

STRUKTUR PERCABANGAN

OLEH : ANDRI HERYANDI, M.T.



06

STRUKTUR UTAMA ALGORITMA

01153 - Algoritma dan Struktur Data 1

- Struktur utama badan algoritma terdiri dari 3 jenis, yaitu :
 - Struktur Runtunan/Sequence
 - **Struktur Percabangan/Pemilihan/Branching/Kondisi**
 - Struktur Pengulangan/Loop/Iterasi



Oleh : Andri Heryandi, M.T.

STRUKTUR PERCABANGAN

01153 - Algoritma dan Struktur Data 1

Bagian #2



Oleh : Andri Heryandi, M.T.

PERCABANGAN DENGAN BANYAK KONDISI

01153 - Algoritma dan Struktur Data 1

- Terkadang dalam sebuah percabangan membutuhkan lebih dari satu kondisi.
- Contoh : Seorang anak akan mendapatkan hadiah dengan syarat :
 - Mendapatkan Rangking 1
 - Mendapatkan nilai rata-rata lebih besar dari 8.
- Dari contoh di atas, ada 2 syarat untuk mendapatkan hadiah. Itu menandakan sebuah percabangan dengan 2 buah kondisi.
- Karena menggunakan lebih dari 1 kondisi, maka dibutuhkan operator logika yang berperan sebagai kata sambung. Operator tersebut adalah **and** dan **or**.

A	B	A and B	A or B
True	True	True	True
True	False	False	True
False	True	False	True
False	False	False	False

Catatan :

- Operator **and** akan menghasilkan kesimpulan True jika kedua (semua) kondisi harus bernilai True.
- Operator **or** akan menghasilkan kesimpulan True jika salah satu kondisi bernilai True



Oleh : Andri Heryandi, M.T.

PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 1:

Menggunakan operator AND

Kamus

```
ranking:integer  
ratarata:real
```

Algoritma

```
input(ranking)  
input(ratarata)  
if(ranking=1)and(ratarata>8) then  
    output("Anak Berhak Dapat Hadiah")  
else  
    output("Anak Tidak Berhak Hadian")  
endif
```

Menggunakan operator OR

Kamus

```
ranking:integer  
ratarata:real
```

Algoritma

```
input(ranking)  
input(ratarata)  
if(ranking=1)or(ratarata>8) then  
    output("Anak Berhak Dapat Hadiah")  
else  
    output("Anak Tidak Berhak Hadian")  
endif
```



PERCABANGAN DENGAN BANYAK KONDISI (PYTHON)

01153 - Algoritma dan Struktur Data 1

■ Contoh 1:

Menggunakan operator **and**

```
ranking = int(input("Ranking : "))
ratarata = float(input("Rata-Rata : "))
if (ranking==1) and (ratarata>8):
    print("Anak berhak dapat hadiah")
else:
    print("Anak tidak berhak dapat hadiah")
```

Hasil Eksekusi

```
Ranking : 1 [true]
Rata-Rata : 9.5 [true]
Anak berhak dapat hadiah [true]
```

```
Ranking : 1 [true]
Rata-Rata : 7.5 [false]
Anak tidak berhak dapat hadiah [false]
```

```
Ranking : 5 [false]
Rata-Rata : 8.5 [true]
Anak tidak berhak dapat hadiah [false]
```

```
Ranking : 2 [false]
Rata-Rata : 7.5 [false]
Anak tidak berhak dapat hadiah [false]
```



PERCABANGAN DENGAN BANYAK KONDISI (PYTHON)

01153 - Algoritma dan Struktur Data 1

■ Contoh 1:

Menggunakan operator **or**

```
ranking = int(input("Ranking : "))
ratarata = float(input("Rata-Rata : "))
if (ranking==1) or (ratarata>8):
    print("Anak berhak dapat hadiah")
else:
    print("Anak tidak berhak dapat hadiah")
```

Hasil Eksekusi

```
Ranking : 1 [true]
Rata-Rata : 9.5 [true]
Anak berhak dapat hadiah [true]
```

```
Ranking : 1 [true]
Rata-Rata : 7.5 [false]
Anak berhak dapat hadiah [true]
```

```
Ranking : 5 [false]
Rata-Rata : 8.5 [true]
Anak berhak dapat hadiah [true]
```

```
Ranking : 2 [false]
Rata-Rata : 7.5 [false]
Anak tidak berhak dapat hadiah [false]
```



