

**KUIS 2 – ALGORITMA DAN PEMROGRAMAN**  
(Dosen Pengampu : Muhammad Fakhurrifqi, S.Kom., M.Cs.)

Nama : Maulana Adam Sahid  
NIM : 18/431735/SV/15706

## 1. PRODUKSI BATIK

Batas Run-time: 1 detik / test-case  
Batas Memori: 32 MB

### DESKRIPSI SOAL

Pak Blangkon kini sedang mencoba mengembangkan startup yang dia bangun bertemakan batik, yang dia beri nama "Butik Batik Batuk Batuk". Proses bisnisnya sederhana, pelanggan bisa menentukan desain batik yang diinginkan melalui web. Lalu setelah bayar, maka Pak Blangkon akan mengerahkan pegawainya untuk membatik sesuai desain yang dipesan.

Ketika memproduksi batik, Pak Blangkon harus menyiapkan beberapa sumber daya, yakni kain, lilin, dan pewarna. Untuk membuat satu produk batik. Dibutuhkan 2 meter persegi kain, 3 batang lilin, dan 2 botol pewarna. Untuk itu, bantulah Pak Blangkon untuk menambahkan fitur pada aplikasi webnya yang akan menampilkan banyak pesanan batik yang bisa Pak Blangkon kerjakan pada hari itu berdasarkan stok bahan baku yang masih ada.

### PETUNJUK MASUKAN

Pada baris pertama akan diberikan 3 buah bilangan bulat A, B, dan C. Artinya saat itu Pak Blangkon hanya memiliki stok A m<sup>2</sup> kain, B batang lilin, dan C botol pewarna. ( $0 \leq A, B, C \leq 100$ )

### PETUNJUK KELUARAN

Outputkan sebuah bilangan bulat banyak pesanan yang bisa dibuat Pak Blangkon.

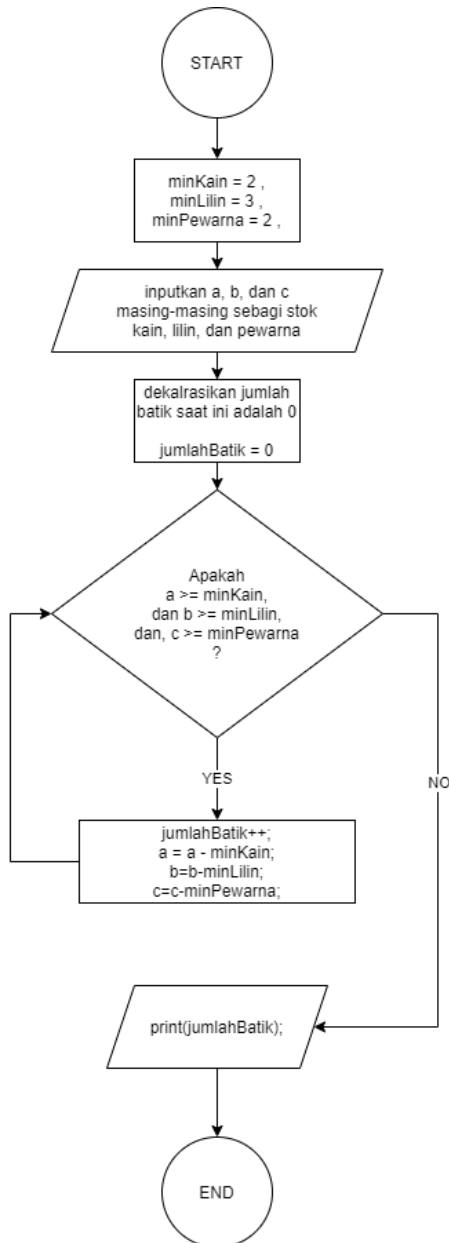
### CONTOH MASUKAN

8 7 8

### CONTOH KELUARAN

2

Jawaban:



