Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 6)

bismillahirrahmanirrahiim adalah modal utama kita dalam memulai segala aktivitas, dengan buku kamus modul ini insyaaALLAH jika dibaca dan dipahami penjelasan dan alur programnya dapat dimengerti.

1. Menampilkan nama-nama hari secara mendatar.

```
nama_hari = ["Senin," "Selasa," "Rabu," "Kamis," "Jum'at," "Sabtu," "dan
Minggu."]

for hari in nama_hari:
    print(hari, end=" ")
```

Penjelasan:
-Baris pertama mendefinisikan sebuah list bernama nama_hari, yang berisi nama-nama hari dalam bahasa Indonesia. Setiap nama hari disimpan sebagai elemen dalam list, yang diapit oleh tanda kutip ganda.
-Baris kedua memulai loop for, dimana variabel hari akan mengambil nilai dari setiap elemen dalam nama_hari.
-Di dalam loop, baris ketiga mencetak nilai variabel hari, dan menggunakan parameter end=" " untuk mencegah pergantian baris otomatis setelah mencetak satu elemen.
-Loop akan terus berjalan hingga semua elemen dalam nama_hari diproses dan dicetak.
Hasil dari program ini adalah mencetak semua nama hari dalam satu baris, dipisahkan oleh spasi, seperti: "Senin Selasa Rabu Kamis Jum'at Sabtu Minggu".

2. Menampilkan nama hari secara menurun.

```
nama_hari = ["Senin", "Selasa", "Rabu", "Kamis", "Jumat", "Sabtu", "Minggu"]

# Menampilkan nama hari secara menurun dalam bentuk kolom
for hari in reversed(nama_hari):
    print(hari)
```

Penjelasan:
-Baris pertama mendefinisikan sebuah list bernama nama_hari, yang berisi nama-nama hari dalam bahasa Indonesia. Setiap nama hari disimpan sebagai elemen dalam list, diapit oleh tanda kutip ganda.
-Baris keempat memulai loop for, dimana variabel hari akan mengambil nilai dari setiap elemen dalam nama_hari.
-Di dalam loop, baris kelima mencetak nilai variabel hari, dan karena kita menggunakan fungsi reversed(nama_hari), maka nilai-nilai akan dicetak secara terbalik, dimulai dari "Minggu" hingga "Senin".
-Setelah mencetak semua elemen, loop akan berakhir dan program selesai.
Hasil dari program ini adalah mencetak daftar nama-nama hari dalam bentuk kolom, dimulai dari "Minggu" hingga "Senin". Setiap nama hari akan dicetak pada baris terpisah.

3. Menghitung luas dan keliling persegi.

```
class Persegi:
    def __init__(self, sisi):
        self.sisi = sisi

    def hitung_luas(self):
        luas = self.sisi ** 2
        return luas

    def hitung_keliling(self):
        keliling = 4 * self.sisi
        return keliling

# Membuat objek Persegi
sisi_persegi = float(input("Masukkan panjang sisi persegi: "))
persegi = Persegi(sisi_persegi)

# Menghitung luas dan keliling
luas = persegi.hitung_luas()
keliling = persegi.hitung_keliling()

print("Luas persegi:", luas)
print("Keliling persegi:", keliling)
```

Penjelasan:
-Program ini mendefinisikan sebuah kelas bernama Persegi yang memiliki tiga metode: __init__, hitung_luas, dan hitung_keliling.
-__init__ adalah konstruktor kelas yang menginisialisasi objek dengan nilai sisi yang diberikan saat objek dibuat.
-hitung_luas dan hitung_keliling adalah metode yang menghitung luas dan keliling persegi berdasarkan nilai sisi yang ada dalam objek.
-Pada baris terakhir program, nantinya pengguna diminta memasukkan panjang sisi persegi. Objek persegi dibuat dengan memanggil kelas Persegi dan memberikan nilai sisi yang diinput.
-Selanjutnya, program menghitung luas dan keliling persegi dengan memanggil metode-metode yang ada pada objek persegi.
-Hasil luas dan keliling dicetak menggunakan pernyataan print.
Hasil program ini adalah mencetak luas dan keliling dari persegi berdasarkan nilai sisi yang diinput oleh pengguna.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 7)

1. Mencari luas segitiga.

```python
class Segitiga:
    def __init__(self, alas, tinggi):
        self.alas = alas
        self.tinggi = tinggi

    def hitung_luas(self):
        luas = 0.5 * self.alas * self.tinggi
        return luas

# Membuat objek Segitiga
alas_segitiga = float(input("Masukkan panjang alas segitiga: "))
tinggi_segitiga = float(input("Masukkan tinggi segitiga: "))
segitiga = Segitiga(alas_segitiga, tinggi_segitiga)

# Menghitung dan menampilkan luas segitiga
luas = segitiga.hitung_luas()
print("Luas segitiga:", luas)
```

Penjelasan:

-Program ini mendefinisikan sebuah kelas bernama Segitiga yang memiliki konstruktor __init__ dan metode hitung_luas.

-Konstruktor __init__ digunakan untuk menginisialisasi objek dengan dua parameter yaitu alas dan tinggi, yang merepresentasikan panjang alas dan tinggi segitiga.

-Metode hitung_luas menghitung luas segitiga berdasarkan rumus luas segitiga yaitu 0.5 * alas * tinggi.

-Nantinya pengguna akan diminta memasukkan panjang alas dan tinggi segitiga. Objek segitiga dibuat dengan memanggil kelas Segitiga dan memberikan nilai alas dan tinggi yang diinput.

-Selanjutnya, program menghitung luas segitiga dengan memanggil metode hitung_luas pada objek segitiga.

-Hasil luas segitiga dicetak menggunakan pernyataan print.

Hasil dari program ini adalah mencetak luas segitiga berdasarkan nilai alas dan tinggi yang diinput oleh pengguna.

2. Mengkonversi suhu dari Celcius ke Fahrenheit dan Reamur.

```
class KonversiSuhu:
    def celcius_ke_fahrenheit(self, suhu_celcius):
        suhu_fahrenheit = (suhu_celcius * 9/5) + 32
        return suhu_fahrenheit

    def celcius_ke_reamur(self, suhu_celcius):
        suhu_reamur = suhu_celcius * 4/5
        return suhu_reamur

# Membuat objek KonversiSuhu
konversi = KonversiSuhu()

# Meminta pengguna untuk memasukkan suhu dalam Celcius
suhu_celcius = float(input("Masukkan suhu dalam Celcius: "))

# Menghitung dan menampilkan hasil konversi
suhu_fahrenheit = konversi.celcius_ke_fahrenheit(suhu_celcius)
suhu_reamur = konversi.celcius_ke_reamur(suhu_celcius)

print(f"{suhu_celcius} Celcius sama dengan {suhu_fahrenheit} Fahrenheit")
print(f"{suhu_celcius} Celcius sama dengan {suhu_reamur} Reamur")
```

Penjelasan:

-Program ini mendefinisikan kelas KonversiSuhu yang memiliki dua metode: celcius_ke_fahrenheit dan celcius_ke_reamur.

-Metode celcius_ke_fahrenheit digunakan untuk mengkonversi suhu dari Celcius ke Fahrenheit berdasarkan rumus yang ada.

-Metode celcius_ke_reamur digunakan untuk mengkonversi suhu dari Celcius ke Reamur berdasarkan rumus yang ada.

-Objek konversi dibuat berdasarkan kelas KonversiSuhu.

-Nantinya pengguna diminta memasukkan suhu dalam Celcius.

-Program akan melakukan perhitungan konversi suhu dengan memanggil metode-metode yang ada pada objek konversi.

-Hasil konversi suhu dari Celcius ke Fahrenheit dan Reamur dicetak menggunakan pernyataan print.

Hasil dari program ini adalah mencetak hasil konversi suhu dari Celcius ke Fahrenheit dan Reamur berdasarkan suhu yang diinput oleh pengguna.

3. Menjumlahkan harga barang yang dibeli.

```
class Pembelian:
    def __init__(self):
        self.daftar_barang = {}

    def tambah_barang(self, nama, harga, jumlah):
        self.daftar_barang[nama] = {"harga": harga, "jumlah": jumlah}

    def hitung_total(self):
        total = 0
        for barang in self.daftar_barang.values():
            total += barang["harga"] * barang["jumlah"]
        return total

# Membuat objek Pembelian
pembelian = Pembelian()

# Menambahkan barang-barang ke dalam daftar pembelian
while True:
    nama_barang = input("Masukkan nama barang (kosongkan jika selesai): ")
    if not nama_barang:
        break
    harga_barang = float(input("Masukkan harga barang: "))
    jumlah_barang = int(input("Masukkan jumlah barang: "))

    pembelian.tambah_barang(nama_barang, harga_barang, jumlah_barang)

# Menghitung dan menampilkan total harga pembelian
total_harga = pembelian.hitung_total()
print("Total harga pembelian:", total_harga)
```

Penjelasan:

-Program ini mendefinisikan kelas Pembelian yang memiliki metode __init__, tambah_barang, dan hitung_total.

-Metode __init__ digunakan untuk menginisialisasi objek Pembelian dengan membuat dictionary daftar_barang untuk mencatat barang yang dibeli.

-Metode tambah_barang digunakan untuk menambahkan barang ke dalam dictionary daftar_barang dengan menyimpan informasi harga dan jumlah barang.

-Metode hitung_total digunakan untuk menghitung total harga pembelian berdasarkan harga dan jumlah setiap barang dalam dictionary.

-Objek pembelian dibuat berdasarkan kelas Pembelian.
-Natinya pengguna akan diminta memasukkan nama, harga, dan jumlah barang secara berulang sampai selesai.
-Program menggunakan metode tambah_barang untuk mencatat informasi barang yang dimasukkan oleh pengguna.
-Setelah selesai memasukkan barang, program menggunakan metode hitung_total untuk menghitung total harga pembelian.
-Hasil total harga pembelian dicetak menggunakan pernyataan print.
Hasil dari program ini adalah mencatat pembelian barang yang dimasukkan oleh pengguna dan menghitung total harga pembelian berdasarkan harga dan jumlah barang yang dimasukkan.

