



DAFTAR ISI

BAB III : RUNTUNAN

A. Pengertian

B. Contoh-contoh

1. Algoritma Luas Segitiga
2. Algoritma Komisi Salesman
3. Algoritma Gaji Karyawan
4. Algoritma Konversi Waktu ke Total Detik
5. Algoritma Konversi Total Detik ke Waktu
6. Algoritma Pembulatan Seratusan
7. Algoritma Pertukaran Data
8. Algoritma Menghitung Selisih Waktu

C. Latihan



BAB III : RUNTUNAN

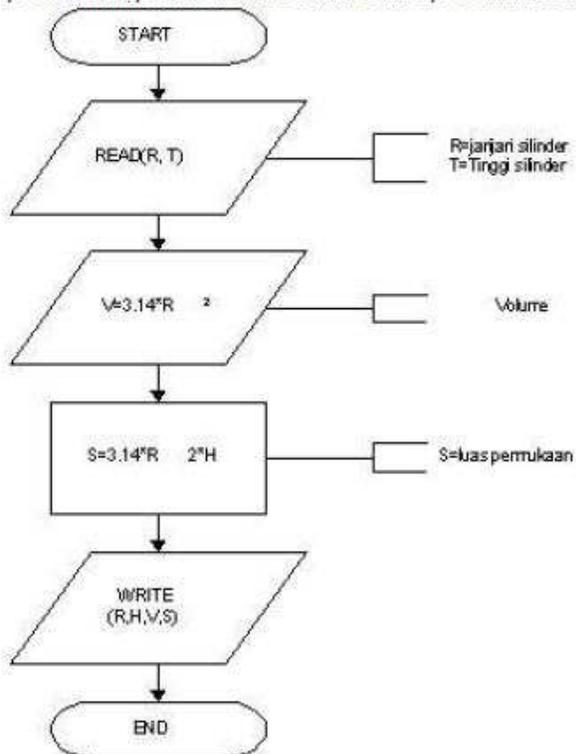
A. Pengertian

Runtunan merupakan struktur algoritma yang mempunyai ciri-ciri sebagai berikut:

1. tiap instruksi dikerjakan satu per satu
2. tiap instruksi dikerjakan tepat hanya sekali, dengan kata lain tidak ada instruksi yang diulang
3. instruksi dikerjakan berurutan mulai dari instruksi pertama hingga instruksi terakhir
4. akhir dari instruksi terakhir merupakan akhir algoritma.

-Sequential (berurutan)

perhitungan volume dan luas permukaan silinder



Dalam algoritma, urutan instruksi adalah penting. Hal ini dapat dilihat dalam contoh berikut ini.

10 Algoritma Contoh_Runtunan_1

20 Deklarasi:

30 a, b : integer

40 Deskripsi

50 a =10



```
60 a =2*a
70 b =a
80 write(b) {mencetak nilai 20}
```

```
10 Algoritma Contoh_Runtunan_2
20 Deklarasi:
30 a, b : integer
40 Deskripsi
50 a ←10
60 b ←a
70 a ←*a
80 write(b) {mencetak nilai 10}
```

Dalam contoh di atas, instruksi ke-2 dan ke-3 dipertukarkan, ternyata menghasilkan output atau keluaran yang berbeda. Hal ini menunjukkan bahwa urutan instruksi merupakan hal yang sangat penting karena sangat mempengaruhi hasil output.

B. Contoh-contoh

1. Algoritma Luas Segitiga

Buatlah algoritma untuk menghitung luas segitiga jika diketahui panjang alas dan tinggi segitiga.

Analisis:

input :

alas (a) dan tinggi (t) segitiga

output : luas (L)

segitiga rumus : $L = 0,5 \times a \times t$ Langkah pengerjaan:

1. meminta input data alas dan tinggi segitiga dari user
2. menghitung luas segitiga menggunakan rumus
3. mencetak output berupa luas segitiga

```
10 Algoritma Menghitung_Luas_Segitiga
```

```
20 {Menghitung luas segitiga jika diketahui panjang alas dan tinggi segitiga. Alas dan tinggi diinput dari user. Kemudian Luas dihitung menggunakan rumus Luas = 0,5 x alas x tinggi. Kemudian mencetak output berupa luas segitiga}
```

```
30 Deklarasi:
```

```
40 a, t, L : real
```

```
50 Deskripsi
```

```
60 read(a,t)
```



```
70 L ← 0.5 * a * t
80 write(L)
```

2. Algoritma Komisi Salesman

Buatlah algoritma untuk menghitung komisi yang diterima salesman berdasarkan jumlah penjualan yang dicapainya. Salesman tersebut mendapat komisi 10% dari hasil penjualannya. Input algoritma ini adalah nama salesman dan jumlah penjualan yang dicapainya.