### Implementasi dalam Python 3 :

```
#Untuk membuat batik dibutuhkan:
# 2m kain, 3 lilin, dan 2 pewarna

minKain=2;
minLilin=3;
minPewarna=2;

#input stok kain, lilin, dan pewarna
a, b, c=input().split();

#merubah dari string ke integer
a=int(a)
b=int(b)
c=int(c)

#inisiasi jumlah batik di awal
jumlahBatik=0;

#pengecekan stok dan proses produksi
while a >= minKain and b >= minLilin and c >= minPewarna:
    jumlahBatik=jumlahBatik+1;
    a=a-minKain;
    b=b-minLilin;
    c=c-minPewarna;

#outputkan batik yang sudah diproduksi
print(jumlahBatik)

#Contoh input      : 8 7 8 || 3 2 4 || 10 21 3 || 101 102 103 || 997 1500 1000
#Ekspektasi output : 2 || 0 || 1 || 34 || 498
```

Algoritma:

1. Mulai program.
2. Deklarasikan variabel minimal kain, minimal lilin, dan minimal pewarna yang dibutuhkan untuk membuat tiap satu buah baju batik.
3. Inputkan A, B, dan C masing-masing secara berurutan adalah sebagai representasi stok kain, lilin, dan pewarna yang dimiliki Pak Blangkon pada hari produksi tsb.
4. Deklarasikan jumlah awal produksi batik pada hari itu, yaitu 0.
5. Kemudian check apakah stok A memenuhi lebih dari sama dengan minimal kain yang dibutuhkan, dan apakah stok B memenuhi lebih dari samadengan minimal lilin yang dibutuhkan, dan apakah stok C memenuhi lebih dari sama dengan minimal pewarna yang dibutuhkan. Jika tidak, skip dan lanjut ke langkah ke-11. Jika ya maka ikuti langkah berikutnya.
6. Tambahkan jumlah produksi batik sebanyak 1 inkremen.
7. Kurangi stok A dengan dekremen sebanyak nilai minimal kain.
8. Kurangi stok B dengan dekremen sebanyak nilai minimal lilin.
9. Kurangi stok C dengan dekremen sebanyak nilai minimal pewarna.
10. Kembali ke langkah 5.
11. Print jumlah batik.
12. Akhiri program.

## Hasil Pengujian:

```
Produksi Batik x
"C:\Users\masad\Documents\UGM 2021\GEN
8 7 8
2
Process finished with exit code 0
```

```
Produksi Batik x
"C:\Users\masad\Documents\UGM 2021\
3 2 4
0
Process finished with exit code 0
```

```
Produksi Batik x
"C:\Users\masad\Documents\UGM 2021\
10 21 3
1
Process finished with exit code 0
```

```
Produksi Batik x
"C:\Users\masad\Documents\UGM 2021\
10 21 3
1
Process finished with exit code 0
```

```
Produksi Batik x
"C:\Users\masad\Documents\UGM 2021\G
997 1500 1000
498
Process finished with exit code 0
```

## 2. BESAR MANA SIH?

Batas Run-time: 1 detik / test-case

Batas Memori: 32 MB

### DESKRIPSI SOAL

Bagong kecil dibuat bingung dengan pembagian pizza oleh Pakde Semar. Bagong kecil yang memperoleh hanya 8/15 bagian merasa bagian yang ia peroleh lebih sedikit dari yang didapat Pakde Semar, yakni 9/20 bagian. Bantulah Bagong kecil untuk memahami pembandingan pecahan dengan membuat program sederhana.

### PETUNJUK MASUKAN

Empat buah bilangan A, B, C, dan D. ( $1 \leq A, B, C, D \leq 1000$ )

### PETUNJUK KELUARAN

Jika  $A/B$  lebih besar dari  $C/D$ , outputkan ">". Jika lebih kecil outputkan "<" dan jika sama outputkan "=".