4. Menghitung nilai akhir mata kuliah dari 2 nilai kuis, 3 nilai tugas, nilai UTS dan UAS, berdasarkan rata-rata.

```
class NilaiMataKuliah:
    def __init__(self, nama, nim):
        self.nama = nama
        self.nim = nim
        self.nilai_kuis = []
        self.nilai_tugas = []
        self.nilai_uts = 0
        self.nilai_uas = 0

    def tambah_nilai_kuis(self, nilai):
        self.nilai_kuis.append(nilai)

    def tambah_nilai_tugas(self, nilai):
        self.nilai_tugas.append(nilai)

    def tambah_nilai_uts(self, nilai):
        self.nilai_uts = nilai

    def tambah_nilai_uas(self, nilai):
        self.nilai_uas = nilai

    def hitung_nilai_akhir(self):
        rata_nilai_kuis = sum(self.nilai_kuis) / len(self.nilai_kuis)
        rata_nilai_tugas = sum(self.nilai_tugas) / len(self.nilai_tugas)
```

```
        nilai_akhir = 0.2 * rata_nilai_kuis + 0.3 * rata_nilai_tugas + 0.2 *
self.nilai_uts + 0.3 * self.nilai_uas
        return nilai_akhir

# Membuat objek NilaiMataKuliah
nama_mahasiswa = input("Masukkan nama mahasiswa: ")
nim_mahasiswa = input("Masukkan NIM mahasiswa: ")
nilai_mata_kuliah = NilaiMataKuliah(nama_mahasiswa, nim_mahasiswa)

# Memasukkan nilai-nilai
for i in range(2):
    nilai_kuis = float(input(f"Masukkan nilai kuis ke-{i+1}: "))
    nilai_mata_kuliah.tambah_nilai_kuis(nilai_kuis)

for i in range(3):
    nilai_tugas = float(input(f"Masukkan nilai tugas ke-{i+1}: "))
    nilai_mata_kuliah.tambah_nilai_tugas(nilai_tugas)

nilai_uts = float(input("Masukkan nilai UTS: "))
nilai_mata_kuliah.tambah_nilai_uts(nilai_uts)

nilai_uas = float(input("Masukkan nilai UAS: "))
nilai_mata_kuliah.tambah_nilai_uas(nilai_uas)

# Menghitung dan menampilkan nilai akhir
nilai_akhir = nilai_mata_kuliah.hitung_nilai_akhir()
print(f"Nama: {nilai_mata_kuliah.nama}")
print(f"NIM: {nilai_mata_kuliah.nim}")
print(f"Nilai Akhir: {nilai_akhir:.2f}")
```

Penjelasan:

-Program mendefinisikan kelas NilaiMataKuliah yang memiliki metode untuk mengelola nilai-nilai mahasiswa dan menghitung nilai akhir.

-Metode __init__ digunakan untuk menginisialisasi objek NilaiMataKuliah dengan parameter nama dan nim, serta menginisialisasi nilai-nilai awal.

-Metode-metode tambah_nilai_kuis, tambah_nilai_tugas, tambah_nilai_uts, dan tambah_nilai_uas digunakan untuk memasukkan nilai-nilai kuis, tugas, UTS, dan UAS.

-Metode hitung_nilai_akhir menghitung nilai akhir berdasarkan rata-rata nilai kuis dan tugas, serta nilai UTS dan UAS dengan bobot tertentu.

-Objek nilai_mata_kuliah dibuat berdasarkan kelas NilaiMataKuliah.

-Nantinya pengguna diminta memasukkan nama dan NIM mahasiswa.

-Nantinnya pengguna diminta memasukkan nilai-nilai kuis, tugas, UTS, dan UAS.
-Metode-metode tambah_nilai_... digunakan untuk mencatat nilai-nilai yang dimasukkan oleh pengguna.
-Metode hitung_nilai_akhir digunakan untuk menghitung nilai akhir berdasarkan nilai-nilai yang telah dimasukkan.
-Hasil nama, NIM, dan nilai akhir dicetak menggunakan pernyataan print.
Program ini akan menghitung dan menampilkan nilai akhir mahasiswa berdasarkan nilai-nilai yang dimasukkan oleh pengguna.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 8)

1. Tuliskan algoritma untuk menampilkan n buah bilangan asli pertama, kemudian tulis kodenya dalam python. (angka n diinput user dari keyboard)

```
n = int(input("Masukkan nilai n: "))  # Langkah 1
bilangan = 1  # Langkah 2

while n > 0:  # Langkah 3
    print(bilangan)  # Langkah 3a
    bilangan += 1  # Langkah 3b
    n -= 1  # Langkah 3c
```

Program tersebut merupakan contoh implementasi dari sebuah perulangan (loop) menggunakan while untuk mencetak sejumlah bilangan teratur dari 1 hingga n.
-Pengguna diminta memasukkan nilai n.
-Variabel bilangan diinisialisasi dengan nilai 1.
-Perulangan dilakukan selama nilai n lebih dari 0:
a. Pernyataan print(bilangan) mencetak nilai bilangan.
b. bilangan ditambah 1, sehingga akan mencetak bilangan selanjutnya pada perulangan berikutnya.
c. Nilai n dikurangi 1, sehingga perulangan akan berhenti ketika n mencapai 0 atau lebih kecil.
Program ini akan mencetak bilangan-bilangan dari 1 hingga n sesuai dengan nilai yang dimasukkan oleh pengguna. Setiap bilangan akan dicetak satu per satu dalam baris terpisah.

2. analisis bagian-bagian program berikut dan jelaskan apa yang dilakukannya.

```
print("Program Pengulangan for Angka")
print("----------------------------")
for j in range(1, 6):
    print("Bilangan ke-%i" % j)
    angka = int(input("Masukkan sebuah angka: "))
```

Dalam kode tersebut, program melakukan hal-hal berikut:
-Mencetak judul program dan pemisah garis.

-Melakukan perulangan for dengan variabel j yang mengambil nilai dari 1 hingga 5 (6 tidak termasuk).
-Di dalam perulangan, mencetak pesan "Bilangan ke-x" dengan nilai x yang didapatkan dari variabel j.
-Meminta pengguna memasukkan sebuah angka dan menyimpannya dalam variabel angka.
Dengan demikian, setiap iterasi perulangan akan mencetak pesan "Bilangan ke-x" dan meminta pengguna memasukkan angka. Ini akan terjadi sebanyak 5 kali karena perulangan mengambil nilai dari 1 hingga 5. Setiap angka yang dimasukkan oleh pengguna akan disimpan dalam variabel angka.

3. analisis bagian-bagian program berikut dan jelaskan apa yang dilakukannya.

```
print("Program Pengulangan Menurun")
print("-----------------------------")
y = 5
while y >= 1:
    print("%i " % y)
    y -= 1  # artinya y=y-1 dan program output dimulai dari 5
```

Dalam kode tersebut, program melakukan hal-hal berikut:
-Mencetak judul program dan pemisah garis.
-Menginisialisasi variabel y dengan nilai 5.
-Melakukan perulangan while selama nilai y lebih besar atau sama dengan 1.
-Di dalam perulangan, mencetak nilai y dengan format yang diberikan.
-Mengurangi nilai y sebesar 1 setiap kali iterasi selesai, sehingga mengurutkan nilai y dari 5 hingga 1.
-Hasil dari kode ini adalah mencetak angka-angka dari 5 hingga 1 secara menurun.

4. Jelaskan bagian-bagian loop pada program berikut dan Angka berapakah yang harus diinput agar menampilkan “Program Selesai” untuk program ini

```
print("Program Pengulangan Input")
print("-------------------------")
bil = 1
while bil != 999:
    bil = int(input("Masukkan sebuah angka genap :"))
print("Program selesai")
```

Penjelasan:
Program tersebut mengajak pengguna untuk memasukkan angka genap secara berulang sampai pengguna memasukkan angka 999. Dapat kita analisis bagian-bagian program ini diantaranya:
-print("Program Pengulangan Input"): Ini adalah perintah untuk mencetak judul program "Program Pengulangan Input".
-print("-----------------------"): Ini adalah perintah untuk mencetak pemisah garis sebagai garis pembatas antara judul dan hasil output.
-bil = 1: Variabel bil diinisialisasi dengan nilai 1. Ini akan menjadi nilai awal sebelum pengguna memasukkan angka.
-while bil != 999:: Ini adalah awal dari perulangan while. Perulangan ini akan terus berjalan selama nilai bil tidak sama dengan 999.
-bil = int(input("Masukkan sebuah angka genap :")): Di dalam perulangan, program akan meminta pengguna memasukkan sebuah angka dan mengkonversi input tersebut menjadi tipe data integer menggunakan fungsi int(). Angka tersebut akan disimpan dalam variabel bil.
-print("Program selesai"): Setelah pengguna memasukkan angka 999, perulangan akan berhenti, dan program akan mencetak "Program selesai".
Sedemikian sehingga, program ini akan berjalan dalam loop hingga pengguna memasukkan angka 999. Setiap iterasi meminta pengguna memasukkan angka dan memeriksa apakah angka yang dimasukkan adalah genap atau bukan. Jika pengguna memasukkan angka genap, program akan terus berlanjut. Jika pengguna memasukkan angka 999, perulangan akan berhenti dan program
akan mencetak "Program selesai".

5. Jelaskan bagian-bagian loop pada program berikut dan Angka berapakah yang harus diinput agar menampilkan “Program Selesai” untuk program ini

```
print("Program Pengulangan Latihan Tambahan")
print("-----------------------------------")
lagi = 1
while lagi == 1:
    bil = int(input("Masukkan sebuah angka: "))
    lagi = int(input("Mau lagi? (1.Ya / 2.Tidak) :"))
print("Program Selesai")
```

Penjelasan:
Program tersebut adalah contoh penggunaan loop while untuk memungkinkan pengguna memasukkan angka secara berulang dan mengulang program jika pengguna ingin melakukannya. Dapat kita analisis bagian-bagian program ini:
-print("Program Pengulangan Latihan Tambahan"): Ini adalah perintah untuk mencetak judul program "Program Pengulangan Latihan Tambahan".
-print("----------"): Ini adalah perintah untuk mencetak pemisah garis sebagai garis pembatas antara judul dan hasil output.
-lagi = 1: Variabel lagi diinisialisasi dengan nilai 1. Ini digunakan untuk mengontrol apakah program akan diulang atau tidak.
-while lagi == 1:: Ini adalah awal dari perulangan while. Perulangan ini akan terus berjalan selama nilai lagi sama dengan 1.
-bil = int(input("Masukkan sebuah angka: ")): Di dalam perulangan, program akan meminta pengguna memasukkan sebuah angka dan mengkonversi input tersebut menjadi tipe data integer menggunakan fungsi int(). Angka tersebut akan disimpan dalam variabel bil.
-lagi = int(input("Mau lagi? (1.Ya / 2.Tidak) :")): Setelah pengguna memasukkan angka, program akan meminta pengguna apakah ingin mengulang program atau tidak. Jika pengguna memasukkan 1 (Ya), perulangan akan terus berlanjut. Jika pengguna memasukkan 2 (Tidak), perulangan akan berhenti.
-print("Program Selesai"): Setelah pengguna memilih untuk tidak mengulang program, perulangan akan berhenti, dan program akan mencetak "Program Selesai".

6. Buatlah algoritma dan kode python untuk Menampilkan angka yang lebih besar dari n dan lebih kecil dari m. (m dan n diinput oleh pengguna)

```
n = int(input("Masukkan nilai n: "))  # Langkah 1
m = int(input("Masukkan nilai m: "))  # Langkah 2

if n > m:  # Langkah 3
    n, m = m, n

angka = n + 1  # Langkah 4
```

```
print("Angka yang lebih besar dari", n, "dan lebih kecil dari", m, "adalah:")
while angka < m:  # Langkah 5
    print(angka)  # Langkah 5a
    angka += 1  # Langkah 5b
```

Penjelasan:

Program tersebut merupakan contoh implementasi dari sebuah program untuk mencetak angka yang lebih besar dari n dan lebih kecil dari m. Berikut penjelasannya:

-Pengguna diminta memasukkan nilai n.

-Pengguna diminta memasukkan nilai m.

-Jika n lebih besar dari m, maka nilai n dan m akan ditukar. Ini dilakukan untuk memastikan n selalu lebih kecil dari m.

-Variabel angka diinisialisasi dengan nilai n + 1.

Perulangan dilakukan selama nilai angka kurang dari m:

a. Pernyataan print(angka) mencetak nilai angka.

b. Nilai angka ditambah 1, sehingga akan mencetak angka selanjutnya pada perulangan berikutnya.

