

STRUKTUR PENGULANGAN

OLEH : ANDRI HERYANDI, M.T.



07

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 PENGULANGAN

01153 - Algoritma dan Struktur Data 1

- Struktur Pengulangan disebut juga sebagai loop atau iterasi (iteration)
- Struktur pengulangan memungkinkan programmer menginstruksikan komputer untuk melakukan **1** atau **banyak** instruksi/ Pernyataan secara berulang.
- Pengulangan yang dilakukan bisa :
 - Berdasarkan ketercapaian kondisi pengulangan. Selama kondisi pengulangan masih tercapai, maka pengulangan akan terus dilakukan.
 - Berdasarkan urutan/runtunan/sequence/rangkaian. Pengulangan dilakukan dengan menelusuri urutan dari awal sampai akhir.
- Ada berbagai jenis pengulangan yaitu :
 - For to do, For downto do, Foreach, For in
 - While Do
 - Repeat Until
 - Do While



STRUKTUR PENGULANGAN

01153 - Algoritma dan Struktur Data 1

Algoritma



Oleh : Andri Heryandi, M.T.

STRUKTUR PENGULANGAN (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

- Dalam algoritma notasi pseudo-code, pengulangan bisa dilakukan dengan berbagai cara, yaitu :
 - **for to do, for downto do**
 - **while do**
 - **repeat until**



Oleh : Andri Heryandi, M.T.

STRUKTUR PENGULANGAN FOR (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

- Pengulangan for adalah pengulangan yang akan melakukan pengulangan berdasarkan nilai variable counter-nya.
- Pengulangan for dipilih ketika banyaknya pengulangan telah dispesifikan (diketahui).
- Variable counter-nya secara otomatis bertambah 1 (for to do), atau berkurang 1 (for downto do)
- Sintak :

```
for counter ← awal to akhir do  
    loop-statement  
endfor
```

```
for counter ← awal downto akhir do  
    loop-statement  
endfor
```

- Keterangan
 - **counter** adalah variable pencacah.
 - **awal** adalah nilai awal untuk variable counter.
 - **akhir** adalah nilai akhir untuk variable counter.
 - **Loop-statement** adalah statement yang akan dieksekusi (diulang)



STRUKTUR PENGULANGAN FOR (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh

Program Deret_Angka

Kamus:

i:integer

Algoritma:

```
for i ← 1 to 5 do
    output(i)
endfor
```

Program Deret_Angka

Kamus:

i:integer

Algoritma:

```
for i ← 5 downto 1 do
    output(i)
endfor
```

Hasil eksekusi

1
2
3
4
5

Hasil eksekusi

5
4
3
2
1



STRUKTUR PENGULANGAN WHILE (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

- Pengulangan **while** adalah pengulangan yang akan melakukan pengulangan berdasarkan kondisi (true/false).
- Jika kondisi pengulangan bernilai true, maka badan pengulangan akan dieksekusi.
- Jika kondisi pengulangan bernilai false, maka pengulangan berakhir.
- Pengulangan **while** bisa digunakan ketika banyaknya pengulangan belum diketahui
- Pengulangan **while** akan melakukan pemeriksaan kondisi (true/false) di awal pengulangan. Sehingga ada kemungkinan tidak pernah masuk pengulangan karena kondisi awal = false.

- Sintak :

```
while kondisi do  
    loop-statment  
endwhile
```

- Keterangan

- **kondisi** adalah kondisi pengulangan (bertipe Boolean). Biasanya berisi operasi perbandingan.
- **Loop-statement** adalah statement yang akan dieksekusi (diulang)



STRUKTUR PENGULANGAN WHILE (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 1

Program Deret_Angka

Kamus:

i:real

Algoritma:

i ← 0

while i ≤ 10 do

 output(i)

 i ← i + 2.5 # increment counter

endwhile

Hasil eksekusi

0.00

2.50

5.00

7.50

10.00

STRUKTUR PENGULANGAN WHILE (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 2

Program Hitung_Total

Kamus:

i,total:real

lagi:char

Algoritma:

i ← 0

total ← 0

lagi ← 'Y'

while lagi = 'Y' do

 output("Input Angka : ")

 input(i)

 total ← total + i

 output("Total Sekarang ", total)

 output("Lagi ? ")

 input(lagi)

endwhile

Hasil eksekusi

Input Angka : 10

Total sekarang 10

Lagi ? Y

Input Angka : 14

Total sekarang 24

Lagi ? Y

Input Angka : 7

Total sekarang 31

Lagi ? Y

Input Angka : 25

Total sekarang 56

Lagi ? I

STRUKTUR PENGULANGAN REPEAT (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

