

Nama : Muhammad Irham Taufik Harahap

NPM : 2514375012

Matkul : Program Dasar

Prodi : Sistem Informasi

Bagian A: Soal Teori

1. Algoritma dan Flowchart

Jelaskan hubungan antara algoritma, pseudocode, dan flowchart. Mengapa ketiganya penting sebelum menulis program komputer?

Jawab:

- Hubungan Ketiganya

Ketiganya berkaitan sebagai tahapan dalam merancang program:

- **Algoritma** → gagasan dasar berupa langkah-langkah logis.
- **Pseudocode** → representasi algoritma dalam bentuk tulisan yang mudah diterjemahkan ke kode.
- **Flowchart** → representasi algoritma dalam bentuk gambar/diagram alir.

Jadi, **pseudocode dan flowchart adalah dua cara berbeda untuk menggambarkan algoritma**, satu dalam bentuk teks, satu dalam bentuk gambar.

Tujuannya sama: membuat logika program jelas sebelum ditulis dalam bahasa pemrograman.

- Mengapa Ketiganya Penting Sebelum Menulis Program?

1. **Memperjelas pemahaman masalah**
Programmer memahami apa yang harus dilakukan sebelum menulis kode.
2. **Mengurangi kesalahan logika**
Kesalahan dapat diketahui lebih awal saat masih dalam tahap desain.
3. **Mempermudah proses coding**
Pseudocode menjadi "draft" yang dapat langsung diterjemahkan ke bahasa pemrograman.
4. **Meningkatkan efisiensi kerja**
Perencanaan yang baik menghindari penulisan ulang kode.
5. **Memudahkan komunikasi**
Flowchart membantu menjelaskan alur kepada tim, guru, atau rekan kerja.

2 . Variabel dan Tipe Data

Apa perbedaan antara variabel dan konstanta?. Berikan contoh penggunaannya dalam program Python atau C

Jawab :

1. Variabel

Variabel adalah tempat untuk menyimpan data **yang nilainya dapat berubah** selama program berjalan.

Ciri-ciri variabel:

- Nilainya fleksibel (bisa diubah kapan saja).
- Digunakan untuk menyimpan input, hasil perhitungan, atau data sementara.

Contoh variabel dalam Python:

```
x = 10
x = x + 5    # nilai x berubah menjadi 15
print(x)
```

2. Konstanta

Konstanta adalah tempat untuk menyimpan data **yang nilainya tidak boleh berubah** selama program berjalan.

Ciri-ciri konstanta:

- Nilainya tetap.
- Biasanya digunakan untuk nilai-nilai penting yang tidak ingin diubah, misalnya π (pi), batas maksimum, atau kecepatan cahaya.

Catatan (Python):

Python **tidak memiliki fitur konstanta secara bawaan**, tetapi programmer biasanya menuliskannya dalam **huruf kapital** sebagai bentuk kesepakatan.

Contoh konstanta dalam Python:

```
PI = 3.14159
RADIUS = 7

area = PI * (RADIUS ** 2)
print(area)
```

Pada contoh di atas, **PI** dan **RADIUS** dianggap sebagai konstanta karena **ditulis dalam huruf kapital** dan tidak diubah selama program berjalan.

3. Operator Logika dan Relasional

Jelaskan fungsi operator, AND, OR , dan NOT dalam pemrograman. Berikan contoh penerapannya dalam kondisi percabangan.

Jawab :

1. Operator AND

Operator **AND** menghasilkan **True** jika *kedua* kondisi bernilai **True**. Jika salah satu atau kedua kondisi **False**, hasilnya **False**.

Contoh logika:

- True AND True → True
- True AND False → False

Contoh dalam percabangan:

```
umur = 20
punya_ktp = True

if umur >= 17 and punya_ktp:
    print("Boleh memasuki area dewasa.")
else:
    print("Tidak boleh masuk.")
```

Penjelasan: Seseorang hanya boleh masuk jika **umur ≥ 17 dan punya KTP**.

2. Operator OR

Operator **OR** menghasilkan **True** jika **salah satu** kondisi bernilai True. Hanya akan menjadi **False** jika *kedua* kondisi False.

Contoh logika:

- True OR False → True
- False OR False → False

Contoh dalam percabangan:

```
hari = "Sabtu"

if hari == "Sabtu" or hari == "Minggu":
    print("Hari ini libur.")
```

```
else:
    print("Hari kerja.")
```

Penjelasan: Jika hari adalah Sabtu **atau** Minggu, maka dianggap libur.

3. Operator NOT

Operator **NOT** digunakan untuk **membalik** nilai logika:

- NOT True → False
- NOT False → True

Contoh dalam percabangan:

```
login = False

if not login:
    print("Silakan login terlebih dahulu.")
else:
    print("Selamat datang!")
```

Penjelasan: Jika kondisi `login` bernilai **False**, operator NOT menjadikannya **True**.

4. Struktur Percabangan

Sebuah toko memberikan potongan harga 10% jika pembelian di atas Rp100.000. Tuliskan algoritma atau pseudocode untuk menentukan total bayar setelah diskon

Jawab :

Algoritma

1. Mulai
 2. Input total pembelian
 3. Jika total pembelian > 100.000
 - Hitung diskon = $10\% \times \text{total pembelian}$
 - Total bayar = total pembelian – diskonJika tidak
 - Total bayar = total pembelian (tanpa diskon)
 4. Tampilkan total bayar
 5. Selesai
-

Pseudocode

```
BEGIN
    INPUT total_pembelian

    IF total_pembelian > 100000 THEN
        diskon ← total_pembelian * 0.10
        total_bayar ← total_pembelian - diskon
    ELSE
        total_bayar ← total_pembelian
    ENDIF

    PRINT total_bayar
END
```

5. Struktur Perulangan

Jelaskan perbedaan utama antara perulangan for and while, serta berikan contoh penggunaan yang tepat untuk masing masing.

Jawab :

Perbedaan Utama: For vs While

1. Perulangan `for`

- Digunakan ketika **jumlah perulangan sudah diketahui** di awal.
- Biasanya digunakan untuk **mengakses elemen pada daftar**, atau **mengulang dalam rentang angka tertentu**.
- Struktur perulangan lebih ringkas dan mudah dibaca.

Cocok untuk:

- ✓ Mengulang sebanyak N kali
 - ✓ Mengulang setiap elemen dalam list, string, atau range
-

2. Perulangan `while`

- Digunakan ketika **jumlah perulangan tidak ditentukan di awal**, tetapi tergantung kondisi tertentu.
- Perulangan akan terus berjalan **selama kondisi bernilai True**.
- Harus hati-hati agar tidak terjadi *infinite loop* (loop tak berakhir).

Cocok untuk:

- ✓ Mengulang sampai kondisi tercapai

- ✓ Menunggu input yang benar
 - ✓ Proses yang tidak memiliki batasan jumlah iterasi
-

Contoh Penggunaan `for` dalam Python

Contoh 1: Mengulang 5 kali

```
for i in range(5):  
    print("Perulangan ke-", i)
```

Contoh 2: Mengakses elemen list

```
buah = ["apel", "jeruk", "mangga"]
```

```
for item in buah:  
    print(item)
```

Contoh Penggunaan `while` dalam Python

Contoh 1: Mengulang sampai kondisi tercapai

```
x = 1  
while x <= 5:  
    print("Nilai x:", x)  
    x += 1
```

Contoh 2: Validasi input

```
password = ""  
  
while password != "1234":  
    password = input("Masukkan password: ")  
  
print("Login berhasil!")
```