Program ini akan mencetak angka-angka yang lebih besar dari n dan lebih kecil dari m, asalkan n lebih kecil dari m. Setiap angka akan dicetak satu per satu dalam baris terpisah.

7. Buatlah algoritma dan kode python untuk Menampilkan Menghitung n! (n faktorial), dengan n diinput oleh pengguna.

```
n = int(input("Masukkan nilai n: "))  # Langkah 1

faktorial = 1  # Langkah 2
original_n = n  # Simpan nilai n untuk mencetak n! pada hasil

while n > 1:  # Langkah 3
    faktorial *= n  # Langkah 3a
    n -= 1  # Langkah 3b

print(f"{original_n}! =", faktorial)  # Langkah 4
```

Penjelasan:

Program tersebut merupakan contoh implementasi dari sebuah program untuk menghitung nilai faktorial dari suatu bilangan n. Berikut penjelasannya:

-Pengguna diminta memasukkan nilai n.
-Variabel faktorial diinisialisasi dengan nilai 1. Ini digunakan untuk menyimpan hasil faktorial.
-Perulangan dilakukan selama nilai n lebih dari 1:
a. Nilai faktorial dikalikan dengan nilai n, sehingga faktorial bertambah dengan faktor n.
b. Nilai n dikurangi 1, sehingga akan mengulang perulangan untuk nilai n yang lebih kecil.
-Setelah perulangan selesai, program akan mencetak hasil faktorial dari nilai original_n yang dimasukkan oleh pengguna.
Program ini akan menghitung dan mencetak nilai faktorial dari n dengan mengalikan n dengan semua bilangan positif lebih kecil dari n.

8. Buatlah algoritma dan kode python untuk Menampilkan Menghitung x^y dengan x dan y diinput oleh pengguna. (tanpa fungsi math.pow() atau operator “**”)

```
x = float(input("Masukkan nilai x: "))  # Langkah 1
y = int(input("Masukkan nilai y: "))    # Langkah 2

hasil = 1.0  # Langkah 3
pangkat = y  # Simpan nilai y untuk mencetak hasil

while y > 0:  # Langkah 4
    hasil *= x  # Langkah 4a
    y -= 1      # Langkah 4b

if pangkat > 1:  # Periksa apakah pangkat lebih dari 1
    print(f"{int(x)}^{pangkat} =", hasil)  # Cetak dengan notasi eksponen
else:
    print(f"{int(x)}^{pangkat} =", int(hasil))  # Cetak tanpa notasi eksponen
```

Program tersebut merupakan contoh implementasi dari sebuah program untuk menghitung nilai x^y. Berikut penjelasannya:
-Pengguna diminta memasukkan nilai x.
-Pengguna diminta memasukkan nilai y.
-Variabel hasil diinisialisasi dengan nilai 1.0. Ini digunakan untuk menyimpan hasil perpangkatan.

-Variabel pangkat diinisialisasi dengan nilai y. Ini digunakan untuk menyimpan nilai y yang akan dicetak nanti.
-Perulangan dilakukan selama nilai y lebih dari 0:
a. Nilai hasil dikalikan dengan nilai x, sehingga hasil akan menjadi hasil dari perpangkatan.
b. Nilai y dikurangi 1, sehingga akan mengulang perulangan untuk nilai y yang lebih kecil.
-Setelah perulangan selesai, program akan memeriksa apakah pangkat lebih dari 1. Jika iya:
a. Program akan mencetak hasil dalam notasi eksponen, misalnya 2^3 = 8.0.
-Jika pangkat tidak lebih dari 1, program akan mencetak hasil tanpa notasi eksponen, misalnya 2^2 = 4.
Program ini akan menghitung dan mencetak hasil dari perpangkatan nilai x dengan nilai y.

9. Buatlah algoritma dan kode python untuk Menampilkan Menghitung rata-rata 5 nilai yang diinput oleh pengguna.

```python
total = 0  # Langkah 1

for i in range(5):  # Langkah 2
    nilai = float(input(f"Masukkan nilai ke-{i+1}: "))  # Langkah 2a
    total += nilai  # Langkah 2b

rata_rata = total / 5  # Langkah 3

print("Rata-rata dari 5 nilai adalah:", rata_rata)  # Langkah 4
```

Program tersebut merupakan contoh implementasi dari sebuah program untuk menghitung rata-rata dari 5 nilai yang dimasukkan oleh pengguna. Berikut penjelasannya:
-Variabel total diinisialisasi dengan nilai 0. Ini akan digunakan untuk menghitung total dari nilai-nilai yang dimasukkan.
-Perulangan dilakukan menggunakan loop for dengan rentang range(5):
a. Pengguna diminta memasukkan nilai menggunakan input() dengan pesan yang sesuai.
b. Nilai yang dimasukkan oleh pengguna dijumlahkan ke dalam variabel total.

-Setelah perulangan selesai, program menghitung nilai rata-rata dengan membagi total dengan jumlah nilai (dalam hal ini, 5).
-Program mencetak nilai rata-rata dari 5 nilai yang dimasukkan oleh pengguna.
Program ini akan meminta pengguna untuk memasukkan 5 nilai, menghitung total nilai, menghitung rata-rata, dan mencetak hasil rata-rata tersebut.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 9)

1. Jelaskan dan analisis bagian-bagian apa yang dilakukannya.

```
print('Program Operasi Logika')
print('----------------------')
bil = int(input('Masukan sebuah bilangan : '))
if bil > 0 and bil < 10:
    print('Bilangan positif kurang dari 10.')
else:
    print('Bukan bilangan positif kurang dari 10.')
print('----------------------')
huruf = input('Masukan sebuah huruf : ')
if huruf == 'A' or huruf == 'a':
    print('Huruf A.')
else:
    print('Bukan huruf A.')
```

Program ini adalah contoh penggunaan operasi logika dan pernyataan kondisional (if-else) dalam Python. Dapat kita analisis bagian-bagian program ini:
-print('Program Operasi Logika'): Ini adalah perintah untuk mencetak judul program "Program Operasi Logika".
-print('----------------------'): Ini adalah perintah untuk mencetak pemisah garis sebagai garis pembatas antara judul dan hasil output.
-bil = int(input('Masukan sebuah bilangan : ')): Ini meminta pengguna memasukkan sebuah bilangan dan mengkonversi input menjadi tipe data integer menggunakan fungsi int(). Nilai bilangan tersebut disimpan dalam variabel bil.
-if bil > 0 and bil < 10:: Ini adalah pernyataan kondisional (if) yang memeriksa apakah nilai bilangan berada di antara 0 dan 10. Jika kondisi terpenuhi, blok kode di bawahnya akan dijalankan.
-print('Bilangan positif kurang dari 10.'): Ini adalah perintah yang akan dijalankan jika kondisi pada langkah sebelumnya benar.
-else:: Ini adalah bagian alternatif dari pernyataan kondisional (else) yang akan dijalankan jika kondisi pada langkah 4 tidak terpenuhi.

-print('Bukan bilangan positif kurang dari 10.'): Ini adalah perintah yang akan dijalankan jika kondisi pada langkah 4 tidak terpenuhi.
-print('----------'): Ini adalah perintah untuk mencetak pemisah garis sebagai garis pembatas antara hasil output pertama dan kedua.
-huruf = input('Masukan sebuah huruf : '): Ini meminta pengguna memasukkan sebuah huruf dan menyimpan input dalam variabel huruf.
-if huruf == 'A' or huruf == 'a':: Ini adalah pernyataan kondisional (if) yang memeriksa apakah input huruf adalah 'A' atau 'a'. Jika kondisi terpenuhi, blok kode di bawahnya akan dijalankan.
-print('Huruf A.'): Ini adalah perintah yang akan dijalankan jika kondisi pada langkah sebelumnya benar.
-else:: Ini adalah bagian alternatif dari pernyataan kondisional (else) yang akan dijalankan jika kondisi pada langkah 10 tidak terpenuhi.
-print('Bukan huruf A.'): Ini adalah perintah yang akan dijalankan jika kondisi pada langkah 10 tidak terpenuhi.
Program ini menggambarkan bagaimana kita dapat menggunakan operasi logika dan pernyataan kondisional dalam Python untuk melakukan pengujian dan pengambilan keputusan berdasarkan kondisi tertentu.

2. buatlah algoritma dan kode untuk Memeriksa apakah suatu persegi merupakan bujur sangkar atau bukan.

```
panjang_sisi = float(input("Masukkan panjang sisi persegi: "))  # Langkah 1
lebar_sisi = float(input("Masukkan lebar sisi persegi: "))      # Langkah 2

if panjang_sisi == lebar_sisi:  # Langkah 3
    print("Persegi merupakan bujur sangkar.")  # Langkah 5
else:  # Langkah 4
    print("Persegi bukan bujur sangkar.")      # Langkah 5
```

Program tersebut merupakan contoh implementasi dari sebuah program untuk memeriksa apakah suatu persegi merupakan bujur sangkar atau bukan. Berikut penjelasannya:
-Pengguna akan diminta untuk memasukkan panjang sisi persegi menggunakan input() dengan pesan yang sesuai.
-Nantinya pengguna juga diminta untuk memasukkan lebar sisi persegi menggunakan input() dengan pesan yang sesuai.

-Program memeriksa apakah panjang sisi dan lebar sisi persegi sama. Jika sama, maka persegi tersebut merupakan bujur sangkar.
Jika panjang sisi dan lebar sisi persegi tidak sama, maka persegi tersebut bukan bujur sangkar.
-Program mencetak hasil berdasarkan pengecekan yang dilakukan pada langkah-langkah sebelumnya.
Program ini akan memberikan hasil apakah persegi tersebut merupakan bujur sangkar atau bukan berdasarkan perbandingan antara panjang sisi dan lebar sisi.

3. buatlah algoritma dan kode untuk Menentukan apakah suatu bilangan lebih besar dari 75 atau tidak.

```
bilangan = float(input("Masukkan sebuah bilangan: "))  # Langkah 1

if bilangan > 75:  # Langkah 2
    print("Bilangan lebih besar dari 75.")  # Langkah 2
else:  # Langkah 3
    print("Bilangan tidak lebih besar dari 75.")  # Langkah 3
```

Program tersebut merupakan contoh implementasi dari sebuah program untuk memeriksa apakah suatu bilangan lebih besar dari 75 atau tidak.
Berikut penjelasannya:
-Pengguna akan diminta untuk memasukkan sebuah bilangan menggunakan input() dengan pesan yang sesuai.
-Kemudian program memeriksa apakah bilangan tersebut lebih besar dari 75.
-Jika bilangan tidak lebih besar dari 75, maka program mencetak pesan bahwa "Bilangan tidak lebih besar dari 75."
-Jika bilangan lebih besar dari 75, maka program mencetak pesan bahwa "Bilangan lebih besar dari 75."
Program ini akan memberikan hasil apakah bilangan yang dimasukkan lebih besar dari 75 atau tidak, berdasarkan perbandingan yang dilakukan pada langkah 2.