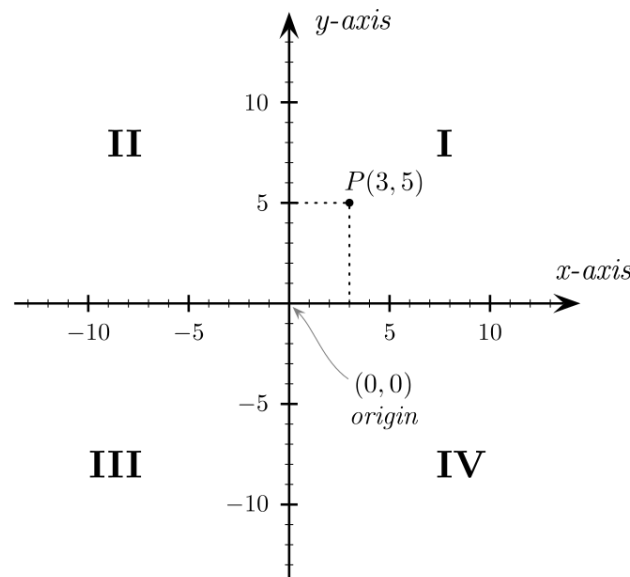
PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 2 :

- Buatlah sebuah aplikasi yang dapat menentukan lokasi kuadran suatu koordinat.
- Input : koordinat x dan y
- Output : Menampilkan posisi dari koordinat tersebut (di kuadran I, II, III atau IV, di titik pusat, di sumbu x atau di sumbu y)
- Ketentuan Kuadran

Kuadran	nilai x	nilai y
I	> 0	> 0
II	< 0	> 0
III	< 0	< 0
IV	> 0	< 0



Hasil Eksekusi

X : 2

Y : 5

Koordinat (2,5) berada di kuadran I

X : -2

Y : 5

Koordinat (-2,5) berada di kuadran II

X : -2.5

Y : -7

Koordinat (-2.5,-7) berada di kuadran III

X : 0

Y : 0

Koordinat (0,0) berada di titik pusat

X : 0

Y : -5

Koordinat (0,-5) berada di sumbu x

PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 2:

Kamus

x,y:real

Algoritma

input(x)

input(y)

if (x>0) and (y>0) then

 output("Kuadran I")

else if (x<0) and (y>0) then

 output("Kuadran II")

else if (x<0) and (y<0) then

 output("Kuadran III")

else if (x>0) and (y<0) then

 output("Kuadran IV")

else if (x=0) and (y=0) then

 output("Titik Pusat")

else if (x=0) and (y<>0) then

 output("Sumbu X")

else if (x<>0) and (y=0) then

 output("Sumbu Y")

end if



PERCABANGAN DENGAN BANYAK KONDISI (PYTHON)

01153 - Algoritma dan Struktur Data 1

■ Contoh 2:

```
x = float(input("X : "))
y = float(input("Y : "))
print(f"Koordinat ({x},{y}) berada di ",end="")
if (x>0) and (y>0):
    print("Kuadran I")
elif (x<0) and (y>0):
    print("Kuadran II")
elif (x<0) and (y<0):
    print("Kuadran III")
elif (x>0) and (y<0):
    print("Kuadran IV")
elif (x==0) and (y==0):
    print("Titik Pusat")
elif (x==0) and (y!=0):
    print("Sumbu X")
elif (x!=0) and (y==0):
    print("Sumbu Y")
```



PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 3 :

- Buatlah sebuah aplikasi yang dapat menghitung nilai akhir, indeks mutu dan keterangan nilai seorang mahasiswa
- Input : nilaitugas, nilaiuts, nilaiuas
- Proses
 - Hitung Nilai Akhir = $30 \% * \text{tugas} + 30\% * \text{uts} + 40\% * \text{uas}$
 - Penentuan Indeks mutu dan keterangan dengan ketentuan :

Nilai	Indeks Mutu	Keterangan
$100 \geq \text{Nilai} \geq 80$	A	Sangat Baik
$80 > \text{Nilai} \geq 68$	B	Baik
$68 > \text{Nilai} \geq 56$	C	Cukup
$56 > \text{Nilai} \geq 45$	D	Kurang
$45 > \text{Nilai} \geq 0$	E	Sangat Kurang

- Output : Nilai Akhir dan indeks dan keterangan

Hasil Eksekusi

Nilai Tugas : 70
Nilai UTS : 80
Nilai UAS : 85
Nilai Akhir : 79.00
Indeks Mutu : B
Keterangan : Baik

Nilai Tugas : 20
Nilai UTS : 30
Nilai UAS : 60
Nilai Akhir : 39.00
Indeks Mutu : E
Keterangan : Sangat Kurang



PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 3:

```
tugas = float(input("Nilai Tugas : "))
uts   = float(input("Nilai UTS   : "))
uas   = float(input("Nilai UAS   : "))
nilaiakhir = 0.30*tugas+0.30*uts+0.40*uas
print(f"Nilai Akhir : {nilaiakhir:6.2f}")
if (nilaiakhir<=100)and(nilaiakhir>=80):
    indeksmutu="A"
    keterangan="Sangat Baik"
elif (nilaiakhir<80)and(nilaiakhir>=68):
    indeksmutu="B"
    keterangan="Baik"
elif (nilaiakhir<68)and(nilaiakhir>=56):
    indeksmutu="C"
    keterangan="Cukup"
```

```
elif (nilaiakhir<56)and(nilaiakhir>=45):
    indeksmutu="D"
    keterangan="Kurang"
elif (nilaiakhir<45)and(nilaiakhir>=0):
    indeksmutu="E"
    keterangan="Sangat Kurang"
else:
    indeksmutu="X"
    keterangan="Nilai tidak sah"
print(f"Indeks Mutu : {indeksmutu}")
print(f"Keterangan  : {keterangan}")
```



PERCABANGAN DENGAN BANYAK KONDISI (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 3:

```
tugas = float(input("Nilai Tugas : "))
uts   = float(input("Nilai UTS   : "))
uas   = float(input("Nilai UAS   : "))
nilaiakhir = 0.30*tugas+0.30*uts+0.40*uas
print(f"Nilai Akhir : {nilaiakhir:6.2f}")
if 100 > nilaiakhir >= 80:
    indeksmutu="A"
    keterangan="Sangat Baik"
elif 80 > nilaiakhir >= 68:
    indeksmutu="B"
    keterangan="Baik"
elif 68 > nilaiakhir >= 56:
    indeksmutu="C"
    keterangan="Cukup"
```

```
elif 56 > nilaiakhir >= 45:
    indeksmutu="D"
    keterangan="Kurang"
elif 45 > nilaiakhir >= 0:
    indeksmutu="E"
    keterangan="Sangat Kurang"
else:
    indeksmutu="X"
    keterangan="Nilai tidak sah"
print(f"Indeks Mutu : {indeksmutu}")
print(f"Keterangan  : {keterangan}")
```



OPERATOR NOT

01153 - Algoritma dan Struktur Data 1

- Operator logika not digunakan untuk membalikkan nilai Boolean.
- Jika anda punya variable/nilai True, maka jika not True akan menghasilkan False. Begitu juga sebaliknya, jika anda mempunyai variable/nilai False, maka jika not False akan menghasilkan True.



Oleh : Andri Heryandi, M.T.

OPERATOR NOT

01153 - Algoritma dan Struktur Data 1

■ Contoh :

- Buatlah sebuah aplikasi yang akan menampilkan status seseorang apakah masih anak kecil atau sudah sudah dewasa.
- Input : Umur
- Proses : Seorang disebut masih anak kecil kalau masih berumur kurang dari 17 tahun. Jika telah berumur lebih besar atau sama dengan 17 maka dianggap dewasa.
- Output : Menampilkan status orang (anak-anak/dewasa).

Hasil Eksekusi

Umur : 7
Anda masih anak-anak

Umur : 20
Anda telah dewasa



OPERATOR NOT

01153 - Algoritma dan Struktur Data 1

■ Contoh:

```
umur = int(input("Umur : "))
dewasa = umur >= 17
#print(f"Dewasa : {dewasa}")
if not dewasa: #boleh diganti dengan "if dewasa == False:"
    print("Anda masih anak-anak")
else:
    print("Anda sudah dewasa")
```



PERCABANGAN BERSARANG (NESTED BRANCHING)

01153 - Algoritma dan Struktur Data 1

- Percabangan bersarang adalah percabangan yang berada dalam percabangan lain (if-else dalam if-else)
- Percabangan bersarang digunakan ketika anda memiliki percabangan yang memerlukan percabangan lain di dalamnya.
- Sintak penulisannya adalah :

```
if kondisi then
    if kondisi_lain then
        pernyataan/statement 1
    else
        pernyataan/statement 2
    endif {end if dari if bagian dalam}
else
    if kondisi_lain then
        pernyataan/statement 1
    else
        pernyataan/statement 2
    endif {end if dari if bagian dalam}
endif {end if dari if bagian luar}
```

IF dalam IF

PERCABANGAN BERSARANG (NESTED BRANCHING)

01153 - Algoritma dan Struktur Data 1

■ Contoh Kasus :

- Buatlah suatu aplikasi untuk memeriksa apakah seorang warga negara Indonesia berhak untuk ikut pemilihan umum atau tidak.
- Input : Umur dan status nikah. Status nikah hanya ditanyakan Ketika umur masih dibawah 17 tahun.
- Proses :
 - Seorang warga negara Indonesia boleh ikut pemilu jika telah mencapai umur 17 tahun atau sudah menikah.
- Output : Menampilkan status boleh atau tidaknya ikut pemilihan umum.

