

LAPORAN PENJELASAN PROGRAM



Nama Dosen
Randi Proska Sandra, S.Pd., M.Sc.

Nama	: YAZID AQIL ASSALAM
NIM	: 22343014
Fakultas	: Teknik
Departemen	: Teknik Elektronika
Prodi	: Informatika
Mata Kuliah	: Praktikum Struktur Data
No. Seksi Mata Kuliah	: 202223430028

2023

A. IDENTITAS MAHASISWA

1. Nama : YAZID AQIL ASSALAM
2. NIM : 22343014
3. Prodi : Informatika
4. Mata Kuliah : Praktikum Struktur Data
5. No. Seksi Mata Kuliah : 202223430028
6. Kode Mata Kuliah : INF1.62.2014

B. NAMA PROGRAM

Program mengurutkan data berdasarkan nilai tertinggi ke nilai terendah

```
print("\n|   YAZID AQIL ASSALAM   |")
print("|   22343014   |")
print("|   Praktikum Struktur Data   |")
print("|   202223430028   |")

class Simpul:
    def __inisial__(self, nim, nama, prodi, nilai):
        self.nim = nim
        self.nama = nama
        self.prodi = prodi
        self.nilai = nilai
        self.selanjutnya = None

class DaftarSimpul:
    def __inisial__(self):
        self.kepala = None

    def tambah_data(self, nim, nama, prodi, nilai):
        simpul = Simpul(nim, nama, prodi, nilai)
        if self.kepala is None:
            self.kepala = simpul
        else:
            saat_ini = self.kepala
            while saat_ini.selanjutnya is not None:
                saat_ini = saat_ini.selanjutnya
            saat_ini.selanjutnya = simpul
        print("Data berhasil ditambahkan")

    def cetak_data(self):
        if self.kepala is None:
            print("Data masih kosong")
        else:
            saat_ini = self.kepala
            print("\nData dalam daftar:")
            while saat_ini is not None:
                print("NIM: {}, Nama: {}, Prodi: {}, Nilai: {}".format(saat_ini.nim, saat_ini.nama, saat_ini.prodi, saat_ini.nilai))
                saat_ini = saat_ini.selanjutnya

    def cari_data(self, nim):
        saat_ini = self.kepala
        while saat_ini is not None:
            if saat_ini.nim == nim:
                return saat_ini
            saat_ini = saat_ini.selanjutnya
        return None

    def ubah_data(self, nim, nama, prodi, nilai):
        simpul = self.cari_data(nim)
        if simpul is not None:
            simpul.nama = nama
            simpul.prodi = prodi
            simpul.nilai = nilai
            print("Data berhasil diubah")
        else:
```

```

        print("Data dengan NIM tersebut tidak ditemukan")

    def hapus_data(self, nim):
        saat_ini = self.kepala
        sebelumnya = None
        while saat_ini is not None:
            if saat_ini.nim == nim:
                if sebelumnya is None:
                    self.kepala = saat_ini.selanjutnya
                else:
                    sebelumnya.selanjutnya = saat_ini.selanjutnya
                print("Data berhasil dihapus")
                return
            sebelumnya = saat_ini
            saat_ini = saat_ini.selanjutnya
        print("Data dengan NIM tersebut tidak ditemukan")

    def tukar_data(self, a, b):
        temp_nilai = a.nilai
        temp_nim = a.nim
        temp_nama = a.nama
        temp_prodi = a.prodi

        a.nilai = b.nilai
        a.nim = b.nim
        a.nama = b.nama
        a.prodi = b.prodi

        b.nilai = temp_nilai
        b.nim = temp_nim
        b.nama = temp_nama
        b.prodi = temp_prodi

    def urut_data(self):
        if self.kepala is None:
            return
        akhir = None
        while akhir != self.kepala:
            saat_ini = self.kepala
            while saat_ini.selanjutnya != akhir:
                if saat_ini.nilai < saat_ini.selanjutnya.nilai:
                    self.tukar_data(saat_ini, saat_ini.selanjutnya)
                saat_ini = saat_ini.selanjutnya
            akhir = saat_ini

daftar_simpul = DaftarSimpul()

while True:
    print("\nMenu:")
    print("1. Tambah Data")
    print("2. Cetak Data")
    print("3. Hapus Data")
    print("4. Ubah Data")
    print("5. Urut Data")
    print("6. Cari Data")
    print("7. Keluar")

    pilihan = int(input("Pilihan Anda: "))

    if pilihan == 1:
        nim = int(input("\nMasukkan NIM: "))
        nama = input("Masukkan Nama: ")
        prodi = input("Masukkan Prodi: ")
        nilai = int(input("Masukkan Nilai: "))
        daftar_simpul.tambah_data(nim, nama, prodi, nilai)
    elif pilihan == 2:
        daftar_simpul.cetak_data()