4. buatlah algoritma dan kode untuk Memeriksa apakah suatu bilangan bulat merupakan bilangan positif, bilangan negatif atau nol.

```
bilangan = int(input("Masukkan sebuah bilangan bulat: "))  # Langkah 1

if bilangan == 0:  # Langkah 2
    print("Bilangan adalah nol.")  # Langkah 2
elif bilangan > 0:  # Langkah 3
    print("Bilangan adalah positif.")  # Langkah 3
else:  # Langkah 4
    print("Bilangan adalah negatif.")  # Langkah 4
```

Program tersebut merupan contoh implementasi dari sebuah program untuk memeriksa apakah suatu bilangan bulat positif, negatif, atau nol. Berikut penjelasannya:

-Pengguna akan diminta untuk memasukkan sebuah bilangan bulat menggunakan input() dengan pesan yang sesuai.
-Kemudian program memeriksa apakah bilangan tersebut sama dengan 0.
-Jika bilangan sama dengan 0, maka program mencetak pesan bahwa "Bilangan adalah nol."
-Jika bilangan tidak sama dengan 0, program memeriksa apakah bilangan lebih dari 0.
-Jika bilangan lebih dari 0, maka program mencetak pesan bahwa "Bilangan adalah positif."
-Jika bilangan tidak sama dengan 0 dan tidak lebih dari 0 (artinya negatif), maka program mencetak pesan bahwa "Bilangan adalah negatif."
Dengan demikian, program akan memberikan informasi apakah bilangan yang dimasukkan adalah positif, negatif, atau nol, berdasarkan kondisi yang ada pada langkah 2, 4, dan 6.

5. buatlah algoritma dan kode untuk Menentukan tahun kabisat atau bukan.

```
tahun = int(input("Masukkan tahun: "))  # Langkah 1

if (tahun % 4 == 0 and tahun % 100 != 0) or (tahun % 400 == 0):  # Langkah 2
    print(f"{tahun} adalah tahun kabisat.")  # Langkah 4
else:  # Langkah 3
    print(f"{tahun} bukan tahun kabisat.")  # Langkah 4
```

Program tersebut merupakan implementasi dari program untuk menentukan apakah suatu tahun adalah tahun kabisat atau bukan. Berikut penjelasannya:
-Pengguna diminta untuk memasukkan tahun menggunakan input() dengan pesan yang sesuai.
-Program melakukan pengecekan kondisi untuk menentukan apakah tahun tersebut adalah tahun kabisat atau tidak.
-Jika kondisi pada langkah 2 tidak terpenuhi, program akan menjalankan blok kode di bagian else.
-Jika kondisi pada langkah 2 terpenuhi, program akan menjalankan blok kode di bagian if, dan akan mencetak bahwa tahun tersebut adalah tahun kabisat.
-Pada langkah 2, program melakukan dua pengecekan kondisi:
Pertama, (tahun % 4 == 0 and tahun % 100 != 0) untuk mengecek apakah tahun habis dibagi 4 dan tidak habis dibagi 100.
Kedua, (tahun % 400 == 0) untuk mengecek apakah tahun habis dibagi 400.
Jika salah satu atau kedua kondisi tersebut terpenuhi, maka tahun tersebut dianggap tahun kabisat.
-Pada langkah 4, program mencetak pesan sesuai dengan hasil pengecekan kondisi, baik itu tahun kabisat atau bukan tahun kabisat.

6. Menampilkan isi tabel berikut:

```
*

* #

* # *

* # * #

* # * # *

1

1 3

1 3 5

1 3 5 7
```

1 3 5 7 9

1 2 3 4 5

2 3 4 5

3 4 5

4 5

5

```
n = 5
for i in range(1, n+1):
    for j in range(1, i+1):
        print("*", end=" ")
        if j < i:
            print("#", end=" ")
    print()
n = 5
for i in range(1, n+1):
    for j in range(1, i*2, 2):
        print(j, end=" ")
    print()
n = 5
for i in range(n, 0, -1):
    for j in range(i, n+1):
        print(j, end=" ")
    print()
```

Program di atas adalah contoh implementasi dari tiga pola cetak menggunakan nested loop. Berikut penjelasan dari masing-masing bagian program:

-Pola pertama

Pola ini mencetak bentuk segitiga dengan kombinasi tanda '*' dan '#'. Setiap baris ke-i memiliki i bintang dan i-1 pagar.

-Pola kedua

Pola ini mencetak angka-angka berurut dari 1 hingga n, tetapi hanya angka ganjil. Setiap baris ke-i memiliki 2 * (i-1) + 1 angka.

-Pola ketiga

Pola ini mencetak angka-angka berurut dalam bentuk segitiga terbalik. Setiap baris ke-i mencetak angka dari i hingga n.
-Setiap pola menggunakan nested loop (loop bersarang) untuk mengatur pembentukan pola sesuai dengan jumlah baris yang diinginkan. Loop pertama mengatur baris-baris, dan loop kedua mengatur elemen-elemen dalam setiap baris. Kombinasi penggunaan print() dan parameter end=" " digunakan untuk mencetak elemen dalam satu baris. Pada setiap akhir baris, digunakan print() tanpa argumen agar baris baru dimulai pada baris berikutnya.

7. Menghitung total belanja, dengan ketentuan diskon berikut:

- Jika total belanja lebih dari Rp. 500.000,- maka akan mendapat diskon 10%

- Jika total belanja lebih dari Rp. 250.000,- maka akan mendapat diskon 5%

- Jika total belanja kurang dari Rp. 250.000,- tidak mendapat diskon

```
total_belanja = float(input("Masukkan total belanja: "))  # Langkah 1

if total_belanja > 500000:  # Langkah 2a
    diskon = 0.1 * total_belanja
elif total_belanja > 250000:  # Langkah 2b
    diskon = 0.05 * total_belanja
else:  # Langkah 2c
    diskon = 0

total_bayar = total_belanja - diskon  # Langkah 3

print(f"Total belanja: Rp. {total_belanja:,.2f}")
print(f"Diskon: Rp. {diskon:,.2f}")
print(f"Total bayar: Rp. {total_bayar:,.2f}")  # Langkah 4
```

Program tersebut adalah program yang menghitung total belanja dengan pemberian diskon, berdasarkan ketentuan yang diberikan. Berikut penjelasan dari masing-masing bagian program:
Langkah pertama:
Program akan meminta pengguna untuk memasukkan total belanja dalam bentuk angka desimal (float).
Langkah kedua:

Program menggunakan beberapa kondisi if-elif-else untuk menentukan jumlah diskon yang akan diberikan berdasarkan total belanja.
Jika total belanja lebih dari 500,000, maka diberikan diskon sebesar 10% dari total belanja.
Jika total belanja lebih dari 250,000, maka diberikan diskon sebesar 5% dari total belanja.
Jika total belanja tidak memenuhi kondisi di atas, tidak diberikan diskon (diskon = 0).
Langkah ketiga:
Program menghitung total bayar dengan mengurangkan diskon dari total belanja.
Langkah keempat:
Program mencetak informasi total belanja, diskon, dan total bayar dalam format yang rapi dengan menggunakan f-string (formatted string literals).
Nantinya program mencetak total belanja, jumlah diskon yang diberikan, dan total bayar setelah diskon. Angka-angka ditampilkan dengan format desimal dua angka dibelakang koma (gunakan :,.2f pada f-string).

8. Konversi nilai angka ke nilai huruf dengan ketentuan berikut:

Nilai Angka Nilai Huruf

85.0 < x ≤ 100.0 A

70.0 < x ≤ 85.0 B

55.0 < x ≤ 70.0 C

45.0 < x ≤ 55.0 D

0 < x ≤ 45.0 E

```
nilai_angka = float(input("Masukkan nilai angka: "))  # Langkah 1

if nilai_angka > 85.0 and nilai_angka <= 100.0:  # Langkah 2
    nilai_huruf = "A"
elif nilai_angka > 70.0 and nilai_angka <= 85.0:  # Langkah 3
    nilai_huruf = "B"
elif nilai_angka > 55.0 and nilai_angka <= 70.0:  # Langkah 4
    nilai_huruf = "C"
elif nilai_angka > 45.0 and nilai_angka <= 55.0:  # Langkah 5
```

```
    nilai_huruf = "D"
elif nilai_angka > 0 and nilai_angka <= 45.0:  # Langkah 6
    nilai_huruf = "E"

print(f"Nilai angka: {nilai_angka}")
print(f"Nilai huruf: {nilai_huruf}")  # Langkah 7
```

Program tersebut merupakan program yang mengkonversi nilai angka menjadi nilai huruf berdasarkan rentang tertentu, sesuai dengan ketentuan yang diberikan. Berikut adalah penjelasan dari langkah-langkah bagian program:

Langkah pertama:
Program meminta pengguna untuk memasukkan nilai angka dalam bentuk angka desimal (float).
Langkah kedua hingga keenam:
Program menggunakan kondisi if-elif-else untuk menentukan nilai huruf berdasarkan rentang nilai angka yang telah ditentukan.
Jika nilai angka lebih dari 85.0 dan kurang dari atau sama dengan 100.0, maka nilai huruf adalah "A".
Jika nilai angka lebih dari 70.0 dan kurang dari atau sama dengan 85.0, maka nilai huruf adalah "B".
Jika nilai angka lebih dari 55.0 dan kurang dari atau sama dengan 70.0, maka nilai huruf adalah "C".
Jika nilai angka lebih dari 45.0 dan kurang dari atau sama dengan 55.0, maka nilai huruf adalah "D".
Jika nilai angka lebih dari 0 dan kurang dari atau sama dengan 45.0, maka nilai huruf adalah "E".
Langkah ketujuh:
Program mencetak nilai angka yang dimasukkan oleh pengguna dan nilai huruf yang sesuai berdasarkan kondisi yang telah ditentukan sebelumnya.
Pada akhirnya, program akan mencetak informasi mengenai nilai angka dan nilai huruf yang sesuai dengan rentang yang telah ditentukan.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 10)

1. Jelaskan yang dilakukan dan yang ditampilkan program berikut!

```
print('Masukkan 5 Buah Bilangan')
print('------------------------')
jml = 0
bil = []
for i in range(5):
    print('Bilangan ke-%d :' % (i+1))
    x = int(input(' '))
    jml = jml + x
    bil.append(x)
rata2 = jml / 5
print('Data yang diinputkan : ')
print(bil)
print('\nRata-rata : %f \n' % rata2)
```

Penjelasan:

-print('Masukkan 5 Buah Bilangan') dan print('------------------------'): Ini adalah perintah cetak untuk menampilkan judul dan garis pembatas.

-jml=0: Inisialisasi variabel jml (jumlah) dengan nilai awal 0. Variabel ini akan digunakan untuk menghitung total dari bilangan-bilangan yang dimasukkan.

-bil=[]: Inisialisasi list kosong bil untuk menyimpan bilangan-bilangan yang dimasukkan oleh pengguna.

-for i in range(5):: Ini adalah loop for yang akan berjalan sebanyak 5 kali, mengulang proses penginputan bilangan.

-print('Bilangan ke-%d :' % (i+1)): Cetak pesan yang menunjukkan nomor urut bilangan yang diminta untuk dimasukkan.

-x = int(input(' ')): Minta pengguna memasukkan sebuah bilangan. Bilangan tersebut akan disimpan dalam variabel x sebagai integer.

-jml = jml + x: Tambahkan bilangan x ke nilai total jml.

-bil.append(x): Tambahkan bilangan x ke dalam list bil untuk menyimpan data yang diinputkan.

-rata2 = jml/5;: Hitung rata-rata dari bilangan-bilangan yang dimasukkan dengan membagi total jml dengan 5 (jumlah bilangan yang dimasukkan).
-print('Data yang diinputkan : ') dan print(bil): Cetak pesan yang menunjukkan data yang diinputkan, yaitu semua bilangan yang dimasukkan oleh pengguna.
-print('\nRata-rata : %f \n' % rata2): Cetak pesan yang menunjukkan rata-rata dari bilangan-bilangan yang dimasukkan.
Program ini meminta pengguna memasukkan 5 bilangan, menghitung total dan rata-rata, dan kemudian menampilkan data yang diinputkan beserta rata-ratanya.

