama   : {karakter['nama']}")

                print(f"    Lv    : {karakter['lv']}")

                print(f"    Skill : {karakter['skill']}")

                print(f"    Ras   : {karakter['ras']}")

                print(line1)

    except IndexError as e:

        print(f"Terjadi kesalahan : {e}")

    input("\nTekan Enter untuk kembali ke menu...")

```

```
clear()
```

3.6 *def addchar(create) dan def readchar(read)*

```
def adminmenu():  
    while True:  
        try:  
            clear()  
  
            print(line1)  
  
            print("=== MENU ADMIN D&D ===".center(50))  
  
            print(line1)  
  
            print("1. Tambah anggota".center(50))  
  
            print("2. Lihat semua anggota".center(50))  
  
            print("3. Update anggota".center(50))  
  
            print("4. Hapus anggota".center(50))  
  
            print("5. Total Level Party".center(50))  
  
            print("6. Logout".center(50))  
  
            print(line1)  
  
            pilih = int(input("Pilih menu : "))  
  
            if pilih == 1:  
                addparty()  
  
            elif pilih == 2:  
                readparty()  
  
            elif pilih == 3:  
                update()
```

```

elif pilih == 4:

    delparty()

elif pilih == 5:

    hasil = total_lv(party)

    print(f"\nTotal Level Party adalah: {hasil}")

    input("\nTekan Enter untuk kembali ke menu...")

elif pilih == 6:

    clear()

    print(line1)

    print("Logout berhasil!".center(50))

    print(line1)

    input("Tekan Enter untuk kembali ke menu awal...")

    break

else:

    print("Pilihan tidak valid!")

    input("Tekan Enter untuk kembali ke menu...")

except ValueError:

    print("Input harus angka!")

    input("Tekan Enter untuk melanjutkan...")

```

3.7 def adminmenu

```

def usermenu(username):

    while True:

        try:

            clear()

            print(line1)

            print(f"=== MENU USER D&D ({username}) ===".center(50))

            print(line1)

            print("1. Lihat anggota party".center(50))

            print("2. Buat karakter sendiri".center(50))

            print("3. Lihat karakter sendiri".center(50))

            print("4. Logout".center(50))

            print(line1)

            pilih = int(input("Pilih menu : "))

            if pilih == 1:

                readparty()

            elif pilih == 2:

                addchar()

            elif pilih == 3:

                readchar()

            elif pilih == 4:

                clear()

                print(line1)

```

```
        print("Logout berhasil!".center(50))

        print(line1)

        input("Tekan Enter untuk kembali ke menu awal...")

        break

    else:

        print("Pilihan tidak valid.")

        input("\nTekan Enter untuk kembali ke menu...")

except ValueError:

    print("Input harus angka!")

    input("Tekan Enter untuk melanjutkan...")
```

3.8 def usermenu


```

def menuwal():
    while True:
        clear()

        print(line1)

        print("=== MENU LOGIN ===".center(50))

        print(line1)

        print("1. Login(admin/user)".center(50))

        print("2. Register Akun User".center(50))

        print("3. Logout".center(50))

        print(line1)

        pilih = input("\nMasukkan pilihan akun : ")

        if pilih == '1':
            user_login = LOGIN()

            if user_login:
                if user_login['role'] == 'admin':
                    adminmenu()

                else:
                    usermenu(user_login['username'])

            elif pilih == '2':
                REGISTER()

            elif pilih == '3':
                clear()

                print("Keluar dari program. Terima kasih Adventurer, sampai jumpa lagi!")

                break

            else:

```

```
        print("Pilihan tidak valid.")  
        input("\nTekan Enter untuk kembali ke menu...")  
menuawal()
```

3.9 def menuawal dan memulai program

4. Hasil Output

```
=====
=== MENU LOGIN ===
=====
1. Login(admin/user)
2. Register Akun User
3. Logout
=====

Masukkan pilihan akun : █
```

Gambar 4.1 Menu login

```
=====
=== LOGIN ===
=====
Masukkan username : admin
Masukkan password : dungeon master█

=====
Login berhasil! Selamat datang, admin (Admin)
=====
Tekan Enter untuk melanjutkan ke menu...█
```

Gambar 4.2 Login admin

```
=====
=== MENU ADMIN D&D ===
=====

1. Tambah anggota
2. Lihat semua anggota
3. Update anggota
4. Hapus anggota
5. Total Level Party
6. Logout

=====
Pilih menu : █
```

Gambar 4.3 Menu admin D&D

```
=====
=== Tambah Anggota Party ===
=====

Berapa anggota yang ingin ditambahkan? 1

Anggota ke-1
Class : cleric
Nama : arta
Lv : 10
Skill utama : cure wand
Ras : Elf
arta berhasil ditambahkan ke party!

