

```
# List untuk menyimpan semua data mahasiswa (tipe tuple)
mahasiswa_list = [] // dalam list ini akan berisi seperti ("123", "Andi", "Informatika", 4, 3.8)
```

Jawaban untuk pertanyaan no 1

Sebelum menambahkan mahasiswa baru, harus dicek apakah NIM tersebut sudah ada di daftar.

```
# Fungsi untuk mengecek apakah NIM sudah ada
```

```
def is_nim_unique(nim, data):
```

```
    return all(mhs[0] != nim for mhs in data)
```

lalu gunakan tuple untuk menyimpan data mahasiswa. Karena tuple bersifat immutable, isinya (termasuk NIM) tidak bisa diubah setelah dimasukkan.

Jawaban untuk pertanyaan no 2

```
# Fungsi untuk menambahkan mahasiswa
```

```
def tambah_mahasiswa(data, nim, nama, jurusan, semester, ipk):
```

```
    if not is_nim_unique(nim, data):
```

```
        print("NIM sudah terdaftar!")
```

```
    return
```

```
    if not (1 <= semester <= 14) or not (0.0 <= ipk <= 4.0):
```

```
        print("Semester/IPK tidak valid!")
```

```
    return
```

```
    mahasiswa = (nim, nama, jurusan, semester, ipk)
```

```
    data.append(mahasiswa)
```

```
    print("Mahasiswa berhasil ditambahkan.")
```

```
#Fungsi mencari mahasiswa berdasarkan Nim atau Jurusan
```

```
def cari_berdasarkan_nim(data, nim):
```

```
    return next((mhs for mhs in data if mhs[0] == nim), None)
```

```
def cari_berdasarkan_jurusan(data, jurusan):
```

```
    return [mhs for mhs in data if mhs[2].lower() == jurusan.lower()]
```

```
#Fungsi menghitung rata rata ipk per jurusan
```

```
def rata_rata_ipk(data, jurusan):
```

```
jurusan_mahasiswa = cari_berdasarkan_jurusan(data, jurusan)
```

```
if not jurusan_mahasiswa:
```

```
    return 0
```

```
total_ipk = sum(mhs[4] for mhs in jurusan_mahasiswa)
```

```
return total_ipk / len(jurusan_mahasiswa)
```

#Fungsi Mengurutkan mahasiswa berdasarkan IPK (descending)

```
def ambil_ipk(mhs):
```

```
    return mhs[4]
```

```
def urutkan_berdasarkan_ipk(data):
```

```
    return sorted(data, key=ambil_ipk, reverse=True)
```

Jawaban untuk pertanyaan no 3

Keuntungan Tuple:

- Tuple bersifat Immutable (tidak bisa diubah) jadi NIM dan data lainnya tidak akan berubah setelah dicatat.
- Tuple lebih efisien dalam memori dibanding list.

Jawaban untuk pertanyaan no 4

Penggunaan tuple untuk menyimpan data mahasiswa lebih menguntungkan dibandingkan list karena:

- Data Tidak Bisa Diubah (Immutable):
Tuple bersifat tetap setelah dibuat. Ini cocok untuk data penting seperti NIM, nama, jurusan, semester, dan IPK yang tidak boleh diubah setelah dicatat.
- Lebih Aman:
Karena datanya tidak bisa diubah, risiko kesalahan (seperti mengganti NIM secara tidak sengaja) dapat dicegah.
- Lebih Efisien:
Tuple lebih ringan dan cepat diproses dibandingkan list, sehingga lebih efisien dalam penggunaan memori.