2. modifikasi program tersebut menjadi untuk menghitung rata rata tinggi badan mahasiswa sebagai berikut:

mahasiswa absen 1, tinggi 185 cm

mahasiswa absen 2, tinggi 175 cm

mahasiswa absen 3, tinggi 171 cm

mahasiswa absen 4, tinggi 157 cm

mahasiswa absen 5, tinggi 169 cm

```
print('Menghitung Rata-rata Tinggi Badan Mahasiswa')
print('----------------------------------------')

# Data tinggi badan mahasiswa
tinggi_badan = [185, 175, 171, 157, 169]

# Menghitung total tinggi badan
total_tinggi = sum(tinggi_badan)

# Menghitung rata-rata tinggi badan
rata2_tinggi = total_tinggi / len(tinggi_badan)

print('Data tinggi badan mahasiswa:')
for i, tinggi in enumerate(tinggi_badan, start=1):
    print(f"Mahasiswa absen {i}, tinggi {tinggi} cm")

print('\nRata-rata tinggi badan mahasiswa: %.2f cm' % rata2_tinggi)
```

Program di atas adalah program yang menghitung rata-rata tinggi badan mahasiswa berdasarkan data yang telah diberikan. Berikut adalah penjelasan dari masing-masing bagian program:
-Langkah 1-2:
Program mencetak judul program untuk memberikan informasi tentang apa yang akan dihitung, yaitu rata-rata tinggi badan mahasiswa.
-Langkah 4:
Program menyediakan data tinggi badan mahasiswa dalam bentuk daftar (list) bernama tinggi_badan.
-Langkah 6:
Program menggunakan fungsi sum() untuk menghitung total dari semua elemen dalam daftar tinggi_badan.

-Langkah 8:
Program menghitung rata-rata tinggi badan dengan membagi total tinggi badan oleh jumlah data mahasiswa (panjang daftar tinggi_badan).
-Langkah 10-13:
Program mencetak data tinggi badan mahasiswa dengan menggunakan perulangan for dan fungsi enumerate(). Setiap elemen dalam daftar tinggi_badan di-cetak bersama dengan nomor absen mahasiswa.
-Langkah 15:
Program mencetak rata-rata tinggi badan mahasiswa dengan menggunakan tanda % untuk memformat output agar hanya memiliki 2 angka desimal.
Sedemikian sehingga, program akan mencetak informasi mengenai data tinggi badan mahasiswa beserta rata-rata tinggi badan dari seluruh mahasiswa.

3. Bagaimana cara mengkombinasikan tipe data list dengan dictionary, sehingga dapat menyimpan data mahasiswa yang berisi nama, NIM, dan nilai 5 mata kuliah. Kemudian tunjukkan cara memanggilnya.

```
# Meminta pengguna memasukkan jumlah mahasiswa
jumlah_mahasiswa = int(input("Masukkan jumlah mahasiswa: "))

# Inisialisasi list untuk menyimpan data mahasiswa
```

```
data_mahasiswa = []

# Meminta pengguna memasukkan data mahasiswa
for i in range(jumlah_mahasiswa):
    print(f"\nData Mahasiswa ke-{i+1}")
    nama = input("Nama: ")
    nim = input("NIM: ")
    nilai = []
    for j in range(5):
        nilai_mata_kuliah = float(input(f"Nilai mata kuliah ke-{j+1}: "))
        nilai.append(nilai_mata_kuliah)

    mahasiswa = {
        "nama": nama,
        "nim": nim,
        "nilai": nilai
    }
    data_mahasiswa.append(mahasiswa)

# Menampilkan data mahasiswa yang diinputkan
print("\nData Mahasiswa:")
for mahasiswa in data_mahasiswa:
    print(f"Nama: {mahasiswa['nama']}")
    print(f"NIM: {mahasiswa['nim']}")
    print(f"Nilai: {mahasiswa['nilai']}")
    print()
```

Dalam contoh di atas, data_mahasiswa adalah list yang berisi dictionary-dictionary. Setiap dictionary mewakili data dari seorang mahasiswa, termasuk nama, NIM, dan nilai mata kuliah dalam bentuk list. Kita dapat menambahkan lebih banyak data mahasiswa sesuai kebutuhan. Program kemudian menggunakan loop untuk memanggil dan mencetak data mahasiswa satu per satu, termasuk nama, NIM, dan nilai-nilai mata kuliah. Kita bisa mengubah, menambahkan, atau mengakses data sesuai dengan kebutuhan kita dengan cara yang sama.

4. Dalam data list, kita dapat melakukan pencarian dan pengurutan entri. Carilah informasi tentang algoritma pencarian: sequential search dan binary search.
   *Sequential Search (Pencarian Berurutan):
   Sequential search adalah algoritma pencarian yang paling sederhana dan langsung. Algoritma ini bekerja dengan mencari elemen satu per satu

dalam suatu list atau array sampai elemen yang dicari ditemukan atau seluruh list telah diperiksa. Jika elemen yang dicari ditemukan, maka pencarian dihentikan. Sequential search cocok digunakan pada list yang tidak terurut maupun terurut.
#Kelebihan:
- Mudah diimplementasikan.
- Cocok untuk list kecil atau ketika elemen yang dicari sering berada di bagian awal list.
#Kekurangan:
- Waktu eksekusi bisa lambat pada list besar.
- Dalam kasus terburuk, harus memeriksa semua elemen dalam list.

*Binary Search (Pencarian Biner):
Binary search adalah algoritma pencarian yang bekerja pada list terurut. Algoritma ini membandingkan elemen tengah dari list dengan elemen yang dicari dan mengeliminasi setengah dari sisa elemen yang perlu diperiksa berdasarkan hasil perbandingan. Jika elemen tengah tidak cocok, pencarian dilanjutkan pada setengah list yang relevan.
#Kelebihan:
- Sangat efisien pada list besar karena mengurangi jumlah perbandingan yang dibutuhkan.
- Pencarian dalam logaritma basis 2, sehingga cepat.
#Kekurangan:
- Hanya berfungsi pada list yang sudah terurut.
- Memerlukan list terurut sebagai prasyarat.

Untuk mengimplementasikan algoritma ini dalam Python, kita dapat menggunakan fungsi bawaan seperti `index()` untuk sequential search dan fungsi seperti `bisect_left()` atau `bisect_right()` dalam modul `bisect` untuk binary search.

5. Dalam data list, kita dapat melakukan pencarian dan pengurutan entri. Carilah informasi tentang algoritma pengurutan: Bubble sort, selection sort dan insertion sort.

*Bubble Sort:
Bubble sort adalah algoritma pengurutan sederhana yang membandingkan setiap pasangan elemen bersebelahan dalam list dan menukarkan posisinya jika urutannya salah. Algoritma ini terus melakukannya sampai seluruh list terurut. Bubble sort adalah salah satu algoritma pengurutan yang paling lambat, tetapi sangat mudah dipahami.
#Kelebihan:
- Sederhana dan mudah dimengerti.
- Cocok untuk list kecil atau hampir terurut.
#Kekurangan:
- Sangat lambat pada list besar.
- Memerlukan banyak perbandingan dan pertukaran.

*Selection Sort:
Selection sort adalah algoritma pengurutan yang memilih elemen terkecil dari list dan menukarkannya dengan elemen pertama. Kemudian, elemen terkecil dari sisa list dipilih dan ditukar dengan elemen kedua, dan seterusnya. Algoritma ini bekerja dengan mengurutkan list dari kiri ke kanan.
#Kelebihan:
- Sederhana dan mudah dipahami.
- Tidak memerlukan banyak pertukaran seperti bubble sort.
#Kekurangan:
- Tetap lambat pada list besar.
- Tidak efisien pada list yang sangat besar.

*Insertion Sort:
Insertion sort adalah algoritma pengurutan yang bekerja dengan cara memindahkan elemen satu per satu dari bagian tidak terurut ke bagian terurut dari list. Setiap elemen dimasukkan ke posisi yang benar di bagian terurut hingga seluruh list terurut.
#Kelebihan:
- Cocok untuk list kecil atau ketika sebagian besar elemen sudah terurut.
- Memiliki kinerja yang baik pada list yang hampir terurut.
#Kekurangan:

- Lambat pada list besar.
- Memerlukan banyak perpindahan elemen.

Untuk mengimplementasikan algoritma pengurutan ini dalam Python, kita dapat membuat fungsi khusus atau menggunakan fungsi bawaan seperti `sort()` atau `sorted()`.

6. Dalam data list, kita dapat melakukan pencarian dan pengurutan entri. Tuliskan pseudocode untuk salah satu algoritma pencarian dan pengurutan!
   Pseudocode adalah deskripsi yang tidak begitu formal dan mengikuti sintaksis yang kaku seperti dalam bahasa pemrograman tertentu. Tujuannya adalah untuk memberikan gambaran umum tentang alur logika suatu algoritma. Berikut adalah contoh pseudocode berupa deskripsi:

   *Pseudocode Algoritma Pencarian (Contoh: Binary Search):

```
BinarySearch(list, target):
   inisialisasi left dan right
   selama left kurang dari atau sama dengan right:
      hitung nilai tengah mid
      jika nilai tengah sama dengan target:
         kembalikan indeks mid
      jika nilai tengah kurang dari target:
         perbarui left
      jika nilai tengah lebih besar dari target:
         perbarui right
   kembalikan -1
```

   *Pseudocode Algoritma Pengurutan (Contoh: Insertion Sort):

```
InsertionSort(list):
   untuk i dari 1 hingga panjang list - 1:
      simpan nilai list[i] sebagai key
      simpan nilai i - 1 sebagai j
      selama j lebih besar dari atau sama dengan 0 dan list[j] lebih besar dari
key:
         pindahkan list[j] ke list[j + 1]
         kurangi j sebesar 1
      simpan key di list[j + 1]
```

7. Buatlah kode program menggunakan Python untuk algoritma berdasarkan pseudocode soal sebelumnya!
8. -Algoritma Pencarian: Binary Search

```python
def binary_search(list, target):
    left = 0
    right = len(list) - 1

    while left <= right:
        mid = (left + right) // 2
        if list[mid] == target:
            return mid
        elif list[mid] < target:
            left = mid + 1
        else:
            right = mid - 1

    return -1

# Contoh penggunaan
my_list = [2, 5, 7, 10, 18, 23, 30]
target = 5
result = binary_search(my_list, target)
if result != -1:
    print(f"Target ditemukan di indeks {result}")
else:
    print("Target tidak ditemukan")
```

Program di atas adalah implementasi algoritma pencarian biner (binary search) dalam Python. Algoritma ini digunakan untuk mencari suatu nilai tertentu dalam sebuah list yang telah diurutkan. Berikut adalah penjelasan dari masing-masing bagian program:

-Fungsi binary_search(list, target):

Ini adalah definisi dari fungsi pencarian biner. Fungsi ini menerima dua parameter: list (list yang akan dicari) dan target (nilai yang ingin dicari dalam list). Fungsi ini mengembalikan indeks dimana nilai target ditemukan dalam list, atau -1 jika nilai target tidak ditemukan.