```

```

elif pilihan == 3:
    nim = int(input("Masukkan NIM yang Akan Dihapus: "))
    daftar_simpul.hapus_data(nim)
elif pilihan == 4:
    nim = int(input("Masukkan NIM yang Akan Diubah: "))
    nama = input("Masukkan Nama Baru: ")
    prodi = input("Masukkan Prodi Baru: ")
    nilai = int(input("Masukkan Nilai Baru: "))
    daftar_simpul.ubah_data(nim, nama, prodi, nilai)
elif pilihan == 5:
    daftar_simpul.urut_data()
    print("Data berhasil diurutkan.")
elif pilihan == 6:
    nim = int(input("Masukkan NIM yang Akan Dicari: "))
    data = daftar_simpul.cari_data(nim)
    if data is not None:
        print("Data ditemukan - NIM: {}, Nama: {}, Prodi: {}, Nilai:
{}".format(data.nim, data.nama, data.prodi, data.nilai))
    else:
        print("Data dengan NIM tersebut tidak ditemukan.")
elif pilihan == 7:
    print("\nTerima kasih telah menggunakan program ini!\n")
    break
else:
    print("Pilihan tidak valid. Silakan pilih menu yang tersedia.")

```

C. LATAR BELAKANG PROGRAM

Program mengurutkan data berdasarkan nilai tertinggi ke nilai terendah ini saya pilih karena program ini dapat memudahkan dalam mengurutkan nilai tertinggi ke terendah baik ketika ada pengeditan data, penambahan data, penghapusan data atau tidak tanpa menginput datanya secara manual.

D. PENJELASAN PROGRAM SECARA UMUM

Program berisikan :

- Penambahan data
- Pengeditan data
- Penghapusan data
- Menampilkan data
- Bisa mengurutkan data berdasarkan nilai tertinggi ke nilai terendah
- Bisa mencari data berdasarkan NIM

Ketika program dijalankan maka diminta input menu 1-7 yaitu :

1. Tambah Data
2. Cetak Data
3. Hapus Data
4. Edit Data
5. Urut Data
6. Cari Data
7. Keluar

Tapi, ketika input menu tak sesuai (1-7) maka akan menampilkan “*Pilihan tidak valid. Silahkan pilih menu yang tersedia.*”

Jika memilih input menu 1, maka akan diminta mengisi beberapa data, yaitu :

- Memasukkan NIM
- Memasukkan Nama
- Memasukkan Prodi
- Memasukkan Nilai

Jika memilih input menu 2, maka akan menampilkan hasil dari data yang sudah ditambah / diinputkan. Tapi jika belum ada penginputan data / data kosong maka program akan menampilkan “Data masih kosong”.

Jika memilih input menu 3, maka akan menghapus data. Disini, program sebelum menghapus data akan meminta input data NIM untuk memastikan data mana yang akan dihapus. Jika data tak tersedia / kosong maka program akan menampilkan “Data dengan NIM tersebut tidak ditemukan”.

Jika memilih input menu 4, maka akan mengedit data yang sebelumnya akan diminta input data berupa NIM untuk memastikan data mana yang akan diedit. Jika data tersedia maka diminta :

- Memasukkan Nama Baru
- Memasukkan Prodi Baru
- Memasukkan Nilai Baru

Jika sudah akan menampilkan “Data berhasil diubah”.

Jika memilih input menu 5, maka akan mengurutkan data berdasarkan nilai tertinggi ke nilai terendah dan jika ingin mengeceknya maka pilih input menu 2.

Jika memilih input menu 6, maka akan mencari data dengan memasukkan NIM yang akan dicari. Jika data ditemukan maka akan menampilkan data yang sesuai dengan input NIM, tapi jika data tidak ditemukan maka program akan menampilkan “Data dengan NIM tersebut tidak ditemukan”

Jika memilih input menu 7, maka akan keluar dari program dan menampilkan “Terima kasih telah menggunakan program ini!”