### CONTOH MASUKAN 1

8 15 9 20

### CONTOH KELUARAN 1

>

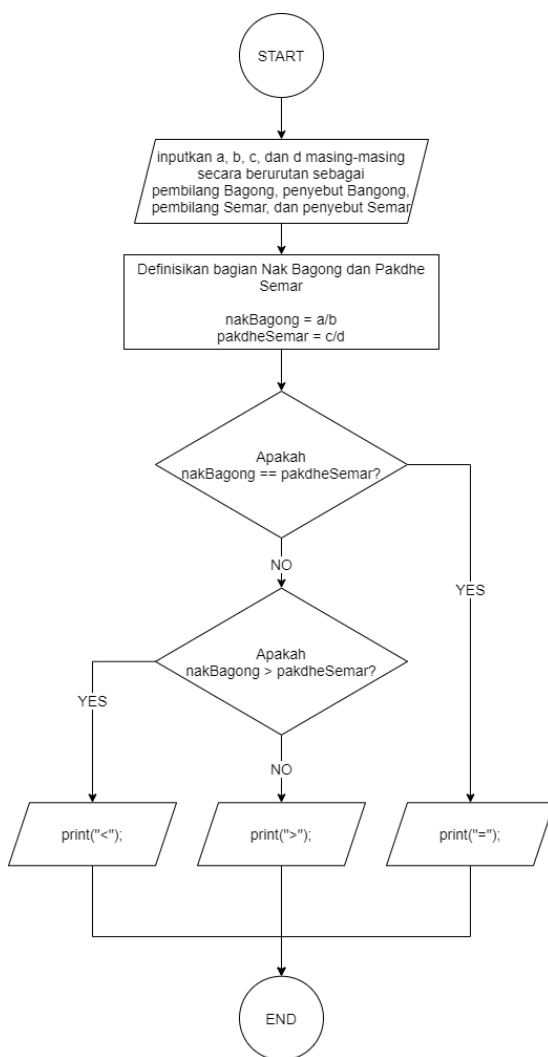
### CONTOH MASUKAN 2

4 2 100 50

### CONTOH KELUARAN 2

=

Jawaban :



### Implementasi dalam Python 3 :

```
# input A B C dan D
a, b, c, d=input().split()

#rubah str ke float
a=float(a)
b=float(b)
c=float(c)
d=float(d)

nakBagong=float(a/b)
paktdeSemar=float(c/d)

if nakBagong == paktdeSemar:
    print("=")
else:
    if nakBagong > paktdeSemar:
        print(">")
    else:
        print("<")

# CONTOH MASUKAN 1
# 8 15 9 20
# CONTOH KELUARAN 1
# >
# CONTOH MASUKAN 2
# 4 2 100 50
# CONTOH KELUARAN 2
# =
```

### Algoritma:

1. Mulai program.
2. Inputkan A, B, C, dan D masing-masing secara berurutan adalah sebagai representasi pembilang bagian Bagong, penyebut bagian Bagong, pembilang bagian Semar, dan penyebut bagian Semar.
3. Definisikan variabel bagian masing-masing Nak Bagong dan Pakdhe Semar. Bagian Nak Bagong adalah A dibagi B, sedangkan bagian Pakdhe Semar adalah C dibagi D.
4. Kemudian check apakah bagian Nak Bagong sama dengan bagian Pakdhe Semar. Jika ya, maka print tanda sama dengan "=" dan lompat ke langkah 6. Jika tidak, maka lanjut ke langkah berikutnya.
5. Apakah bagian Nak Bagong lebih besar dari bagian dari bagian Pak Semar. Jika ya, maka print tanda lebih besar dari ">" dan langsung lanjut ke langkah 6. Jika tidak, maka print tanda kurang dari sama dengan "<" dan lanjut ke langkah berikutnya.
6. Akhiri program.

## Hasil Pengujian:

```
Besar Mana Sih x
"C:\Users\masad\Documents\UGM 2021\
10 21 56 261
>
Process finished with exit code 0
```

```
Besar Mana Sih x
"C:\Users\masad\Documents\UGM 2021\GEI
230 50 120 25
<
Process finished with exit code 0
```

```
Besar Mana Sih x
"C:\Users\masad\Documents\UGM 2021\GEI
8 15 9 20
>
Process finished with exit code 0
```

```
Besar Mana Sih x
"C:\Users\masad\Documents\UGM 2021\
2512 42 3768 63
=
Process finished with exit code 0
```

```
Besar Mana Sih x
"C:\Users\masad\Documents\UGM 2021\GEI
4 2 100 50
=
Process finished with exit code 0
```