- Pengulangan **repeat** adalah pengulangan yang akan melakukan pengulangan berdasarkan kondisi (true/false).
- Jika kondisi pengulangan bernilai false, maka badan pengulangan akan dieksekusi.
- Jika kondisi pengulangan bernilai true, maka pengulangan berakhir.
- Pengulangan **repeat** bisa digunakan ketika banyaknya pengulangan belum diketahui
- Pengulangan **repeat** akan melakukan pemeriksaan kondisi (true/false) di akhir badan pengulangan. Sehingga dipastikan minimal 1 kali pernah masuk ke pengulangan.

■ Sintak :

```
repeat  
    loop-statment  
until kondisi
```

■ Keterangan

- **kondisi** adalah kondisi pengulangan (bertipe Boolean). Biasanya berisi operasi perbandingan.
- **Loop-statement** adalah statement yang akan dieksekusi (diulang)



STRUKTUR PENGULANGAN REPEAT (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 1

Program Deret_Angka

Kamus:

i:real

Algoritma:

i ← 0

repeat

 output(i)

 i ← i + 2.5 # increment counter

until i > 10

Hasil eksekusi

0.00

2.50

5.00

7.50

10.00



STRUKTUR PENGULANGAN REPEAT (ALGORITMA)

01153 - Algoritma dan Struktur Data 1

■ Contoh 2

Program Hitung_Total

Kamus:

i,total:real

lagi:char

Algoritma:

i ← 0

total ← 0

repeat

 output("Input Angka : ")

 input(i)

 total ← total + i

 output("Total Sekarang ", total)

 output("Lagi ? ")

 input(lagi)

until lagi <> 'Y'

Hasil eksekusi

Input Angka : 10

Total sekarang 10

Lagi ? Y

Input Angka : 14

Total sekarang 24

Lagi ? Y

Input Angka : 7

Total sekarang 31

Lagi ? Y

Input Angka : 25

Total sekarang 56

Lagi ? I



STRUKTUR PENGULANGAN

01153 - Algoritma dan Struktur Data 1

Python



Oleh : Andri Heryandi, M.T.

STRUKTUR PENGULANGAN (PYTHON)

01153 - Algoritma dan Struktur Data 1

- Di python, ada 2 jenis struktur pengulangan, yaitu :
 - **while** : Pengulangan ini digunakan untuk melakukan pengulangan berdasarkan kondisi tertentu. Selama kondisi masih terpenuhi (True) maka instruksi yang ada di dalam badan akan dijalankan. Jika kondisi sudah tidak terpenuhi (False) maka instruksi yang berada dalam badan pengulangan tidak akan diulang.
 - **for** : Pengulangan ini digunakan untuk menelusuri (traversal) suatu urutan/sederetan/*sequence*.



PENGULANGAN WHILE

01153 - Algoritma dan Struktur Data 1

- Pengulangan while adalah struktur yang akan melakukan eksekusi satu atau sekumpulan perintah (statement) selama kondisi pengulangan masih bernilai True.

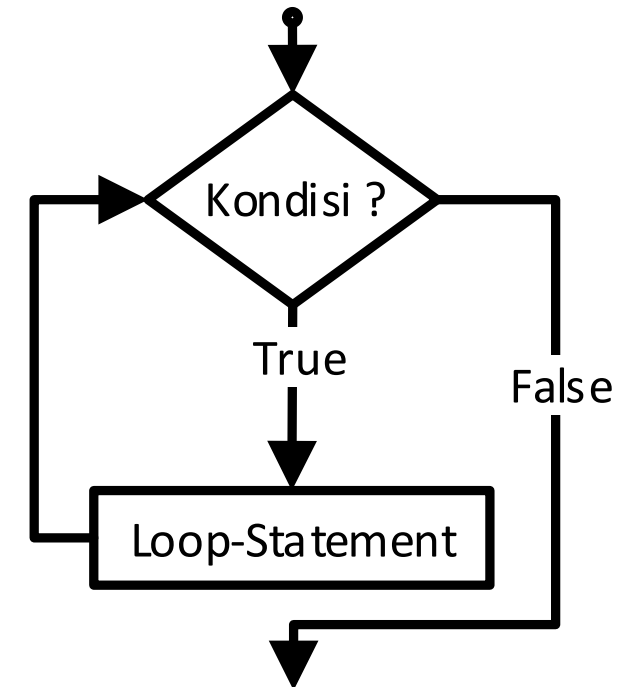
- Sintaknya adalah :

```
while kondisi:  
    loop-statement  
else:  
    else-statement
```