Hasil Eksekusi

Umur : 20
Anda boleh ikut pemilu

Umur : 16
Menikah [Y/T] : I
Anda belum boleh ikut pemilu.

Umur : 15
Menikah [Y/T] : Y
Anda boleh ikut pemilu.



PERCABANGAN BERSARANG (NESTED BRANCHING)

01153 - Algoritma dan Struktur Data 1

■ Contoh Algoritma:

Program Seleksi_Pemilu

Kamus

umur : integer
menikah : char

Algoritma

```
input(umur)
if umur<17 then
    input(menikah)
    if menikah="Y" then
        output("Anda boleh ikut pemilu")
    else
        output("Anda belum boleh ikut pemilu")
    end if
else
    output("Anda boleh ikut pemilu")
end if
```



PERCABANGAN BERSARANG (NESTED BRANCHING)

01153 - Algoritma dan Struktur Data 1

■ Contoh Program Python:

```
umur = int(input("Umur : "))
if umur<17:
    menikah = input("Menikah [Y/T] : ").upper()
    if menikah=="Y":
        print("Anda boleh ikut pemilu")
    else:
        print("Anda belum boleh ikut pemilu")
else:
    print("Anda boleh ikut pemilu")
```



Latihan

LATIHAN 1

(TAHUN KABISAT)

01153 - Algoritma dan Struktur Data 1

- **Buatlah suatu program yang akan memeriksa apakah suatu tahun adalah tahun kabisat.**
- **Input : Tahun**
- **Proses : Pemeriksaan status tahun kabisat dengan ketentuan :**
 - **Sebuah tahun disebut kabisat kalau**
 - Tahunnya bukan kelipatan 100 tapi tahunnya merupakan kelipatan 4
 - Tahunnya kelipatan 100 tapi tahunnya juga harus kelipatan 400.
- **Output : Tahun dan status kabisatnya.**

Hasil Eksekusi

Tahun : 1996
Tahun 1996 adalah tahun kabisat

Tahun : 1998
Tahun 1998 bukan tahun kabisat

Tahun : 2000
Tahun 2000 adalah tahun kabisat

Tahun : 1900
Tahun 1900 bukan tahun kabisat



LATIHAN 2

(VALIDASI TANGGAL)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu program yang akan memeriksa validitas sebuah tanggal.
- Input : tanggal, bulan, tahun
- Proses : Pemeriksaan validitas tanggal. Sebuah tanggal disebut valid jika :
 - Bulan bernilai 1 s.d 12
 - Tanggal bernilai 1 s.d tgl maksimal bulan, dimana tanggal maksimal bulan adalah :
 - 31 untuk bulan ke 1, 3, 5, 7, 8, 10, 12
 - 30 untuk bulan ke 4, 6, 9, 11
 - 28 untuk bulan 2, kecuali kalau tahunnya kabisat maka tanggal maksimalnya adalah 29.
- Output : Menampilkan status validitas tanggal (valid atau tidak valid)

Hasil Eksekusi

Tanggal : 20
Bulan : 1
Tahun : 2021
Tanggal 20/1/2021 adalah tanggal valid

Tanggal : 30
Bulan : 2
Tahun : 2021
Tanggal 30/2/2021 bukan tanggal valid

Tanggal : 29
Bulan : 2
Tahun : 2020
Tanggal 29/2/2020 adalah tanggal valid



LATIHAN 3

(PERHITUNGAN TIKET)

01153 - Algoritma dan Struktur Data 1

- Buatlah program perhitungan biaya tiket kereta api.

	Jakarta	Yogyakarta	Surabaya
Ekonomi	100.000	200.000	300.000
Bisnis	400.000	500.000	600.000
Eksekutif	700.000	800.000	900.000

- Input : jurusan, kelas, banyak tiket dan kode voucher. Kode voucher hanya ditanyakan jika Yogyakarta Ekonomi dan Surabaya Eksekutif.
- Proses :
 - Menentukan harga tiket
 - Menghitung Sub Total, Diskon dan Total Bayar
 - Diskon diberikan sebesar 10% jika mengisi kode voucher "PROMO".
- Output :
 - Tampilkan Harga Tiket, Sub Total, Diskon dan Total Bayar.