E. PENJELASAN BARIS PROGRAM

```
print("\n|    YAZID AQIL ASSALAM    |")
print("|    22343014    |")
print("|    Praktikum Struktur Data    |")
print("|    202223430028    |")
```

Baris ini mencetak informasi pembuat program

```
class Simpul:
    def __inisial__(self, nim, nama, prodi, nilai):
        self.nim = nim
        self.nama = nama
        self.prodi = prodi
        self.nilai = nilai
        self.selanjutnya = None
```

Baris ini mendefinisikan kelas Simpul yang merepresentasikan simpul atau node dalam linked list. Setiap simpul memiliki atribut NIM, nama, program studi (prodi), nilai, dan atribut selanjutnya yang menunjukkan simpul berikutnya dalam linked list.

```
class DaftarSimpul:
    def __inisial__(self):
        self.kepala = None
```

Baris ini mendefinisikan kelas DaftarSimpul yang merepresentasikan linked list itu sendiri. Kelas ini memiliki atribut kepala yang menunjukkan simpul pertama dalam linked list

```
def tambah_data(self, nim, nama, prodi, nilai):
    simpul = Simpul(nim, nama, prodi, nilai)
    if self.kepala is None:
        self.kepala = simpul
    else:
        saat_ini = self.kepala
        while saat_ini.selanjutnya is not None:
            saat_ini = saat_ini.selanjutnya
        saat_ini.selanjutnya = simpul
    print("Data berhasil ditambahkan")
```

Metode `tambah_data` pada kelas `DaftarSimpul` digunakan untuk menambahkan simpul baru ke dalam linked list. Jika linked list masih kosong, simpul baru tersebut akan menjadi kepala. Jika tidak, metode akan mencari simpul terakhir dalam linked list dan menambahkan simpul baru di belakang simpul terakhir tersebut.

```
def cetak_data(self):
    if self.kepala is None:
        print("Data masih kosong")
    else:
        saat_ini = self.kepala
        print("\nData dalam daftar:")
        while saat_ini is not None:
            print("NIM: {}, Nama: {}, Prodi: {}, Nilai: {}".format(saat_ini.nim, saat_ini.nama, saat_ini.prodi, saat_ini.nilai))
            saat_ini = saat_ini.selanjutnya
```

Metode `cetak_data` digunakan untuk mencetak seluruh data yang ada dalam linked list. Metode ini akan mengecek apakah linked list kosong atau tidak, kemudian mencetak setiap atribut (NIM, nama, prodi, dan nilai) dari setiap simpul dalam linked list.

```
def cari_data(self, nim):
    saat_ini = self.kepala
    while saat_ini is not None:
        if saat_ini.nim == nim:
            return saat_ini
        saat_ini = saat_ini.selanjutnya
    return None
```

Metode `cari_data` digunakan untuk mencari simpul dengan NIM yang diberikan dalam linked list. Metode ini akan melakukan iterasi melalui seluruh simpul dalam linked list dan mengembalikan simpul dengan NIM yang cocok jika ditemukan, atau `None` jika tidak ditemukan.

```
def ubah_data(self, nim, nama, prodi, nilai):
    simpul = self.cari_data(nim)
    if simpul is not None:
        simpul.nama = nama
        simpul.prodi = prodi
        simpul.nilai = nilai
        print("Data berhasil diubah")
    else:
        print("Data dengan NIM tersebut tidak ditemukan")
```

Metode `ubah_data` digunakan untuk mengubah data (nama, prodi, dan nilai) dari simpul dengan NIM yang diberikan dalam linked list. Metode ini menggunakan metode `cari_data` untuk mencari simpul dengan NIM yang cocok. Jika simpul ditemukan, data pada simpul tersebut akan diubah. Jika tidak ditemukan, akan dicetak pesan bahwa data tidak ditemukan.

```
def hapus_data(self, nim):
    saat_ini = self.kepala
    sebelumnya = None
    while saat_ini is not None:
        if saat_ini.nim == nim:
            if sebelumnya is None:
                self.kepala = saat_ini.selanjutnya
            else:
                sebelumnya.selanjutnya = saat_ini.selanjutnya
            print("Data berhasil dihapus")
            return
        sebelumnya = saat_ini
        saat_ini = saat_ini.selanjutnya
    print("Data dengan NIM tersebut tidak ditemukan")
```