- Keterangan :

- **Kondisi** bernilai Boolean (True / False). Boleh menggunakan nama variable atau pun perbandingan.
- **Loop-statement** adalah statement/perintah yang akan diulang.
- **Else** dan **else-statement** adalah statement yang akan dieksekusi jika pengulangan telah selesai. Bagian ini boleh tidak ada.

Flowchart Pengulangan



PENGULANGAN WHILE

01153 - Algoritma dan Struktur Data 1

■ Contoh 1:

Pengulangan untuk menampilkan angka 1 sampai 10.

```
i = 1
while i<=10:
    print(i)
    i += 1 # i = i + 1
else:
    print("Selesai")
```

Keterangan :

Assignment Operator

- $i += 1 \rightarrow i = i + 1$
- $i += 2 \rightarrow i = i + 2$
- $x *= 5 \rightarrow x = x * 5$
- $x /= 2 \rightarrow x = x / 2$

Hasil eksekusi

```
1
2
3
4
5
6
7
8
9
10
Selesai
```

PENGULANGAN WHILE

01153 - Algoritma dan Struktur Data 1

■ Contoh 2:

Buatlah suatu aplikasi tebak angka. Angka yang akan ditebak didapat dari hasil random oleh komputer. Angka yang akan ditebak adalah angka 1 s.d 10.

Jika tebakan salah, maka tampilkan bahwa tebakan terlalu besar atau terlalu kecil.

Hasil eksekusi

```
Tebakan anda [1..10] ? 10
Salah. Tebakan Anda Terlalu Besar
Tebakan anda [1..10] ? 5
Salah. Tebakan Anda Terlalu Besar
Tebakan anda [1..10] ? 3
Salah. Tebakan Anda Terlalu Kecil
Tebakan anda [1..10] ? 4
Selamat. Tebakan Anda Benar
```



PENGULANGAN WHILE

01153 - Algoritma dan Struktur Data 1

■ Contoh 2:

```
import random
kunci = random.randint(1,10)
tebakan=0
while(tebakan!=kunci):
    tebakan=int(input("Tebakan anda [1..10] ? "))
    if tebakan==kunci:
        print("Selamat. Tebakan Anda Benar")
    elif tebakan > kunci:
        print("Salah. Tebakan Anda Terlalu Besar")
    else:
        print("Salah. Tebakan Anda Terlalu Kecil")
```



PENGULANGAN WHILE

01153 - Algoritma dan Struktur Data 1

- Hal-hal penting ketika menggunakan pengulangan while.
 - Jika kondisi awal sudah bernilai False, maka ada kemungkinan tidak akan pernah masuk ke dalam perulangan.

```
i = 20
while i <= 10:
    print(i)
    i+=1
print("Bye")
```

Hasil eksekusi

Bye

- Jangan lupa melakukan modifikasi terhadap variable yang menjadi perbandingan dalam kondisi pengulangan. Jika anda lupa, ada kemungkinan terjadi infinite loop (pengulangan tanpa henti)

```
i = 1
while i <= 10:
    print(i, end=" ")
```

Hasil eksekusi

[illegible]

PENGULANGAN FOR

01153 - Algoritma dan Struktur Data 1

- Pengulangan for di python berbeda dengan pengulangan for di Bahasa lain seperti C, java, atau pun Pascal.
- Pengulangan for di python digunakan untuk melakukan penelusuran dari suatu deretan/urutan/sequence data seperti list, tuple, dictionary, set, string atau objek/tipe iterable objek lain.
- Sintaknya adalah :

```
for val in iterable:  
    loop-statement  
else:  
    else-statement
```

Lebih lengkap mengenai List, Tuple, Dictionary, Set akan dijelaskan di bab lain.