### 3. ULAR TANGGA

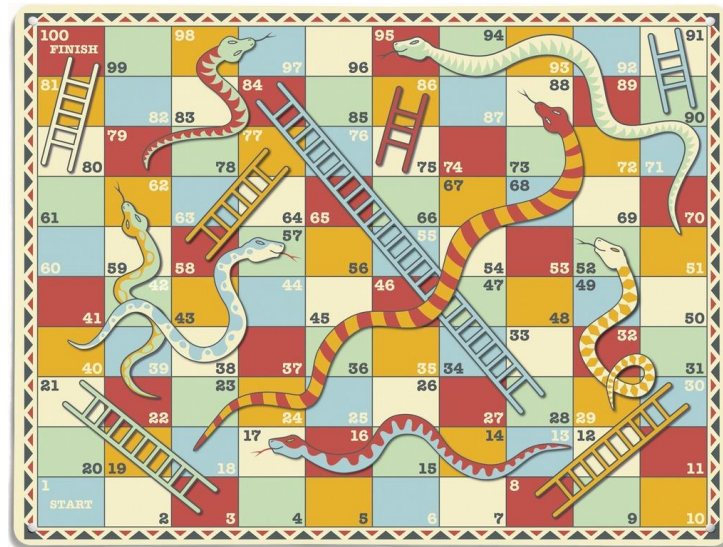
Batas Run-time: 1 detik / test-case  
Batas Memori: 32 MB

### DESKRIPSI SOAL

Perhatikan papan permainan ular tangga di atas. Buatlah program yang jika diinputkan posisi seorang pemain dan angka dadu yang dia peroleh, program dapat mengoutputkan posisi sekarang saat ini.

Aturan ular tangga sama pada umumnya, hanya saja ada beberapa catatan:

- Jika pemain dapat mencapai posisi lebih dari sama dengan 100 maka dianggap telah menang.
- Ekor ular yang kepalanya pada petak nomor 17, berada pada petak 15



## PETUNJUK MASUKAN

Dua buah bilangan bulat, posisi pemain dan angka dadu yang dia peroleh. Posisi pemain ada diantara 1-100 dan angka dadu yang diperoleh antara 1-6.

## PETUNJUK KELUARAN

Sebuah bilangan bulat yang merupakan posisi dia sekarang. Jika posisinya lebih dari 100 maka outputkan "MENANG"

## CONTOH MASUKAN

98 3

## CONTOH KELUARAN

MENANG

### CONTOH MASUKAN

53

## CONTOH KELUARAN

30

Jawaban :

### Implementasi dalam Python 3 :

Dalam implementasi kasus ini, dapat kita gunakan dua cara. Dapat dengan menggunakan percabangan if dan else secara manual. Tetapi juga bisa dengan cara membuat fungsi switch/dictionary terlebih dahulu.

### MENGGUNAKAN PERCABANGAN IF DAN ELSE

Untuk cara pertama langkah algoritmanya sebagai berikut:

1. Mulai program.
2. Inputkan posisi awal dan dadu yang dilempar (dan pastikan dalam tipe data integer).
3. Definisikan variabel posisi akhir dengan valuenya adalah hasil penjumlahan posisi awal dengan dadu.
4. Kemudian cek apakah posisi akhir memenuhi lebih dari 100. Jika ya/benar, maka deklarasikan variabel status dengan value string "MENANG" dan langsung lanjut melompat ke langkah 7. Jika tidak/salah, maka lanjut ke langkah 5 sesuai urutan.
5. Apakah posisi akhir adalah 3? Jika benar, maka ubah posisi akhir dengan 21. Jika tidak, apakah posisi akhir adalah 8? Jika benar, maka ubah posisi akhir dengan 30. Jika tidak ..., dst. hingga kondisi khusus terakhir sesuai pada papan dan keterangan dalam soal.
6. Kemudian deklarasikan variabel status dengan value adalah posisi akhir.
7. Outputkan/print status.
8. Akhiri program

```
#Input posisi Awal
posisiAwal, dadu = input().split()

#ubah ke Integer
posisiAwal = int(posisiAwal)
dadu = int(dadu)

#tentukan posisi setelah melempar dadu
posisiAkhir = posisiAwal + dadu

#cek apakah posisi akhir berada pada posisi tertentu
if posisiAkhir >= 100 or posisiAkhir == 80:
    status = "MENANG" #Bila terpenuhi maka statusnya adalah MENANG
else:
    if posisiAkhir == 3: # Kondisi Naik
        posisiAkhir = 21
    elif posisiAkhir == 8:
        posisiAkhir = 30
    elif posisiAkhir == 28:
        posisiAkhir = 84
    elif posisiAkhir == 58:
```

