

Nama : Heriandi Limbong
NPM : 2514375029
Prodi : Sistem Informasi
Matkul : Pemrograman Dasar
Kelas : I FSD REG 2A MALAM

Bagian A : Soal Teori

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

jawaban:

Hubungan antara algoritma,pseudocode dan flowchart yaitu algoritma berupa ide atau serangkaian instruksi sistematis untuk memecahkan suatu masalah.pseudocode dan flowchart adalah dua cara berbeda untuk menulis atau memvisualisasikan algoritma yang dimana pseudocode menjelaskan logika program menggunakan bahasa sederhana yang mudah dibaca manusia,tanpa terikat aturan sintaksis bahasa pemrograman,sementara flowchart menggambarkan alur algoritma secara visual menggunakan simbol-simbol seperti oval (awal/akhir), persegi Panjang (proses), jajar genjang(input/output), belah ketupat (percabangan logika), panah (arah alur proses)

Algoritma,pseudocode,dan flowchart sangat penting sebelum menulis program pada komputer karena membantu merencanakan logika, mengidentifikasi kesalahan lebih awal,mempermudah komunikasi tim,serta membuat program lebih terstruktur dan merupakan fondasi untuk memecahkan masalah secara

sistematis sebelum beralih ke sintaks bahasa pemrograman dan menghasilkan kode yang lebih efektif dan efisien.

2. Apa perbedaan antara variabel dan konstanta ? Berikan contoh penggunaannya dalam program python/ C

Jawaban:

perbedaan variabel dan konstanta adalah **variabel** memiliki nilai yang dapat berubah selama eksekusi program dan digunakan untuk menyimpan data yang nilainya akan diproses atau diubah, sedangkan **konstanta** memiliki nilai yang tetap dan tidak dapat diubah setelah didefinisikan dan digunakan untuk nilai-nilai yang bersifat tetap, seperti satuan gravitasi (g) yang tidak berubah sepanjang program dijalankan

contoh dalam program python:

varibel (nilai bisa berubah)

umur = 22

umur = 25 #nilai berubah

konstanta

PI = 3.14159

GRAVITASI = 9.8

3. Jelaskan fungsi operator and, or, dan not dalam pemrograman. Berikan contoh penerapannya dalam kondisi percabangan.

Jawaban:

fungsi AND : untuk mengevaluasi dua kondisi(operand) dan menghasilkan nilai true hanya jika kedua kondisi bernilai true.

fungsi OR : untuk mengevaluasi dua kondisi dan menghasilkan true jika jika salah satu dari kondisi bernilai true (atau keduanya bernilai true).

fungsi NOT : untuk mengembalikan nilai boolean dari suatu kondisi, Jika suatu kondisi bernilai true, NOT akan mengubahnya menjadi false, jika bernilai false, NOT akan mengubahnya menjadi true

contoh penerapannya dalam kondisi percabangan:

```
status_anggota = "premium"
```

```
total_belanja = 500000
```

```
hari_libur = false
```

```
#Logika : ( anggota premium atau belanja besar) DAN bukan hari libur
```

```
if (status_anggota == "premium" or total_belanja > 400000) and not
```

```
    hari_libur:
```

```
        print("anda mendapatkan diskon 10%!")
```

```
else:
```

```
    print("tidak ada diskon yang berlaku")
```

4. sebuah toko memberikan potongan harga 10% jika pembelian di atas Rp.100.000. Tuliskan algoritma atau pseudocode untuk menentukan total bayar setelah diskon.

Jawaban:

Algoritma:

1. mulai proses perhitungan
2. terima jumlah total belanja (misalnya, dari kasir atau input pengguna)
3. periksa apakah total belanja lebih dari Rp.100.000
4. jika iya (kondisi terpenuhi) :
 1. hitung jumlah diskon (10% dari total belanja)
 2. hitung total bayar dengan mengurangi diskon dari total belanja.
5. jika tidak (kondisi tidak terpenuhi) :
 1. total bayar sama dengan total belanja (tidak ada diskon)
6. tampilkan atau berikan informasi mengenai jumlah total yang harus dibayarkan.
7. selesai

5. jelaskan perbedaan utama antara perulangan for dan while, serta berikan contoh penggunaan yang tepat untuk masing-masing.

Jawaban:

perulangan **for** yaitu digunakan Ketika jumlah iterasi sudah diketahui atau dapat dihitung di awal perulangan, struktur for biasanya mengkonsolidasikan inisialisasi variabel perhitungan, kondisi penghentian, dan langkah kenaikan/penurunan dalam satu baris, sedangkan perulangan **while** yaitu digunakan Ketika jumlah

iterasi tidak diketahui diawal dan perulangan harus berlanjut selama suatu kondisi boolean (benar/salah) tetap terpenuhi.

contoh perulangan FOR :

python:

```
print("mulai menghitung: ")
```

```
# fungsi range (1,7) menghasilkan angka 1,2,3,4,5,6
```

```
for angka in range (1, 7):
```

```
    print("angka saat ini:" , angka
```

```
print("selesai menghitung")
```

output:

angka saat ini: 1

angka saat ini: 2

angka saat ini: 3

angka saat ini: 4

angka saat ini: 5

angka saat ini: 6

selesai menghitung

contoh perulangan while :

python:

```
angka = 0
```

```
while angka % 2 != 0 :
```

```
    input_str = input("masukkan angka genap: ")
```

```
try :  
    angka = int( input_str)  
  
except ValueError:  
    print("input tidak valid.harap masukkan angka: ")  
  
print(f"bagus,anda memasukkan angka genap: {angka}")
```