- Keterangan :
 - **val** : variable yang akan menerima nilai (value) dari item iterable
 - **iterable** : variable yang berisi deretan data
 - **else** dan **else-statement** adalah statement yang akan dijalankan ketika pengulangan telah berhasil.



PENGULANGAN FOR

01153 - Algoritma dan Struktur Data 1

■ Contoh :

Pengulangan for (menggunakan struktur data list).

```
ibukota = ["Bandung", "Semarang", "Yogyakarta", "Surabaya"]  
for kota in ibukota:  
    print(kota)  
else:  
    print("-" * 20)
```

Hasil eksekusi

```
Bandung  
Semarang  
Yogyakarta  
Surabaya  
-----
```



PENGULANGAN FOR

01153 - Algoritma dan Struktur Data 1

■ Contoh :

Pengulangan for (menggunakan struktur data tuple).

```
daftar_nilai = (70, 80, 50, 67, 25)
for nilai in daftar_nilai:
    print(nilai)
```

Hasil eksekusi

```
70
80
50
67
25
```



PENGULANGAN FOR

01153 - Algoritma dan Struktur Data 1

■ Contoh :

Pengulangan for (menggunakan struktur data string).

```
nama = "Asep"  
for huruf in nama:  
    print(huruf)
```

Hasil eksekusi

```
A  
s  
e  
p
```



PENGULANGAN FOR

01153 - Algoritma dan Struktur Data 1

■ Contoh :

Pengulangan for untuk melakukan perhitungan data.

```
daftar_nilai = [70, 80, 50, 67, 25]
total = 0
for nilai in daftar_nilai:
    total += nilai
rata_rata = total / len(daftar_nilai)
print("Total      : {}".format(total))
print("Rata-rata  : {}".format(rata_rata))
```

Hasil eksekusi

```
Total      : 292
Rata-rata  : 58.4
```



PENGULANGAN FOR (RANGE)

01153 - Algoritma dan Struktur Data 1

- Jika anda ingin membuat pengulangan berdasarkan deretan angka, maka deretan angka tersebut dapat anda buat dengan menggunakan class range.
- Tipe/class range berisi deretan/urutan angka yang tidak dapat diubah.
- Parameter untuk membuat range adalah
 - Class range(stop) atau range(start, stop, [step])
 - Stop : angka terakhir. Jika hanya mengisi stop saja, maka deretan angka akan dimulai dari 0.
 - Start : angka awal deretan. Jika start tidak diisi, maka defaultnya adalah 0.
 - Step : perubahan nilai dari satu item ke item lainnya. Jika tidak ditulis, maka defaultnya adalah 1



PENGULANGAN FOR (RANGE)

01153 - Algoritma dan Struktur Data 1

■ Contoh :

```
>>> range(10)
range(0, 10)
>>> list(range(10))
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
>>> list(range(5,10))
[5, 6, 7, 8, 9]
>>> list(range(5,20,2))
[5, 7, 9, 11, 13, 15, 17, 19]
>>> list(range(-1,-10,-1))
[-1, -2, -3, -4, -5, -6, -7, -8, -9]
```



PENGULANGAN FOR (RANGE)

01153 - Algoritma dan Struktur Data 1

■ Contoh :

Pengulangan for menggunakan range.

```
for i in range(5):  
    print(i)  
print("-" * 10)  
for i in range(20,25):  
    print(i)  
print("+" * 10)  
for i in range(50,100,10):  
    print(i)
```

Hasil eksekusi

```
0  
1  
2  
3  
4  
-----  
20  
21  
22  
23  
24  
+++++++  
50  
60  
70  
80  
90
```



STATEMENT BREAK

01153 - Algoritma dan Struktur Data 1

- Statement **break** digunakan untuk keluar dari perulangan, walau pun pengulangan belum berhasil.
- Jika suatu pengulangan mengeksekusi **break**, maka bagian else (jika ada) tidak akan dieksekusi.

```
for i in range(1,6):  
    if i == 3:  
        break  
    print(i)  
else:  
    print("Selesai")
```

Hasil eksekusi

1
2



STATEMENT CONTINUE

01153 - Algoritma dan Struktur Data 1

- Statement **continue** digunakan untuk meneruskan pengulangan ke item selanjutnya dengan mengabaikan perintah setelahnya.

```
for i in range(1,6):  
    if i == 3:  
        continue # lanjutkan ke item selanjutnya  
    print(i)  
else:  
    print("Selesai")
```