-Langkah 2-9:

Program ini adalah implementasi dari algoritma pencarian biner dalam bentuk perulangan while. Variabel left dan right menandakan batas-batas

indeks antara elemen-elemen yang sedang dicari. Proses pencarian dilakukan selama left masih kurang dari atau sama dengan right. Di dalam perulangan, nilai tengah (mid) dihitung dan dibandingkan dengan nilai target. Jika nilai tengah sama dengan target, maka indeks tengah merupakan indeks dimana target ditemukan, dan fungsi mengembalikan nilai mid. Jika nilai tengah lebih kecil dari target, pencarian dilanjutkan di setengah kanan list dengan menggeser left ke mid + 1. Jika nilai tengah lebih besar dari target, pencarian dilanjutkan di setengah kiri list dengan menggeser right ke mid - 1.

-Langkah 11-16:

Ini adalah contoh penggunaan dari fungsi pencarian biner. List my_list berisi nilai-nilai yang sudah diurutkan. Nilai target yang ingin dicari adalah 5. Fungsi binary_search() dipanggil dengan list dan target sebagai argumen. Jika hasil pencarian bukan -1 (artinya nilai target ditemukan), maka program mencetak pesan yang menyebutkan indeks dimana target ditemukan. Jika hasil pencarian adalah -1 (artinya nilai target tidak ditemukan), maka program mencetak pesan "Target tidak ditemukan". Program tersebut menggambarkan bagaimana algoritma pencarian biner bekerja dan bagaimana mengimplementasikannya dalam Python untuk mencari nilai tertentu dalam list yang sudah diurutkan.

-Program Pengurutan: Insertion Sort

```
def insertion_sort(list):
    for i in range(1, len(list)):
        key = list[i]
        j = i - 1

        while j >= 0 and list[j] > key:
            list[j + 1] = list[j]
            j = j - 1

        list[j + 1] = key

# Contoh penggunaan
my_list = [12, 5, 8, 9, 3, 15, 7]
insertion_sort(my_list)
print("List yang sudah diurutkan:", my_list)
```

Program di atas adalah implementasi algoritma pengurutan penyisipan (insertion sort) dalam Python. Algoritma ini bekerja dengan cara membandingkan dan memindahkan elemen-elemen dalam list sehingga menghasilkan list yang terurut. Berikut adalah penjelasan dari masing-masing bagian program:

-Fungsi insertion_sort(list): Ini adalah definisi dari fungsi pengurutan penyisipan. Fungsi ini menerima satu parameter yaitu list (list yang akan diurutkan). Fungsi ini tidak mengembalikan nilai, tetapi mengurutkan list yang diberikan berdasarkan metode pengurutan penyisipan.

-Langkah 2-11:

Ini adalah implementasi dari algoritma pengurutan penyisipan. Perulangan dimulai dari indeks 1 hingga akhir list. Setiap elemen pada indeks i akan dianggap sebagai elemen yang akan disisipkan ke dalam bagian yang telah diurutkan. Variabel key menyimpan nilai dari elemen yang akan disisipkan. Variabel j mengindikasikan indeks dari elemen yang akan dibandingkan dengan key.

Selama j lebih besar atau sama dengan 0 dan nilai pada list[j] lebih besar dari key, elemen pada indeks j akan digeser ke posisi j + 1. Proses ini akan dilakukan secara berulang sampai j kurang dari 0 atau nilai pada list[j] tidak lagi lebih besar dari key. Setelah selesai, elemen key akan disisipkan ke dalam posisi yang benar dalam bagian yang telah diurutkan.

-Langkah 13-15:

Ini adalah contoh penggunaan dari fungsi pengurutan penyisipan. List my_list berisi nilai-nilai yang akan diurutkan menggunakan algoritma insertion sort. Fungsi insertion_sort() dipanggil dengan list sebagai argumen untuk mengurutkan list tersebut. Setelah selesai pengurutan, program mencetak list yang sudah diurutkan.

Program tersebut menggambarkan bagaimana algoritma pengurutan penyisipan bekerja dan bagaimana mengimplementasikannya dalam Python untuk mengurutkan elemen-elemen dalam list.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 11)

1. Buatlah program dengan menu sebagai berikut:

1. Input Data

2. Pencarian Data, dengan:

a. Pencarian Berurutan

b. Pencarian Biner

```
def sequential_search(arr, target):
    for i in range(len(arr)):
        if arr[i] == target:
            return i  # Mengembalikan indeks elemen yang ditemukan
    return -1  # Mengembalikan -1 jika tidak ditemukan

def binary_search(arr, target):
    low = 0
    high = len(arr) - 1
    while low <= high:
        mid = (low + high) // 2
        if arr[mid] == target:
            return mid  # Mengembalikan indeks elemen yang ditemukan
        elif arr[mid] < target:
            low = mid + 1
        else:
            high = mid - 1
    return -1  # Mengembalikan -1 jika tidak ditemukan

data = []

while True:
    print("\nMenu:")
    print("1. Input Data")
    print("2. Pencarian Data")
    print("3. Keluar")

    menu = input("Pilih menu: ")
```

```
if menu == '1':
    n = int(input("Masukkan jumlah data: "))
    data = []
    for i in range(n):
        nilai = int(input(f"Masukkan data ke-{i+1}: "))
        data.append(nilai)
    print("Data berhasil dimasukkan!")

elif menu == '2':
    if not data:
        print("Data belum dimasukkan.")
        continue
    target = int(input("Masukkan nilai yang ingin dicari: "))
    print("\nPencarian Berurutan:")
    result_seq = sequential_search(data, target)
    if result_seq != -1:
        print(f"Elemen {target} ditemukan di indeks {result_seq}")
    else:
        print("Elemen tidak ditemukan")

    print("\nPencarian Biner:")
    result_bin = binary_search(data, target)
    if result_bin != -1:
        print(f"Elemen {target} ditemukan di indeks {result_bin}")
    else:
        print("Elemen tidak ditemukan")

elif menu == '3':
    print("Terima kasih!")
    break

else:
    print("Menu tidak valid. Silakan pilih lagi.")
```

Program di atas adalah sebuah program yang memungkinkan pengguna untuk melakukan input data, melakukan pencarian data menggunakan metode pencarian berurutan dan pencarian biner, serta keluar dari program. Berikut adalah penjelasan dari masing-masing bagian program:
-Fungsi sequential_search(arr, target):
Ini adalah definisi dari fungsi pencarian berurutan. Fungsi ini menerima dua parameter, yaitu arr (array/list data) dan target (nilai yang dicari). Fungsi ini akan mengiterasi melalui setiap elemen dalam arr dan mencari elemen

yang sesuai dengan target. Jika ditemukan, fungsi akan mengembalikan indeks elemen yang ditemukan, jika tidak, akan mengembalikan -1.
-Fungsi binary_search(arr, target):
Ini adalah definisi dari fungsi pencarian biner. Fungsi ini menerima dua parameter, yaitu arr (array/list data) dan target (nilai yang dicari). Fungsi ini melakukan pencarian menggunakan algoritma pencarian biner. Jika ditemukan, fungsi akan mengembalikan indeks elemen yang ditemukan, jika tidak, akan mengembalikan -1.
-Langkah 31-49:
Ini adalah bagian utama dari program yang membentuk menu utama. Pengguna diberikan opsi untuk memilih menu, yaitu input data, pencarian data, atau keluar dari program.
-Langkah 34-40:
Jika menu yang dipilih adalah '1', pengguna diminta untuk memasukkan jumlah data yang akan diinput dan kemudian memasukkan data-data tersebut.
-Langkah 42-51:
Jika menu yang dipilih adalah '2', program akan melakukan pencarian data yang dimasukkan oleh pengguna. Jika data belum dimasukkan, program akan memberikan pesan bahwa data belum dimasukkan. Hasil pencarian menggunakan metode pencarian berurutan dan pencarian biner akan ditampilkan.
-Langkah 54-56:
Jika menu yang dipilih adalah '3', program akan memberikan pesan terima kasih dan keluar dari program.
-Langkah 58-61:
Jika menu yang dipilih tidak valid, program akan memberikan pesan bahwa menu tidak valid dan meminta pengguna untuk memilih lagi.
Program tersebut memberikan pengguna kemampuan untuk menginput data, melakukan pencarian berurutan dan pencarian biner pada data yang sudah dimasukkan, serta keluar dari program jika diperlukan.

Nama: Aji Andriansyah

Latihan Praktikum dasar-dasar pemrograman (Modul 12)

1. Apakah nilai balik dari suatu fungsi hanya satu nilai?
   Tidak, nilai balik dari suatu fungsi tidak terbatas pada hanya satu nilai. Dalam banyak bahasa pemrograman, termasuk Python, sebuah fungsi bisa mengembalikan satu nilai tunggal, atau bahkan nilai-nilai yang berupa tuple, daftar, objek, atau tipe data lainnya. Jadi, nilai balik dari suatu fungsi tidak terbatas pada satu nilai. Kita bisa mengembalikan tipe data apa pun sesuai kebutuhan program yang kita inginkan.

2. Tuliskan kode program berikut!

```
import math

def segitiga_siku_siku(a, t):
    L = 0.5 * a * t
    K = a + t + math.sqrt(a*a + t*t)
    return L, K

print('Program Segitiga Siku-siku')
print('--------------------------')

alas = float(input('Masukkan alas: '))
tinggi = float(input('Masukkan tinggi: '))

luas, keliling = segitiga_siku_siku(alas, tinggi)

print('--- Hasil ---')
print('Luas : ' + str(luas))
print('Keliling: ' + str(keliling))
```

   Program di atas adalah program yang menghitung luas dan keliling segitiga siku-siku berdasarkan nilai alas dan tinggi yang diinputkan oleh pengguna. Berikut adalah penjelasan dari masing-masing bagian program:
   -Baris 1:
   Baris pertama adalah pernyataan import math, yang mengimport modul math untuk digunakan dalam menghitung akar kuadrat.

-Definisi Fungsi segitiga_siku_siku(a, t):
Ini adalah definisi dari fungsi segitiga_siku_siku yang menerima dua parameter, yaitu a (alas segitiga) dan t (tinggi segitiga). Di dalam fungsi ini, dilakukan perhitungan luas dan keliling segitiga siku-siku berdasarkan rumus yang diberikan. Fungsi ini mengembalikan dua nilai, yaitu luas (L) dan keliling (K) segitiga.
-Baris 6-8:
Cetak judul program "Program Segitiga Siku-siku" untuk memberi tahu pengguna apa yang dilakukan oleh program.
-Baris 10-11:
Nantinya pengguna diminta untuk memasukkan nilai alas dan tinggi segitiga dalam bentuk angka desimal (float).
-Baris 13-15:
Panggil fungsi segitiga_siku_siku(alas, tinggi) dengan parameter nilai alas dan tinggi yang diinputkan oleh pengguna. Hasil luas dan keliling segitiga disimpan dalam variabel luas dan keliling.
-Baris 17-19:
Cetak hasil perhitungan luas dan keliling segitiga siku-siku dengan menggunakan variabel luas dan keliling.
Program tersebut memungkinkan pengguna untuk menghitung luas dan keliling segitiga siku-siku berdasarkan nilai alas dan tinggi yang dimasukkan, dan kemudian menampilkan hasil perhitungannya.

3. Tentukan nama subprogram yang terlibat!
   Nama subprogram yang terlibat dalam program tersebut adalah:
   - `segitiga_siku_siku`: Subprogram ini menghitung luas dan keliling segitiga siku-siku berdasarkan panjang alas dan tinggi yang diberikan.