Tekan Enter untuk kembali ke menu... █
```

Gambar 4.4 buat anggota party

```
=====
          === DAFTAR PARTY D&D ===
=====
1. Class : cleric
   Nama  : arta
   Lv    : 10
   Skill : cure wand
   Ras   : Elf
=====

Tekan Enter untuk melanjutkan...|
```

Gambar 4.5 lihat anggota party

```
=====
          === DAFTAR PARTY D&D ===
=====
1. Class : cleric
   Nama  : arta
   Lv    : 10
   Skill : cure wand
   Ras   : Elf
=====

Tekan Enter untuk melanjutkan...
=====
          === Update Anggota Party ===
=====

Pilih nomor yang ingin diupdate: 1

Masukkan Data Baru :
Class baru : cleric
Nama baru  : arta
Lv baru    : 15
Skill baru : heal area
Ras baru   : Elf

Data berhasil diupdate!
Tekan Enter untuk kembali ke menu...[]
```

Gambar 4.6 update anggota

```
=====
===Daftar Anggota Untuk Dihapus===
=====
Pilih nomor yang ingin dihapus: 1

Alasan dihapus :
1. Meninggalkan Party
2. Gugur Dalam Pertempuran
Pilih alasan (1/2): 1
arta telah meninggalkan party.
Tekan Enter untuk kembali ke menu...
```

Gambar 4.7 hapus anggota

```
=====
=== Total Level Party ===
=====

Total Level Party adalah: 15

Tekan Enter untuk kembali ke menu...
```

Gambar 4.8 Total level party

```
=====
=== REGISTER AKUN USER ===
=====
Buat username : arya
Buat password : 66

Pengguna 'arya' berhasil terdaftar!

Tekan Enter untuk kembali ke menu...
```

Gambar 4.9 register

```
=====
=== LOGIN ===
=====
Masukkan username : arya
Masukkan password : 66
Login berhasil! Selamat datang, arya (User)
=====
Tekan Enter untuk melanjutkan ke menu...
```

Gambar 4.10 login user

```
=====
=== MENU USER D&D ===
=====
1. Lihat anggota party
2. Buat karakter sendiri
3. Lihat karakter sendiri
4. Logout
=====
Pilih menu :
```

Gambar 4.11 menu user


```
=====
                        === DAFTAR PARTY D&D ===
=====
1. Class : cleric
   Nama  : arta
   Lv    : 10
   Skill : cure wand
   Ras   : Elf
=====

Tekan Enter untuk melanjutkan...|
```

Gambar 4.12 Lihat daftar party sebagai user

```
=====
                        === Buat Karakter Sendiri ===
=====
Class : warloc
Nama  : rei
Lv    : 9
Skill utama : blast
Ras   : human

Karakter rei berhasil dibuat!

Tekan Enter untuk kembali ke menu...|
```

Gambar 4.13 buat karakter

```
=====
=== KARAKTER ===
=====
1. Class : warloc
   Nama  : rei
   Lv    : 9
   Skill : blast
   Ras   : human
=====

Tekan Enter untuk kembali ke menu...
```

Gambar 4.14 lihat karakter

```
=====
Logout berhasil!
=====
Tekan Enter untuk kembali ke menu awal...
```

Gambar 4.15 Logout

```
Keluar dari program. Terima kasih Adventurer, sampai jumpa lagi!
PS C:\Post_Test_Apd>
```

Gambar 4.16 Keluar dari program

5. Langkah-langkah GIT

```
PS C:\Post_Test_Apd> git add .
warning: in the working copy of 'Post-Test-Apd/post-test-apd-5/2509106066-Aryasatya Rakha Ph
be replaced by CRLF the next time Git touches it
PS C:\Post_Test_Apd> git commit -m "Surrend"
[main 6d12eb4] Surrend
 3 files changed, 454 insertions(+), 17 deletions(-)
 create mode 100644 Post-Test-Apd/post-test-apd-7/2509106066-Aryasatya Rakha Phanyputra-PT-7
PS C:\Post_Test_Apd> git push -u origin main
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 3.49 KiB | 715.00 KiB/s, done.
```

5.1 GIT Add

git add untuk menambahkan file yang sudah diubah atau baru di repository.

5.2 GIT Commit

git commit untuk menyimpan perubahan repository dengan pesan commit.

5.3 GIT Push

git push berfungsi untuk mengirim commit dari repository lokal ke repository remote (GitHub) sekaligus menetapkan branch default.