Hasil eksekusi

```
1  
2  
4  
5  
Selesai
```



STATEMENT PASS

01153 - Algoritma dan Struktur Data 1

- Statement **pass** tidak melakukan apa-apa.
- Statement ini dapat digunakan ketika pernyataan diperlukan secara sintaksis tetapi program tidak memerlukan tindakan.

```
for i in range(1,6):  
    if i == 3:  
        pass #jangan lakukan apa-pun  
    else:  
        print(i)  
else:  
    print("Selesai")
```

Hasil eksekusi

```
1  
2  
4  
5  
Selesai
```



PENGULANGAN BERSARANG

01153 - Algoritma dan Struktur Data 1

- Pengulangan bersarang (nested loop) adalah pengulangan yang didalamnya mengandung pengulangan lain (pengulangan dalam pengulangan)
- Untuk pengulangan yang digunakan, Anda boleh menggunakan pengulangan while, atau pun for atau pun campuran antara while dan for.
- Jika diasumsikan ada 2 buah pengulangan yaitu pengulangan A dan B, dimana pengulangan B berada di dalam pengulangan A, maka :
 - Pengulangan A berada di bagian luar
 - Pengulangan B berada di bagian dalam
 - Pengulangan B harus selesai dulu, maka pengulangan A akan melanjutkan ke tahap selanjutnya.



PENGULANGAN BERSARANG

01153 - Algoritma dan Struktur Data 1

■ Contoh 1 (for di dalam for):

```
for a in range(1,6):  
    for b in range(1,4):  
        print("{} x {} = {}".format(a,b,a*b))  
    print("-"*10)
```

■ Keterangan :

- Pengulangan a baru akan melanjutkan ke tahap selanjutnya setelah pengulangan b selesai mengulang 1 s.d 3

Hasil eksekusi

```
1 x 1 = 1  
1 x 2 = 2  
1 x 3 = 3  
-----  
2 x 1 = 2  
2 x 2 = 4  
2 x 3 = 6  
-----  
3 x 1 = 3  
3 x 2 = 6  
3 x 3 = 9  
-----  
4 x 1 = 4  
4 x 2 = 8  
4 x 3 = 12  
-----  
5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
-----
```



PENGULANGAN BERSARANG

01153 - Algoritma dan Struktur Data 1

■ Contoh 2 (for di dalam while):

```
a = 1
while a<=5:
    for b in range(1,4):
        print("{} x {} = {}".format(a,b,a*b))
    print("-"*10)
    a += 1
```

■ Keterangan :

- Pengulangan a baru akan melanjutkan ke tahap selanjutnya setelah pengulangan b selesai mengulang 1 s.d 3

Hasil eksekusi

```
1 x 1 = 1
1 x 2 = 2
1 x 3 = 3
-----
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
-----
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
-----
4 x 1 = 4
4 x 2 = 8
4 x 3 = 12
-----
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
-----
```



Latihan

LATIHAN 1

(PERHITUNGAN TABUNGAN)

01153 - Algoritma dan Struktur Data 1

- Buatlah sebuah aplikasi yang dapat menghitung rincian saldo suatu tabungan untuk setiap bulannya.
- Input : saldo awal, bunga, jangka waktu
- Proses : Menghitung saldo akhir di setiap bulannya.
- Output : Menampilkan saldo akhir di setiap bulan.

```
Saldo Awal    : Rp. 100000 [input]
Bunga (%)      : 10         [input]
Jangka Waktu  : 3          [input]
-----
Saldo Akhir bulan ke-1 : Rp. 110000
Saldo Akhir bulan ke-2 : Rp. 121000
Saldo Akhir bulan ke-3 : Rp. 133100
```



LATIHAN 2

(MENGHITUNG HURUF VOKAL)

01153 - Algoritma dan Struktur Data 1

- Hitung berapa huruf vocal yang berada dalam suatu kalimat. Huruf kapital dan huruf kecil dianggap sama.

Hasil Eksekusi

Kalimat : SAYA SEDANG Belajar Python

Huruf vokal ada 8 (AAEAeaa)



LATIHAN 3

(PERMAINAN GUNTING-KERTAS-BATU)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu aplikasi permainan Gunting Kertas Batu yang dimainkan oleh manusia dan komputer. Setiap permainan terdiri dari 3 kali kesempatan.