Sedangkan outputnya adalah nama salesman dan besar komisi yang diperolehnya.

Analisis:

input: nama salesman (nama) dan jumlah penjualan (j)

output: nama salesman (nama) dan besar komisi (komisi) rumus: komisi = 10% x jumlah penjualan

Langkah pengerjaan:

1. menginput data nama salesman dan jumlah penjualan
2. menghitung komisi menggunakan rumus
3. mencetak nama dan komisi

```
10 Algoritma Menghitung_Komisi_Salesman
```

```
20 {Menghitung besarnya komisi yang diperoleh salesman berdasarkan jumlah penjualan. Besarnya komisi adalah 10% dari jumlah penjualan yang dicapainya. Kemudian algoritma akan mencetak nama salesman dan komisi sebagai outputnya.}
```

```
30 Deklarasi:
```

```
40 nama : string {nama salesman}
```

```
50 j : integer {jumlah penjualan}
```

```
60 komisi : real {komisi yang diperoleh}
```

```
70 Deskripsi
```

```
80 read(nama, j)
```

```
90 komisi ← 0.1 * j
```

```
100 write(nama, komisi)
```

3. Algoritma Gaji Karyawan

Buatlah algoritma untuk menghitung gaji karyawan. Diberikan nama karyawan dan besarnya gaji pokok. Gaji bersih yang diterima pegawai adalah gaji pokok ditambah besarnya tunjangan kemudian dikurangi pajak. Tunjangan karyawan dihitung 20% dari gaji pokok, sedangkan pajak adalah 15% dari gaji pokok ditambah tunjangan. Keluaran yang diharapkan adalah nama karyawan, besarnya tunjangan, pajak, dan gaji bersihnya.

Analisis:

input: nama karyawan (nama) dan besarnya gaji pokok (gaji_pokok)



output: nama karyawan (nama), tunjangan (tunj), pajak (pjk), dan gaji bersih (gaji_bersih)
rumus: $tunj = 20\% \times gaji_pokok$ $pjk = 15\% \times (gaji_pokok + tunj)$ $gaji_bersih = gaji_pokok + tunj - pajak$.

Langkah pengerjaan:

1. menginput nama karyawan dan gaji pokok
2. menghitung tunjangan
3. menghitung pajak
4. menghitung gaji bersih
5. mencetak nama, tunjangan, pajak, dan gaji bersih

10 Algoritma Menghitung_Gaji_Karyawan

20 {Diberikan nama karyawan dan gaji pokok. Kemudian menghitung tunjangan, pajak, dan gaji bersih berdasarkan gaji pokok. Tunjangan adalah 20% dari gaji pokok. Pajak adalah 15% dari gaji pokok ditambah tunjangan. Gaji bersih adalah gaji pokok ditambah tunjangan dikurangi pajak. Keluaran yang diharapkan adalah nama karyawan, tunjangan, pajak, dan gaji bersih.}

30 Deklarasi:

40 nama : string

50 gaji_pokok : integer

60 tunj, pjk, gaji_bersih : real

70 Deskripsi

80 read(nama, gaji_pokok)

90 $tunj \leftarrow 0.2 * gaji_pokok$

100 $pjk \leftarrow 0.15 * (gaji_pokok + tunj)$

110 $gaji_bersih \leftarrow gaji_pokok + tunj - pjk$

120 write(nama, tunj, pjk, gaji_bersih)

4. Algoritma Konversi Waktu ke Total Detik

Buatlah algoritma untuk mengkonversi jam-menit-detik ke total detik. Data jam-menit-detik diinput dari user.

Contoh, misalnya data jam-menit-detiknya adalah 1 jam 30 menit 40 detik, maka besarnya total detik adalah 5440 detik.