BAGIAN B : SOAL PRAKTIKUM

1. SISTEM NILAI AKADEMIK

kode python:

```
Coding Python
Auto saved at 24:01:00
RUN MENU

1 #sistem nilai akademik
2
3 nama = (input("masukkan nama: "))
4 nilai = float(input("masukkan nilai: "))
5
6 if nilai >= 85:
7     grade = "mendapatkan nilai A"
8 elif nilai >=80:
9     grade = "mendapatkan nilai B"
10 elif nilai >=75:
11     grade = "mendapatkan nilai C"
12 elif nilai >=60:
13     grade = "mendapatkan nilai D"
14 elif nilai <=44:
15     grade = "mendapatkan nilai E"
16 else:
17     print("tidak lulus")
18 print(f"{nama} {grade}")
19
20
21
22
23
24
25
26
27
28
```

hasil run:

```
Compile Result

masukkan nama: diki
masukkan nilai: 88
diki mendapatkan nilai A

[Process completed - press Enter]
```

2.DAFTAR BELANJA HARIAN

kode python:

```
Coding Python
Auto saved at 22:54:34
RUN MENU

1 print("==== Daftar belanja harian =====")
2
3 jumlah_barang = int(input("Masukkan jumlah barang "))
4
5 total_belanja = 0
6
7 for i in range(1, jumlah_barang + 1):
8     print(f"\nBarang ke-{i}")
9     nama_barang = input("Nama barang: ")
10    harga = float(input("Harga per item (Rp): "))
11    jumlah_item = int(input("Jumlah item: "))
12    subtotal = harga * jumlah_item
13    total_belanja += subtotal
14    print(f"Subtotal {nama_barang}: Rp{subtotal:,.0f}")
15
16 print("\n-----")
17 print(f"Total belanja sebelum diskon: Rp{total_belanja:,.0f}")
18
19 if total_belanja > 100000:
20     diskon = 0.1 * total_belanja
21     print("Anda mendapat diskon 10%!")
22 else:
23     diskon = 0
24     print("Belanja kurang dari Rp100.000, tidak mendapat diskon.")
25
26 total_akhir = total_belanja - diskon
27
28 print("\n===== STRUK PEMBAYARAN =====")
29 print(f"Total Belanja : Rp{total_belanja:,.0f}")
30 print(f"Total Diskon : Rp{diskon:,.0f}")
31 print(f"Total Bayar : Rp{total_akhir:,.0f}")
32 print("=====")
33
```

hasil run:

```
Compile Result

==== Daftar belanja harian ====
Masukkan jumlah barang: 2

Barang ke-1
Nama barang: beras
Harga per item (Rp): 145000
Jumlah item: 3
Subtotal beras: Rp435,000

Barang ke-2
Nama barang: minyak
Harga per item (Rp): 35000
Jumlah item: 2
Subtotal minyak: Rp70,000

-----
Total belanja sebelum diskon: Rp505,000
Anda mendapat diskon 10%!
Total Belanja : Rp505,000
Total Diskon : Rp50,500
Total Bayar : Rp454,500
```

3.SISTEM ANTRIAN KLINIK

kode python:

```
Coding Python
Auto saved at 23:42:13
RUN MENU

1 # Aplikasi antrian klinik
2
3 antrian = []
4 print("==== sistem antrian klinik====")
5
6 jumlah_pasien = int(input("masukkan jumlah pasien: "))
7
8 for i in range(jumlah_pasien):
9     nama = input(f"pasien{i+1}(nama): ")
10    keluhan = input(f"masalah kesehatan {nama}: ")
11    antrian.append(nama)
12
13 print("\n daftar antrian saat ini: ")
14 print(antrian)
15
16 while len(antrian) > 0:
17     selesai = input("\n masukkan nama pasien yang telah selesai: ")
18
19     ditemukan = "false"
20     for pasien in antrian:
21         if pasien.startswith(selesai):
22             antrian.remove(pasien)
23             ditemukan = True
24             break
25
26 if not ditemukan:
27     print("!pasien tidak ditemukan dalam daftar antrian!")
28 else:
29     print("daftar antrian saat ini: ",antrian)
30
31 print("\nsemua pasien telah dilayani.")
32
33
34
35
```

hasil run :

```
Compile Result

==== sistem antrian klinik====
masukkan jumlah pasien: 3
pasien1(nama): diki
masalah kesehatan diki: batuk
pasien2(nama): aryo
masalah kesehatan aryo: demam
pasien3(nama): rizal
masalah kesehatan rizal: sakit kepala

daftar antrian saat ini:
['diki', 'aryo', 'rizal']

masukkan nama pasien yang telah selesai:
diki
daftar antrian saat ini: ['aryo', 'rizal']

semua pasien telah dilayani.
[Process completed - press Enter]
```