4. Pisahkan parameter formal dan actual (jika ada), serta nilai baliknya (jika ada)!
   Parameter formal dan actual dalam program tersebut adalah:
   - Parameter formal dalam subprogram `segitiga_siku_siku`: `a` dan `t`
   - Parameter actual dalam pemanggilan subprogram `segitiga_siku_siku`: `alas` dan `tinggi`

Tidak ada nilai balik yang dijelaskan dalam subprogram `segitiga_siku_siku` pada kode yang Anda berikan.

5. Tuliskan pseudocodenya!
   Fungsi segitiga_siku_siku(a, t)
   L = 0.5 * a * t
   K = a + t + akar_kuadrat(a*a + t*t)
   Kembalikan L, K

   Tampilkan "Program Segitiga Siku-siku"
   Tampilkan "-------------------------"

   Input alas dari pengguna dan simpan dalam 'alas'
   Input tinggi dari pengguna dan simpan dalam 'tinggi'

   luas, keliling = panggil fungsi segitiga_siku_siku dengan parameter alas dan tinggi

   Tampilkan "--- Hasil ---"
   Tampilkan "Luas : " + ubah_ke_string(luas)
   Tampilkan "Keliling: " + ubah_ke_string(keliling)

6. Buatlah subprogram dalam bentuk pseudocode dan kode python untuk konversi nilai dari angka ke huruf, kemudian buat program utama yang memanggilnya, dengan ketentuan:
   Nilai Angka Nilai Huruf
   $85.0 < x \leq 100.0$ A
   $70.0 < x \leq 85.0$ B
   $55.0 < x \leq 70.0$ C
   $45.0 < x \leq 55.0$ D
   $0 \leq x \leq 45.0$ E
   Parameter : nilai angka
   Nilai balik : tidak ada

   Pseudocode

Subprogram konversi_nilai_ke_huruf(nilai_angka)
  Jika nilai_angka > 85 dan nilai_angka <= 100
    Kembalikan "A"
  Jika nilai_angka > 70 dan nilai_angka <= 85
    Kembalikan "B"
  Jika nilai_angka > 55 dan nilai_angka <= 70
    Kembalikan "C"
  Jika nilai_angka > 45 dan nilai_angka <= 55
    Kembalikan "D"
  Jika nilai_angka >= 0 dan nilai_angka <= 45
    Kembalikan "E"
  Jika tidak
    Kembalikan "Nilai tidak valid"

Tampilkan "Program Konversi Nilai Angka ke Huruf"
Input nilai_angka dari pengguna

hasil_konversi = panggil subprogram konversi_nilai_ke_huruf dengan parameter nilai_angka

Tampilkan "Nilai Huruf:", hasil_konversi

Program python

```python
def konversi_nilai_ke_huruf(nilai_angka):
    if 85 < nilai_angka <= 100:
        return "A"
    elif 70 < nilai_angka <= 85:
        return "B"
    elif 55 < nilai_angka <= 70:
        return "C"
    elif 45 < nilai_angka <= 55:
        return "D"
    elif 0 <= nilai_angka <= 45:
        return "E"
    else:
        return "Nilai tidak valid"

print("Program Konversi Nilai Angka ke Huruf")
```

```
nilai_angka = float(input("Masukkan nilai angka: "))
hasil_konversi = konversi_nilai_ke_huruf(nilai_angka)
print("Nilai Huruf:", hasil_konversi)
```

Program di atas adalah program yang mengkonversi nilai angka ke dalam nilai huruf berdasarkan rentang nilai tertentu. Berikut adalah penjelasan dari masing-masing bagian program:

-Definisi Fungsi konversi_nilai_ke_huruf(nilai_angka):

Ini adalah definisi dari fungsi konversi_nilai_ke_huruf yang menerima satu parameter, yaitu nilai_angka. Fungsi ini menggunakan sejumlah kondisi untuk menentukan nilai huruf berdasarkan rentang nilai angka yang diberikan. Jika nilai angka sesuai dengan salah satu rentang, maka fungsi akan mengembalikan nilai huruf yang sesuai. Jika nilai angka di luar rentang, maka akan dikembalikan "Nilai tidak valid".

-Baris 8:

Cetak judul program "Program Konversi Nilai Angka ke Huruf" untuk memberi tahu pengguna tentang tujuan program.

-Baris 9:

Pengguna diminta untuk memasukkan nilai angka dalam bentuk angka desimal (float).

-Baris 10-11:

Panggil fungsi konversi_nilai_ke_huruf(nilai_angka) dengan parameter nilai angka yang diinputkan oleh pengguna. Hasil konversi nilai angka ke huruf disimpan dalam variabel hasil_konversi.

-Baris 12:

Cetak hasil konversi nilai angka ke huruf dengan menggunakan variabel hasil_konversi.

Program tersebut memungkinkan pengguna untuk mengkonversi nilai angka menjadi nilai huruf berdasarkan rentang nilai yang ditentukan, dan kemudian menampilkan hasil konversinya.

7. Buatlah fungsi untuk menentukan bilangan terbesar dan terkecil dari 2 bilangan yang diinputkan, kemudian buat program utama yang memanggilnya, dengan ketentuan:
   Parameter : bilangan 1 dan bilangan 2.
   Nilai balik : bilangan terbesar dan terkecil.

```python
def cari_terbesar_terkecil(bilangan1, bilangan2):
    if bilangan1 > bilangan2:
        terbesar = bilangan1
        terkecil = bilangan2
    else:
        terbesar = bilangan2
        terkecil = bilangan1
    return terbesar, terkecil

print("Program Menentukan Bilangan Terbesar dan Terkecil")
bilangan1 = float(input("Masukkan bilangan pertama: "))
bilangan2 = float(input("Masukkan bilangan kedua: "))

bilangan_terbesar, bilangan_terkecil = cari_terbesar_terkecil(bilangan1,
bilangan2)

print("Bilangan terbesar:", bilangan_terbesar)
print("Bilangan terkecil:", bilangan_terkecil)
```

Program di atas adalah program yang menerima dua bilangan dan mengembalikan bilangan terbesar dan terkecil di antara keduanya. Berikut adalah penjelasan dari masing-masing bagian program:

-Definisi Fungsi cari_terbesar_terkecil(bilangan1, bilangan2):

Ini adalah definisi dari fungsi cari_terbesar_terkecil yang menerima dua parameter, yaitu bilangan1 dan bilangan2. Fungsi ini membandingkan kedua bilangan dan menentukan bilangan terbesar dan terkecil di antara keduanya. Fungsi mengembalikan dua nilai tersebut dalam urutan (terbesar, terkecil).

-Baris 7:

Cetak judul program "Program Menentukan Bilangan Terbesar dan Terkecil" untuk memberi tahu pengguna tentang tujuan program.

-Baris 8-9:

Pengguna diminta untuk memasukkan dua bilangan dalam bentuk angka desimal (float).

-Baris 10-11:

Panggil fungsi cari_terbesar_terkecil(bilangan1, bilangan2) dengan parameter kedua bilangan yang diinputkan oleh pengguna. Hasil dari fungsi ini disimpan dalam variabel bilangan_terbesar dan bilangan_terkecil.
-Baris 13-14:
Cetak hasil bilangan terbesar dan terkecil dengan menggunakan variabel bilangan_terbesar dan bilangan_terkecil.
Program tersebut memungkinkan pengguna untuk memasukkan dua bilangan dan kemudian menampilkan bilangan terbesar dan terkecil di antara keduanya.

5. Buatlah algoritma dan program tentang “Menghitung Luas dan Keliling Bidang Datar”dengan menu sebagai berikut:

1. Bujur Sangkar

2. Persegi Panjang

3. Segitiga

4. Lingkaran

Buatlah subprogram untuk setiap menu di atas yang mempunyai parameter (banyaknya parameter disesuaikan dengan kebutuhan) dan nilai balik (luas dan keliling).

Subprogram hitung_luas_keliling_bujur_sangkar(sisi)

luas = sisi * sisi

keliling = 4 * sisi

Kembalikan luas, keliling

Subprogram hitung_luas_keliling_persegi_panjang(panjang, lebar)

luas = panjang * lebar

keliling = 2 * (panjang + lebar)

Kembalikan luas, keliling

```
Subprogram hitung_luas_keliling_segitiga(alas, tinggi, sisi1, sisi2, sisi3)
    luas = 0.5 * alas * tinggi
    keliling = sisi1 + sisi2 + sisi3
    Kembalikan luas, keliling

Subprogram hitung_luas_keliling_lingkaran(jari_jari)
    luas = 3.14 * jari_jari * jari_jari
    keliling = 2 * 3.14 * jari_jari
    Kembalikan luas, keliling

Tampilkan "Program Menghitung Luas dan Keliling Bidang Datar"
Tampilkan "-----------------------------------------"
Tampilkan "1. Bujur Sangkar"
Tampilkan "2. Persegi Panjang"
Tampilkan "3. Segitiga"
Tampilkan "4. Lingkaran"
Pilihan = input("Pilih menu (1/2/3/4): ")
Jika Pilihan == "1"
    sisi = float(input("Masukkan panjang sisi: "))
    luas, keliling = panggil subprogram hitung_luas_keliling_bujur_sangkar dengan parameter sisi
    Tampilkan "Luas:", luas
    Tampilkan "Keliling:", keliling
Jika Pilihan == "2"
```

panjang = float(input("Masukkan panjang: "))

lebar = float(input("Masukkan lebar: "))

luas, keliling = panggil subprogram hitung_luas_keliling_persegi_panjang dengan parameter panjang dan lebar

Tampilkan "Luas:", luas

Tampilkan "Keliling:", keliling

Jika Pilihan == "3"

alas = float(input("Masukkan alas: "))

tinggi = float(input("Masukkan tinggi: "))

sisi1 = float(input("Masukkan panjang sisi 1: "))

sisi2 = float(input("Masukkan panjang sisi 2: "))

sisi3 = float(input("Masukkan panjang sisi 3: "))

luas, keliling = panggil subprogram hitung_luas_keliling_segitiga dengan parameter alas, tinggi, sisi1, sisi2, sisi3

Tampilkan "Luas:", luas

Tampilkan "Keliling:", keliling

Jika Pilihan == "4"

jari_jari = float(input("Masukkan jari-jari: "))

luas, keliling = panggil subprogram hitung_luas_keliling_lingkaran dengan parameter jari_jari

Tampilkan "Luas:", luas

Tampilkan "Keliling:", keliling

Jika tidak

Tampilkan "Pilihan tidak valid"