Analisis:

input: jam (j), menit (m), detik (d)

output: total detik (total)

Rumus: ingat bahwa 1 jam = 3600 detik dan 1 menit = 60 detik. maka total detik = jam x 3600 + menit x 60 + detik

Langkah pengerjaan:

1. menginput jam, menit, dan detik
2. menghitung total detik menggunakan rumus



3. mencetak total detik

10 Algoritma Konversi_Waktu_Ke_Detik

20 {mengkonversi jam-menit-detik ke total detik. Data yang diinput adalah jam, menit, dan detik. Outputnya adalah total detiknya.}

30 Deklarasi:

40 j, m, d, total : integer

50 Deskripsi

60 read(j,m,d)

70 total $\leftarrow j*3600 + m*60 + d$

80 write(total)

5. Algoritma Konversi Total Detik ke Waktu

Buatlah algoritma untuk mengkonversi total detik ke bentuk jam-menit-detik. Data total detik diinput oleh user.

Contohnya, misalnya data total detiknya adalah 5440 detik, maka outputnya adalah 1 jam 30 menit 40 detik.

Analisis: input: total detik (total) output: jam (j), menit (m), detik (d)

rumus: Pada dasarnya, yang hendak dikerjakan adalah mencari tahu total detik tersebut sama dengan berapa jam dan berapa sisa detiknya.

Dari sisa detik tersebut, barulah dicari berapa menit dan sisa berapa detik.

Contohnya:

misalkan total detiknya adalah 5440 detik.

Maka 5440 detik dibagi 3600 adalah 1 jam sisa 1840 detik.

Kemudian, 1840 detik dibagi 60 adalah 30 menit sisa 40 detik.

Jadi 5440 detik = 1 jam 30 menit 40 detik.

Dari contoh di atas dapat disimpulkan rumus-rumus

berikut: jam = total div 3600 sisa = total mod 3600 menit = sisa div 60 detik = sisa mod 60

Langkah pengerjaan:

1. menginput total detik
2. menghitung jam dari total detik
3. menghitung sisa pembagian jam dari total detik
4. menghitung menit dari sisa
5. menghitung detik dari sisa
6. mencetak jam, menit, dan detik

10 Algoritma Konversi_Total_Detik_ke_Waktu



```
20 {Diberikan suatu total detik. Hendak dicari berapa jam, menit, dan detiknya.}
30 Deklarasi:
40 j,m,d, sisa, total : integer
50 Deskripsi
60 read(total)
70 j ←total div 3600
80 sisa ←total mod 3600
90 m←sisa div 60
100 d ←sisa mod 60
110 write(j,m,d)
```

6. Algoritma Pembulatan Seratusan

Buatlah algoritma untuk membulatkan suatu data harga ke dalam ratusan.

Data harus dibulatkan ke atas.

Misalnya harga suatu barang adalah 8374 rupiah.

Maka harga dibulatkan menjadi 8400 rupiah.

Analisis: input: data harga (harga)

output: hasil pembulatan (hasil) Caranya:

1. Ambil dua angka terakhirnya dengan cara di mod dengan 100.
Dalam kasus di atas, $8374 \text{ mod } 100 = 74$.
2. Kemudian hitunglah 100 dikurangi 2 angka terakhir tersebut.

Dalam kasus di atas, $100 - 74 = 26$.

Hasil pembulatannya adalah harga ditambah hasil langkah ke-2.

Dalam kasus di atas, hasil pembulatan dari 8374 adalah $8374 + 26 = 8400$.

10 Algoritma Pembulatan_Ratusan

```
20 {menginput suatu harga barang. Kemudian harga barang tersebut akan dibulatkan dalam ratusan ke atas.}
```

```
30 Deklarasi:
```

```
40 harga, hasil : integer
```

```
50 Deskripsi
```

```
60 read(harga)
```

```
70 hasil ←100 - harga mod 100 + harga
```

```
80 write(hasil)
```

Algoritma di atas mempunyai kelemahan, yaitu akan menghasilkan output yang salah apabila nilai dari harga dalam kelipatan 100.

Terdapat cara lain dalam melakukan pembulatan ratusan.



Hal ini dituliskan dalam algoritma berikut ini.