- 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 dengan score terbanyak.

Hasil Eksekusi

Game Ke-1

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

Komputer memilih Gunting. Komputer Menang!

Score [Anda] 0 - [Komputer] 1

Game Ke-2

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

Komputer memilih Gunting. Draw!

Score [Anda] 0 - [Komputer] 1

Game Ke-3

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

Komputer memilih Gunting. Komputer Menang!

Score [Anda] 0 - [Komputer] 2

Game Over. Komputer Menang



Oleh : Andri Heryandi, M.T.

LATIHAN 4

(PERMAINAN TEBAK ANGKA)

01153 - Algoritma dan Struktur Data 1

- Modifikasi aplikasi permainan tebak angka (contoh 2 di materi pengulangan while). Tambahkan fitur level dan pembatasan banyaknya percobaan.
- Aturan :
 - Setiap level akan makin rumit karena batas bilangan acaknya diperbanyak. Setiap naik level, maka batas angka yang diacak akan bertambah 10. Contoh jika mengisi level 1 maka angka yang diacak adalah 1 s.d 10, jika level 2 maka angka yang diacak adalah 1 s.d 20, dst.
 - Banyak percobaan tiap level adalah 5 kali. Jika dalam 5 kali tidak bisa menebak dengan benar, maka permainan selesai.
 - Setiap ganti level, maka banyak percobaan direset kembali.

Hasil Eksekusi

```
Level 1 [1..10]
Tebakan Anda ? 2
Terlalu Kecil
Tebakan Anda ? 5
Terlalu Besar
Tebakan Anda ? 4
Selamat. Tebakan Anda Benar

Level 2 [1..20]
Tebakan Anda ? 1
Terlalu Kecil
Tebakan Anda ? 10
Terlalu Besar
Tebakan Anda ? 3
Terlalu Kecil
Tebakan Anda ? 9
Terlalu Besar
Tebakan Anda ? 5
Terlalu Kecil.
Game Over. Anda telah salah menebak 5 kali.
Angka rahasia adalah 7
```



LATIHAN 5

(BENTUK SEGITIGA)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu aplikasi yang akan menampilkan tampilan teks berbentuk segitiga (lihat contoh) dengan input berupa angka (N).

Hasil Eksekusi

```
N : 5
-----
1:  *
2:  ***
3:  *****
4:  *****
5:  *****
```



LATIHAN 6

(REKAPITULASI DATA)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu aplikasi yang dapat melakukan input data dan melakukan penjumlahan berdasarkan data tersebut. Pembacaan data dilakukan kalau data yang diinputkan tidak negatif.

- Input : Data yang akan dihitung

- Proses :

- Hitung total
- Hitung Rata-Rata

- Output :

- Tampilkan Total
- Tampilkan Banyak Data
- Tampilkan Rata-Rata

Hasil eksekusi

Masukan Data Ke-1 : 6

Masukan Data Ke-2 : 8

Masukan Data Ke-3 : 0

Masukan Data Ke-4 : 10

Masukan Data Ke-5 : 3

Masukan Data Ke-6 : -1

Total : 27

Banyak Data : 5

Rata-Rata : 5.4



LATIHAN 7

(PERHITUNGAN HARI)

01153 - Algoritma dan Struktur Data 1

- Buatlah suatu program untuk menghitung hari ke berapa sejak 01 Januari 1900.
- Tanggal 01 Januari 1900 menghasilkan angka 1, tanggal 02 Januari 1900 menghasilkan angka 2, dst.
- Input : Tanggal [YYYY-MM-DD]
- Output : Banyak hari sejak 01 Januari 1900

Tanggal [YYYY-MM-DD] : 1900-01-01
Hari ke 1

Tanggal [YYYY-MM-DD] : 1900-01-02
Hari ke 2

Tanggal [YYYY-MM-DD] : 1900-02-01
Hari ke 32

Tanggal [YYYY-MM-DD] : 1945-08-17
Hari ke 16665

Tanggal [YYYY-MM-DD] : 2021-02-29
Tanggal Tidak Sah.

Tanggal [YYYY-MM-DD] : 2021-11-10
Hari ke 44509



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.