Repositori tugas <https://github.com/NPC-7/Alpro-Genap-2021>



```

        posisiAkhir = 77
    elif posisiAkhir == 75:
        posisiAkhir = 86
    elif posisiAkhir == 90:
        posisiAkhir = 91

    elif posisiAkhir == 97: # Kondisi Turun
        posisiAkhir = 79
    elif posisiAkhir == 95:
        posisiAkhir = 51
    elif posisiAkhir == 88:
        posisiAkhir = 18
    elif posisiAkhir == 62:
        posisiAkhir = 22
    elif posisiAkhir == 57:
        posisiAkhir = 40
    elif posisiAkhir == 52:
        posisiAkhir = 29
    elif posisiAkhir == 17:
        posisiAkhir = 15

    #Tentukan value status dengan posisi akhir
    status = posisiAkhir

#Luaran status di posisi akhir
print(status)

```

### Hasil Pengujian:

```

Ular tangga x
"C:\Users\masad\Documents\UGM 2021\G
98 3
MENANG
Process finished with exit code 0

```

```

Ular tangga x
"C:\Users\masad\Documents\UGM 2021\G
74 6
MENANG
Process finished with exit code 0

```

```

Ular tangga x
"C:\Users\masad\Documents\UGM 2021\G
5 3
30
Process finished with exit code 0

```

```

Ular tangga x
"C:\Users\masad\Documents\UGM 2021\G
93 4
79
Process finished with exit code 0

```

```

Ular tangga x
"C:\Users\masad\Documents\UGM 2021\G
18 2
20
Process finished with exit code 0

```

## MENGGUNAKAN PERCABNGAN DENGAN SWITCH/Dictionary

Berbeda jika menggunakan switch/dictionary, implementasinya jauh lebih ramping, ringkas, mudah dibaca, dan ringan ketika dijalankan ketimbang menggunakan if dan else.

Adapun langkah algoritmanya sebagai berikut:

1. Mulai program
2. Inputkan posisi awal dan dadu yang dilempar (dan pastikan dalam tipe data integer).
3. Buat sebuah fungsi switch (contoh : switchPosisi()) dengan satu parameter bernama posisi. Dan berikan perintah didalamnya sebagai berikut:
  - a) Deklarasikan sebuah variabel **switcher** yang isinya adalah list asosiatif dengan keywordsnya adalah posisi-posisi dari dasar tangga ataupun posisi-posisi dari sebuah kepala ular. Sedangkan value dari masing-masing keywordsnya adalah posisi-posisi ujung atas tangga ataupun posisi-posisi ujung ekor ular.
  - b) Kemudian lakukan **return** pada fungsi tersebut value dari list **switcher** yang berkeyword sesuai dengan nilai posisi. Bila tidak ada keyword yang match dengan posisi, maka **returnkan** nilai posisi itu sendiri.
4. Selanjutnya adalah mendefinisikan variabel posisi akhir dengan nilainya adalah hasil pemanggilan fungsi switchPosisi() dengan argumen posisinya adalah hasil jumlah posisi awal dengan dadu. Sehingga fungsi yang dipanggil adalah switchPosisi(posisiAwal+dadu)
5. Kemudian check nilai apakah posisi akhir memenuhi lebih dari sama dengan 100. Jika benar, maka outputkan "MENANG". Jika tidak, maka outputkan nilai posisi akhir tersebut.
6. Akhiri program.

```
#Inputkan Posisi Awal dan Dadu
posisiAwal, dadu = input().split()

#ubah ke Integer
posisiAwal = int(posisiAwal)
dadu = int(dadu)

#Membuat fungsi switch
def switchPosisi(posisi):
    #membuat object atau list asosiatif sebagai dictionary
    switcher={
        3 : 21, #Kondisi Naik
        8 : 30,
        28 : 84,
        58 : 77,
        75 : 86,
        80 : 100,
        90 : 91,

        97 : 79, #Kondisi Turun
        95 : 51,
        88 : 18,
```

```

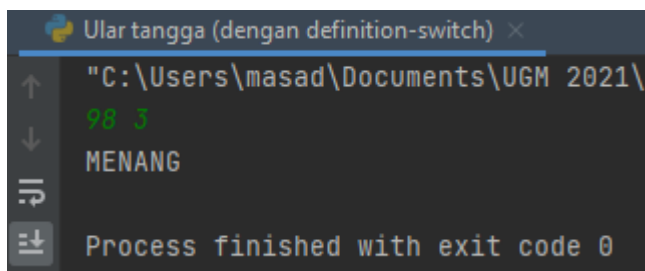
        62 : 22,
        57 : 57,
        52 : 29,
        17 : 15,
    }

    # Mengambil data sesuai keyword, bila tidak ada
    # keyword yg sesuai maka balikkan nilai posisi itu sendiri
    return switcher.get(posisi, posisi);

#Definisikan posisi akhir menggunakan fungsi yg telah dibuat
posisiAkhir = switchPosisi(posisiAwal+dadu);

#check apakah posisi akhir memenuhi lebih dari sama dengan 100 atau tidak
if posisiAkhir >= 100:
    print("MENANG") #Print "MENANG" bila memenuhi
else:
    print(posisiAkhir) #Print posisiAkhir bila kurang dari 100

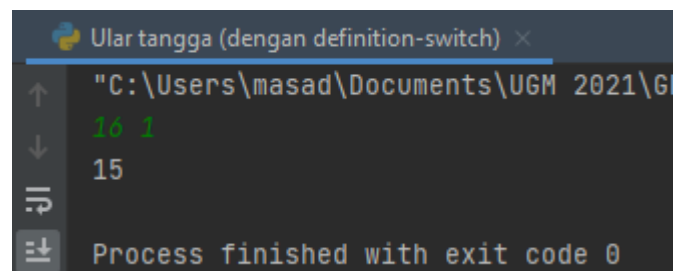
```



```

Ular tangga (dengan definition-switch) x
"C:\Users\masad\Documents\UGM 2021\G
98 3
MENANG
Process finished with exit code 0

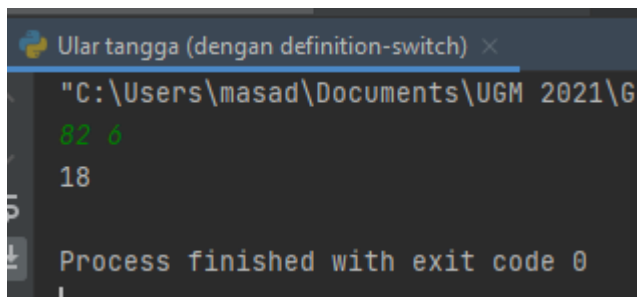
```



```

Ular tangga (dengan definition-switch) x
"C:\Users\masad\Documents\UGM 2021\G
16 1
15
Process finished with exit code 0

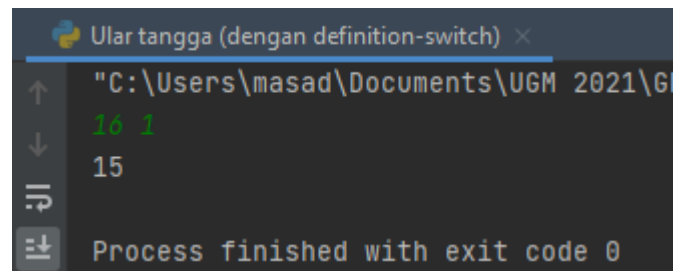
```



```

Ular tangga (dengan definition-switch) x
"C:\Users\masad\Documents\UGM 2021\G
82 6
18
Process finished with exit code 0

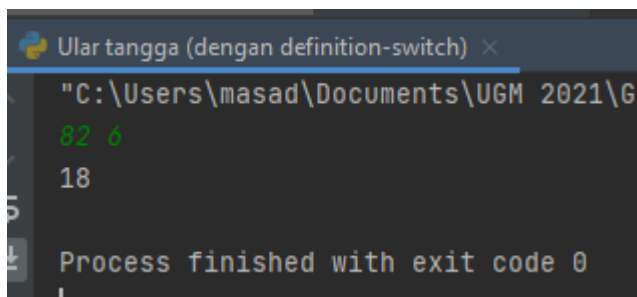
```



```

Ular tangga (dengan definition-switch) x
"C:\Users\masad\Documents\UGM 2021\G
16 1
15
Process finished with exit code 0

```



```

Ular tangga (dengan definition-switch) x
"C:\Users\masad\Documents\UGM 2021\G
82 6
18
Process finished with exit code 0

```