Kode Program

```
def hitung_luas_keliling_bujur_sangkar(sisi):
    luas = sisi * sisi
    keliling = 4 * sisi
    return luas, keliling

def hitung_luas_keliling_persegi_panjang(panjang, lebar):
    luas = panjang * lebar
    keliling = 2 * (panjang + lebar)
    return luas, keliling

def hitung_luas_keliling_segitiga(alas, tinggi, sisi1, sisi2, sisi3):
    luas = 0.5 * alas * tinggi
    keliling = sisi1 + sisi2 + sisi3
    return luas, keliling

def hitung_luas_keliling_lingkaran(jari_jari):
    luas = 3.14 * jari_jari * jari_jari
    keliling = 2 * 3.14 * jari_jari
    return luas, keliling

print("Program Menghitung Luas dan Keliling Bidang Datar")
print("-----------------------------------------")

print("1. Bujur Sangkar")
print("2. Persegi Panjang")
print("3. Segitiga")
print("4. Lingkaran")

Pilihan = input("Pilih menu (1/2/3/4): ")

if Pilihan == "1":
    sisi = float(input("Masukkan panjang sisi: "))
    luas, keliling = hitung_luas_keliling_bujur_sangkar(sisi)
    print("Luas:", luas)
    print("Keliling:", keliling)
elif Pilihan == "2":
    panjang = float(input("Masukkan panjang: "))
    lebar = float(input("Masukkan lebar: "))
    luas, keliling = hitung_luas_keliling_persegi_panjang(panjang, lebar)
    print("Luas:", luas)
    print("Keliling:", keliling)
elif Pilihan == "3":
    alas = float(input("Masukkan alas: "))
```

```
    tinggi = float(input("Masukkan tinggi: "))
    sisi1 = float(input("Masukkan panjang sisi 1: "))
    sisi2 = float(input("Masukkan panjang sisi 2: "))
    sisi3 = float(input("Masukkan panjang sisi 3: "))
    luas, keliling = hitung_luas_keliling_segitiga(alas, tinggi, sisi1,
sisi2, sisi3)
    print("Luas:", luas)
    print("Keliling:", keliling)
elif Pilihan == "4":
    jari_jari = float(input("Masukkan jari-jari: "))
    luas, keliling = hitung_luas_keliling_lingkaran(jari_jari)
    print("Luas:", luas)
    print("Keliling:", keliling)
else:
    print("Pilihan tidak valid")
```

Program di atas adalah program yang memungkinkan pengguna untuk menghitung luas dan keliling berbagai bentuk bidang datar seperti bujur sangkar, persegi panjang, segitiga, dan lingkaran. Berikut adalah penjelasan dari masing-masing bagian program:

-Definisi Fungsi hitung_luas_keliling_bujur_sangkar(sisi):
Fungsi ini menerima satu parameter sisi yang merupakan panjang sisi dari bujur sangkar. Fungsi menghitung luas dan keliling bujur sangkar berdasarkan panjang sisi yang diberikan dan mengembalikan dua nilai tersebut dalam urutan (luas, keliling).

-Definisi Fungsi hitung_luas_keliling_persegi_panjang(panjang, lebar):
Fungsi ini menerima dua parameter, yaitu panjang dan lebar, yang merupakan panjang dan lebar persegi panjang. Fungsi menghitung luas dan keliling persegi panjang berdasarkan panjang dan lebar yang diberikan dan mengembalikan dua nilai tersebut dalam urutan (luas, keliling).

-Definisi Fungsi hitung_luas_keliling_segitiga(alas, tinggi, sisi1, sisi2, sisi3):
Fungsi ini menerima lima parameter, yaitu alas, tinggi, sisi1, sisi2, dan sisi3, yang merupakan informasi tentang segitiga. Fungsi menghitung luas dan keliling segitiga berdasarkan informasi yang diberikan dan mengembalikan dua nilai tersebut dalam urutan (luas, keliling).

-Definisi Fungsi hitung_luas_keliling_lingkaran(jari_jari):

Fungsi ini menerima satu parameter jari_jari yang merupakan jari-jari lingkaran. Fungsi menghitung luas dan keliling lingkaran berdasarkan jari-jari yang diberikan dan mengembalikan dua nilai tersebut dalam urutan (luas, keliling).
-Baris 22-25:
Program mencetak judul "Program Menghitung Luas dan Keliling Bidang Datar" dan daftar opsi menu yang tersedia, yaitu bujur sangkar, persegi panjang, segitiga, dan lingkaran.
-Baris 27-29:
Program meminta pengguna untuk memasukkan pilihan menu.
-Baris 31-67:
Struktur percabangan if-elif-else digunakan untuk menghandle pilihan menu yang dimasukkan pengguna. Bergantung pada pilihan menu yang dipilih, program akan meminta input sesuai dengan bentuk bidang datar yang dipilih dan kemudian menghitung serta mencetak luas dan kelilingnya menggunakan fungsi yang telah didefinisikan sebelumnya.
-Baris 69-70:
Jika pengguna memasukkan pilihan menu yang tidak valid, program mencetak pesan "Pilihan tidak valid".
Program ini memungkinkan pengguna untuk menghitung luas dan keliling dari berbagai bentuk bidang datar berdasarkan pilihan yang kita buat.

6. Buatlah algoritma dan program tentang pengolahan statistika dasar, meliputi:

1. Mencari nilai rata-rata
2. Mencari nilai standar deviasi data
3. Mencari nilai minimum
4. Mencari nilai maksimum

Buatlah subprogram untuk setiap poin di atas dengan parameter dan nilai balik sesuai kebutuhan.

Algoritma
Subprogram hitung_rata_rata(data)

```
    jumlah = 0
    untuk setiap nilai dalam data:
        jumlah = jumlah + nilai
    rata_rata = jumlah / panjang(data)
    Kembalikan rata_rata

Subprogram hitung_standar_deviasi(data)
    rata_rata = panggil subprogram hitung_rata_rata dengan parameter data
    jumlah = 0
    untuk setiap nilai dalam data:
        jumlah = jumlah + (nilai - rata_rata) ** 2
    standar_deviasi = akar kuadrat(jumlah / (panjang(data) - 1))
    Kembalikan standar_deviasi

Subprogram hitung_minimum(data)
    minimum = data[0]
    untuk setiap nilai dalam data:
        jika nilai < minimum:
            minimum = nilai
    Kembalikan minimum

Subprogram hitung_maksimum(data)
    maksimum = data[0]
    untuk setiap nilai dalam data:
        jika nilai > maksimum:
            maksimum = nilai
    Kembalikan maksimum

Tampilkan "Program Pengolahan Statistika Dasar"
Tampilkan "-----------------------------------"
data = []
jumlah_data = int(input("Masukkan jumlah data: "))
untuk i dari 1 hingga jumlah_data:
    nilai = float(input("Masukkan nilai ke-" + i + ": "))
    tambahkan nilai ke data
```

Tampilkan "1. Mencari nilai rata-rata"
Tampilkan "2. Mencari nilai standar deviasi data"
Tampilkan "3. Mencari nilai minimum"
Tampilkan "4. Mencari nilai maksimum"

Pilihan = input("Pilih menu (1/2/3/4): ")

Jika Pilihan == "1"
  rata_rata = panggil subprogram hitung_rata_rata dengan parameter data
  Tampilkan "Rata-rata:", rata_rata
Jika Pilihan == "2"
  standar_deviasi = panggil subprogram hitung_standar_deviasi dengan parameter data
  Tampilkan "Standar Deviasi:", standar_deviasi
Jika Pilihan == "3"
  minimum = panggil subprogram hitung_minimum dengan parameter data
  Tampilkan "Nilai Minimum:", minimum
Jika Pilihan == "4"
  maksimum = panggil subprogram hitung_maksimum dengan parameter data
  Tampilkan "Nilai Maksimum:", maksimum
Jika tidak
  Tampilkan "Pilihan tidak valid"
Kode program

```
import math

def hitung_rata_rata(data):
    jumlah = sum(data)
    rata_rata = jumlah / len(data)
    return rata_rata

def hitung_standar_deviasi(data):
```

```
    rata_rata = hitung_rata_rata(data)
    jumlah = sum([(nilai - rata_rata) ** 2 for nilai in data])
    standar_deviasi = math.sqrt(jumlah / (len(data) - 1))
    return standar_deviasi

def hitung_minimum(data):
    minimum = min(data)
    return minimum

def hitung_maksimum(data):
    maksimum = max(data)
    return maksimum

print("Program Pengolahan Statistika Dasar")
print("-----------------------------------")

data = []
jumlah_data = int(input("Masukkan jumlah data: "))
for i in range(1, jumlah_data + 1):
    nilai = float(input("Masukkan nilai ke-" + str(i) + ": "))
    data.append(nilai)

print("1. Mencari nilai rata-rata")
print("2. Mencari nilai standar deviasi data")
print("3. Mencari nilai minimum")
print("4. Mencari nilai maksimum")

Pilihan = input("Pilih menu (1/2/3/4): ")

if Pilihan == "1":
    rata_rata = hitung_rata_rata(data)
    print("Rata-rata:", rata_rata)
elif Pilihan == "2":
    standar_deviasi = hitung_standar_deviasi(data)
    print("Standar Deviasi:", standar_deviasi)
elif Pilihan == "3":
    minimum = hitung_minimum(data)
    print("Nilai Minimum:", minimum)
elif Pilihan == "4":
    maksimum = hitung_maksimum(data)
    print("Nilai Maksimum:", maksimum)
else:
    print("Pilihan tidak valid")
```

Program di atas adalah program untuk melakukan pengolahan statistika dasar terhadap suatu set data. Program ini memungkinkan pengguna untuk menghitung nilai rata-rata, standar deviasi, nilai minimum, dan nilai maksimum dari sekumpulan data yang dimasukkan. Berikut adalah penjelasan dari masing-masing bagian program:

-Definisi Fungsi hitung_rata_rata(data):

Fungsi ini menerima satu parameter data, yaitu sebuah list yang berisi data numerik. Fungsi menghitung nilai rata-rata dari data yang diberikan dan mengembalikan nilai rata-rata tersebut.

-Definisi Fungsi hitung_standar_deviasi(data):

Fungsi ini juga menerima satu parameter data. Fungsi ini pertama-tama memanggil fungsi hitung_rata_rata(data) untuk menghitung nilai rata-rata. Kemudian, fungsi ini menghitung nilai standar deviasi dari data berdasarkan rumus statistika yang umum. Fungsi mengembalikan nilai standar deviasi tersebut.

-Definisi Fungsi hitung_minimum(data):

Fungsi ini menerima satu parameter data. Fungsi ini menggunakan fungsi min(data) bawaan Python untuk mencari nilai terkecil dalam data. Fungsi mengembalikan nilai terkecil tersebut.

-Definisi Fungsi hitung_maksimum(data):

Fungsi ini menerima satu parameter data. Fungsi ini menggunakan fungsi max(data) bawaan Python untuk mencari nilai terbesar dalam data. Fungsi mengembalikan nilai terbesar tersebut.

-Baris 24-27:

Program meminta pengguna untuk memasukkan jumlah data yang akan diolah. Selanjutnya, program melakukan looping untuk meminta pengguna memasukkan nilai-nilai data.

-Baris 29-32:

Program mencetak pilihan menu yang tersedia: mencari nilai rata-rata, standar deviasi data, nilai minimum, dan nilai maksimum.

-Baris 34-37:

Program meminta pengguna memilih menu.

-Baris 39-56:

Struktur percabangan if-elif-else digunakan untuk menghandle pilihan menu yang dimasukkan pengguna. Bergantung pada pilihan menu yang dipilih, program akan memanggil fungsi yang sesuai dengan data yang dimasukkan pengguna dan mencetak hasil perhitungan sesuai dengan pilihan.
-Baris 58-59:
Jika pengguna memasukkan pilihan menu yang tidak valid, program mencetak pesan "Pilihan tidak valid".
Program ini memberikan kemampuan kepada pengguna untuk melakukan pengolahan statistika dasar terhadap suatu set data numerik dengan mudah dan cepat.

REFERENSI JAWABAN

-Buku dasar-dasar pemrograman v2021.

-Modul praktikum dasar-dasar pemrograman.