Kode Kota:
1. Jakarta
2. Yogyakarta
3. Surabaya
Pilihan Kota [1/2/3] ? 1
Kode Kelas :
1. Ekonomi
2. Bisnis
3. Eksekutif
Pilihan Kelas [1/2/3] ? 3
Banyak Tiket : 2
Harga Tiket : Rp. 700000
Sub Total : Rp. 1400000
Diskon : Rp. 0
Total Bayar : Rp. 1400000

Kode Kota:
1. Jakarta
2. Yogyakarta
3. Surabaya
Pilihan Kota [1/2/3] ? 2
Kode Kelas :
1. Ekonomi
2. Bisnis
3. Eksekutif
Pilihan Kelas [1/2/3] ? 1
Banyak Tiket : 3
Kode Voucher : PROMO
Harga Tiket : Rp. 200000
Sub Total : Rp. 600000
Diskon : Rp. 60000
Total Bayar : Rp. 540000

LATIHAN 4

(PERHITUNGAN GAJI KARYAWAN)

01153 - Algoritma dan Struktur Data 1

■ Buatlah program perhitungan gaji karyawan

■ Aturan :

■ Gaji Pokok

- Direktur Rp. 5.000.000
- Manager Rp. 3.000.000
- Staf Rp. 2.000.000

■ Jika status menikah, maka

- Hitung tunjangan keluarga sebesar 20% dari gaji pokok.
- Tanyakan banyaknya anak.
- Tunjangan anak adalah 10% dari gaji pokok, hanya sampai anak ke-3

■ $\text{Gaji Kotor} = \text{Gaji Pokok} + \text{Tunjangan Keluarga} + \text{Tunjangan Anak}$

■ Pajak adalah 10% dari Total Kotor jika status tidak menikah. Jika menikah, pajak sebesar 5%.

■ $\text{Total Pendapatan} : \text{Total Kotor} - \text{Pajak}$

Perhitungan Gaji Karyawan

Nama Pegawai : Acep Sugema

Jabatan : Manager

Menikah (Y/T): I

Gaji Pokok : Rp. 3000000

Tunjangan Keluarga : Rp. 0

Tunjangan Anak : Rp. 0

Gaji Kotor : Rp. 3000000

Pajak (10%) : Rp. 300000

Gaji Bersih : Rp. 2700000

Perhitungan Gaji Karyawan

Nama Pegawai : Budi Palastra

Jabatan : Direktur

Menikah (Y/T): Y

Banyak Anak : 5

Gaji Pokok : Rp. 5000000

Tunjangan Keluarga : Rp. 1000000

Tunjangan Anak : Rp. 1500000

Gaji Kotor : Rp. 7500000

Pajak (5%) : Rp. 375000

Gaji Bersih : Rp. 7125000

LATIHAN 5

(PERMAINAN GUNTING-KERTAS-BATU)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu aplikasi permainan Gunting Kertas Batu yang dimainkan oleh manusia dan komputer..

- Ketentuan Pemenang Permainan

	Gunting	Kertas	Batu
Gunting	Draw	Gunting	Batu
Kertas	Gunting	Draw	Kertas
Batu	Batu	Kertas	Draw

- Input : Pilihan pemain (G, K, B)

- Proses :

- Komputer mengacak pilihan(G, K, B)
- Menentukan siapa pemenang (pemain atau komputer)

- Output : Menampilkan Pemenang

Catatan :

- Untuk pilihan komputer gunakan function **choice** yang ada di module **random**

```
import random
komputer = random.choice(['G','K','B'])
print(komputer)
```

Hasil Eksekusi

Komputer telah memilih. Sekarang giliran anda.
Pilihan Anda (G/K/B) ? **K**
Komputer memilih Gunting. Komputer Menang!

Komputer telah memilih. Sekarang giliran anda.
Pilihan Anda (G/K/B) ? **G**
Komputer memilih Gunting. Draw!

Komputer telah memilih. Sekarang giliran anda.
Pilihan Anda (G/K/B) ? **B**
Komputer memilih Gunting. Anda Menang!



FORUM DISKUSI

01153 - Algoritma dan Struktur Data 1



LMS UNIKOM

<https://lms.unikom.ac.id>



**Group Whatsapp
Perkuliahahan**



Youtube Playlist

<https://unikom.id/YT-ASD1>



Oleh : Andri Heryandi, M.T.