10 Algoritma Pembulatan_Ratusan_2

20 {menginput suatu harga barang. Kemudian harga barang tersebut akan dibulatkan dalam ratusan ke atas.}

30 Deklarasi:

40 Harga, hasil : integer

50 Deskripsi

60 Read(harga)

70 Hasil \leftarrow (harga + 99) div 100 * 100

80 Write(hasil)

7. Algoritma Pertukaran Data

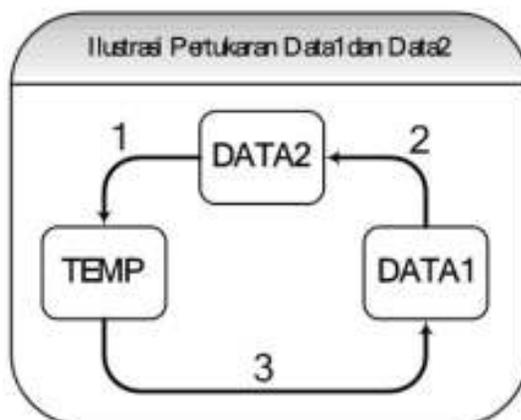
Buatlah algoritma yang saling menukarkan isi dari 2 variabel. Tipe variabel dianggap bertipe real.

Analisis:

Langkah pertama tentu saja adalah meminta input dua data real ke variabel tersebut. Misalkan nama variabelnya adalah data1 dan data2.

Data1 tidak boleh langsung dipindahkan ke data2, demikian juga sebaliknya. Misalnya langsung diberi pernyataan data2 &data1, maka informasi yang terdapat di data2 akan hilang.

Oleh karena itu, **sebelumnya, informasi dari data2 harus dipindahkan terlebih dahulu ke suatu variabel sementara**, misalnya diberi nama **temp**. Selanjutnya informasi di data1 dapat dipindah ke variabel data2. Kemudian informasi data2 yang tersimpan di variabel temp dapat dipindahkan ke variabel data1. Untuk lebih jelasnya, perhatikan ilustrasi berikut:



10 Algoritma Pertukaran_Data



```
20 {menukar isi dari 2 variabel bertipe real}
30 Deklarasi:
40 data1, data2, temp : real
50 Deskripsi
60 read(data1,data2)
70 temp ←data2
80 data2 ←data1
90 data1 ←temp
100 write(data1,data2)
```

8. Algoritma Menghitung Selisih Waktu

Diberikan dua buah data waktu dalam bentuk jam-menit-detik. Buatlah algoritma untuk menghitung selisihnya dan ditampilkan dalam bentuk jam-menit-detik. Asumsi waktu pertama adalah waktu awal, dan waktu kedua adalah waktu akhir. Selisih adalah waktu kedua dikurangi waktu pertama.
analisis: input: waktu1: jam (j1), menit (m1), detik (d1) waktu2: jam (j2), menit (m2), detik (d2)
output: waktu3: jam (j3), menit (m3), detik (d3)

Cara pengerjaan:

1. ubah waktu1 menjadi total detik (total1)
2. ubah waktu2 menjadi total detik (total2)
3. cari selisih total detik (total3), yaitu $total3 = total2 - total1$
4. ubah total3 menjadi dalam bentuk jam (j3), menit (m3), dan detik (d3)
5. cetak j3, m3, dan d3

```
10 Algoritma Menghitung_Selisih_waktu
20 {Diberikan waktu awal dan waktu akhir dalam bentuk jam-menit-detik. Kemudian mencari selisih waktunya dalam bentuk jam-menit-detik}
30 Deklarasi:
40 j1, m1, d1,
50 j2, m2, d2,
60 j3, m3, d3,
70 total1, total2, total3, sisa : integer
80 Deskripsi
90 read(j1,m1,d1) {menginput waktu awal}
100 read(j2,m2,d2) {menginput waktu akhir}
110
120 {mengubah waktu awal ke total detik}
130 total1 ←3600*j1 + 60*m1 + d1
140
150 {mengubah waktu akhir ke total detik}
```



```
160 total2 ←3600*j2 + 60*m2 + d2
170 {mencari selisih total waktu}
180 total3 ←total2 – total1
190 {mengubah total3 menjadi bentuk jam-menit-detik}
200 j3 ←total3 div 3600
210 sisa ←total3 mod 3600
220 m3 ←sisa div 60
230 d3 ←sisa mod 60
240 {mencetak hasilnya}
250 write(j3,m3,d3)
```