Metode `hapus_data` digunakan untuk menghapus simpul dengan NIM yang diberikan dalam linked list. Metode ini akan mencari simpul dengan NIM yang cocok dan menghapusnya dari linked list. Jika simpul ditemukan, simpul sebelumnya akan terhubung dengan simpul setelahnya, sehingga simpul tersebut tidak lagi terhubung dalam linked list. Jika tidak ditemukan, akan dicetak pesan bahwa data tidak ditemukan.

```
def tukar_data(self, a, b):
    temp_nilai = a.nilai
    temp_nim = a.nim
    temp_nama = a.nama
    temp_prodi = a.prodi

    a.nilai = b.nilai
    a.nim = b.nim
    a.nama = b.nama
    a.prodi = b.prodi

    b.nilai = temp_nilai
    b.nim = temp_nim
    b.nama = temp_nama
    b.prodi = temp_prodi
```

Metode `tukar_data` digunakan untuk menukar data antara dua simpul. Metode ini mengambil dua simpul sebagai argumen dan menukar semua atribut (NIM, nama, prodi, dan nilai) di antara keduanya.

```
def urut_data(self):
    if self.kepala is None:
        return
    akhir = None
    while akhir != self.kepala:
        saat_ini = self.kepala
        while saat_ini.selanjutnya != akhir:
            if saat_ini.nilai < saat_ini.selanjutnya.nilai:
                self.tukar_data(saat_ini, saat_ini.selanjutnya)
            saat_ini = saat_ini.selanjutnya
        akhir = saat_ini
```

Metode `urut_data` digunakan untuk mengurutkan data dalam linked list berdasarkan nilai mahasiswa secara menurun. Metode ini menggunakan algoritma pengurutan bubble sort untuk melakukan pengurutan. Data akan diurutkan dengan membandingkan nilai simpul saat ini dengan nilai simpul berikutnya. Jika nilai saat ini lebih kecil, maka akan dilakukan pertukaran data menggunakan metode `tukar_data`.

```
daftar_simpul = DaftarSimpul()
```

Baris ini membuat objek `daftar_simpul` dari kelas `DaftarSimpul`. Objek ini akan digunakan untuk memanipulasi linked list.

```
while True:
    print("\nMenu:")
    print("1. Tambah Data")
    print("2. Cetak Data")
    print("3. Hapus Data")
    print("4. Ubah Data")
    print("5. Urut Data")
    print("6. Cari Data")
    print("7. Keluar")
```

Kode ini menginisialisasi loop `while` yang akan terus berjalan selama kondisi `True`. Setiap kali iterasi loop, program akan mencetak menu pilihan kepada pengguna. Menu ini memiliki tujuh opsi yang dapat dipilih oleh pengguna. Setiap opsi memiliki nomor yang terkait dengannya. Berikut adalah penjelasan singkat tentang setiap opsi:

1. Tambah Data: Opsi ini memungkinkan pengguna untuk menambahkan data ke dalam linked list.
2. Cetak Data: Opsi ini mencetak seluruh data yang ada dalam linked list.
3. Hapus Data: Opsi ini memungkinkan pengguna untuk menghapus data berdasarkan NIM yang dimasukkan.
4. Ubah Data: Opsi ini memungkinkan pengguna untuk mengubah data berdasarkan NIM yang dimasukkan.
5. Urut Data: Opsi ini akan mengurutkan data dalam linked list berdasarkan nilai secara menurun.
6. Cari Data: Opsi ini memungkinkan pengguna untuk mencari data berdasarkan NIM yang dimasukkan.
7. Keluar: Opsi ini akan mengakhiri program dan keluar dari loop `while`.

Setelah mencetak menu, program akan menunggu input dari pengguna untuk memilih salah satu opsi menu. Pengguna harus memasukkan nomor yang sesuai dengan opsi yang ingin mereka pilih.

```
pilihan = int(input("Pilihan Anda: "))
```

Baris ini meminta pengguna untuk memasukkan pilihan mereka dengan menggunakan fungsi `input`. Input yang diterima akan dikonversi menjadi tipe data integer menggunakan fungsi `int()` dan disimpan dalam variabel `pilihan`.

```
if pilihan == 1:
    nim = int(input("\nMasukkan NIM: "))
    nama = input("Masukkan Nama: ")
    prodi = input("Masukkan Prodi: ")
    nilai = int(input("Masukkan Nilai: "))
    daftar_simpul.tambah_data(nim, nama, prodi, nilai)
```

Jika pilihan yang dimasukkan pengguna adalah 1, program akan meminta pengguna untuk memasukkan NIM, nama, prodi, dan nilai mahasiswa baru menggunakan fungsi `input`. Nilai NIM dan nilai akan dikonversi menjadi tipe data integer menggunakan fungsi `int()`. Kemudian, metode `tambah_data` dari objek `daftar_simpul` akan dipanggil dengan argumen yang sesuai untuk menambahkan data mahasiswa baru ke linked list.

```
elif pilihan == 2:
    daftar_simpul.cetak_data()
```

Jika pilihan adalah 2, program akan memanggil metode `cetak_data` dari objek `daftar_simpul` untuk mencetak seluruh data mahasiswa yang ada dalam linked list.


```
elif pilihan == 3:
    nim = int(input("Masukkan NIM yang Akan Dihapus: "))
    daftar_simpul.hapus_data(nim)
```

Jika pilihan adalah 3, program akan meminta pengguna untuk memasukkan NIM mahasiswa yang akan dihapus menggunakan fungsi input. NIM akan dikonversi menjadi tipe data integer menggunakan fungsi int(). Kemudian, metode hapus_data dari objek daftar_simpul akan dipanggil dengan NIM yang sesuai untuk menghapus data mahasiswa dengan NIM tersebut dari linked list.

```
elif pilihan == 4:
    nim = int(input("Masukkan NIM yang Akan Diubah: "))
    nama = input("Masukkan Nama Baru: ")
    prodi = input("Masukkan Prodi Baru: ")
    nilai = int(input("Masukkan Nilai Baru: "))
    daftar_simpul.ubah_data(nim, nama, prodi, nilai)
```

Jika pilihan adalah 4, program akan meminta pengguna untuk memasukkan NIM mahasiswa yang akan diubah, serta memasukkan nama, prodi, dan nilai baru menggunakan fungsi input. NIM dan nilai akan dikonversi menjadi tipe data integer menggunakan fungsi int(). Kemudian, metode ubah_data dari objek daftar_simpul akan dipanggil dengan argumen yang sesuai untuk mengubah data mahasiswa dengan NIM tersebut dalam linked list.

```
elif pilihan == 5:
    daftar_simpul.urut_data()
    print("Data berhasil diurutkan.")
```

Jika pilihan adalah 5, program akan memanggil metode urut_data dari objek daftar_simpul untuk mengurutkan data mahasiswa dalam linked list berdasarkan nilai. Setelah itu, program akan mencetak pesan bahwa data berhasil diurutkan.

```
elif pilihan == 6:
    nim = int(input("Masukkan NIM yang Akan Dicari: "))
    data = daftar_simpul.cari_data(nim)
    if data is not None:
        print("Data ditemukan - NIM: {}, Nama: {}, Prodi: {}, Nilai: {}".format(data.nim, data.nama, data.prodi, data.nilai))
    else:
        print("Data dengan NIM tersebut tidak ditemukan.")
```

Jika pilihan adalah 6, program akan meminta pengguna untuk memasukkan NIM mahasiswa yang akan dicari menggunakan fungsi input. NIM akan dikonversi menjadi tipe data integer menggunakan fungsi int(). Kemudian, metode cari_data dari objek daftar_simpul akan dipanggil dengan NIM yang sesuai untuk mencari data mahasiswa dengan NIM tersebut dalam linked list. Jika data ditemukan, program akan mencetak informasi tentang data mahasiswa. Jika data tidak ditemukan, akan dicetak pesan bahwa data dengan NIM tersebut tidak ditemukan.

```
elif pilihan == 7:
    print("\nTerima kasih telah menggunakan program ini!\n")
    break
```

Jika pilihan adalah 7, program akan mencetak pesan terima kasih dan kemudian keluar dari loop utama menggunakan pernyataan break, mengakhiri program.

```
else:
    print("Pilihan tidak valid. Silakan pilih menu yang tersedia.")
```

Jika pilihan yang dimasukkan tidak valid (tidak ada dalam rentang 1-7), program akan mencetak pesan bahwa pilihan tidak valid dan meminta pengguna untuk memilih menu yang tersedia.