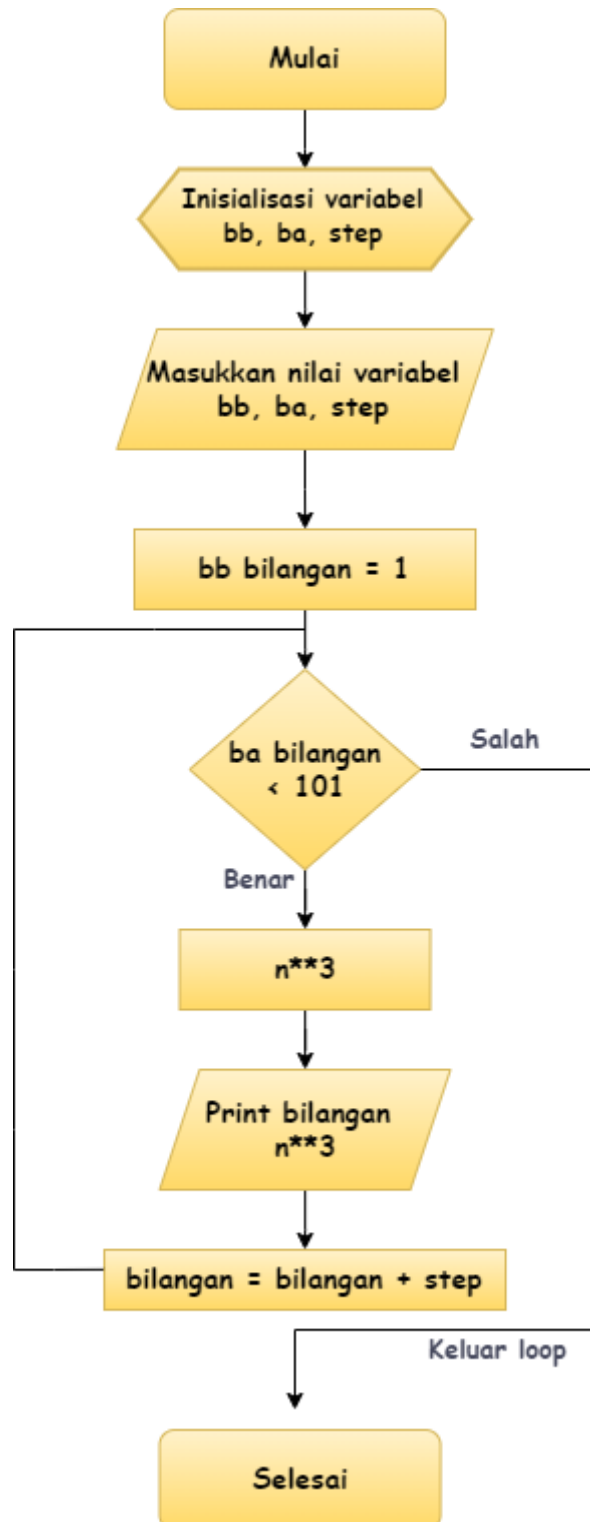


A. Struktur Perulangan Tunggal

1. Me-looping Bilangan Pangkat 3, mulai dari 1 s/d 100.

➤ Flowchart



➤ Kode Script Python

```
A. Struktur Perulangan Tunggal

1. Me-looping Bilangan Pangkat 3

In [2]: bb = int(input("Masukkan batas bawah : "))
        ba = int(input("Masukkan batas atas : "))
        step = int(input("Masukkan step : "))
        for n in range(bb,ba,step):
            n**3
            print("Hasil pangkat 3 dari", n, "=", n**3)

Masukkan batas bawah : 1
Masukkan batas atas : 101
Masukkan step : 1
Hasil pangkat 3 dari 1 = 1
Hasil pangkat 3 dari 2 = 8
Hasil pangkat 3 dari 3 = 27
Hasil pangkat 3 dari 4 = 64
Hasil pangkat 3 dari 5 = 125
Hasil pangkat 3 dari 6 = 216
Hasil pangkat 3 dari 7 = 343
Hasil pangkat 3 dari 8 = 512
Hasil pangkat 3 dari 9 = 729
Hasil pangkat 3 dari 10 = 1000
.. ..
```

Penjelasan Deskriptif :

✓ `bb = int(input("Masukkan batas bawah : "))`

Fungsi integer di sini di gunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di gunakan untuk memberikan nilai kepada variabel bb (batas bawah). Dengan variabel bb (batas bawah) di berikan nilai sebesar 1.

✓ `ba = int(input("Masukkan batas atas : "))`

Fungsi integer di sini di gunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di gunakan untuk memberikan nilai kepada variabel ba (batas atas). Dengan variabel ba (batas atas) di berikan nilai sebesar 101.

✓ `step = int(input("Masukkan step : "))`

Fungsi integer di sini di gunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di gunakan untuk memberikan nilai kepada variabel step. Dengan variabel step di berikan nilai sebesar 1.

✓ `for n in range(bb,ba,step):`

For n in range berfungsi di gunakan untuk n dalam jangkauan bb, ba, dan step. Dengan rincian bahwa n dalam jangkauan batas bawah 1,

batas atas 101, dan step 1. Maka, akan di tampilkan nilai atau hasil dari bilangan 1-100 setelah di pangkatkan 3 dengan step 1.

✓ `n**3`

`n**3` di sini di pergunakan untuk menuliskan rumus yang akan di gunakan untuk menghitung bilangan pangkat 3 dari 1-100 dengan step 1.

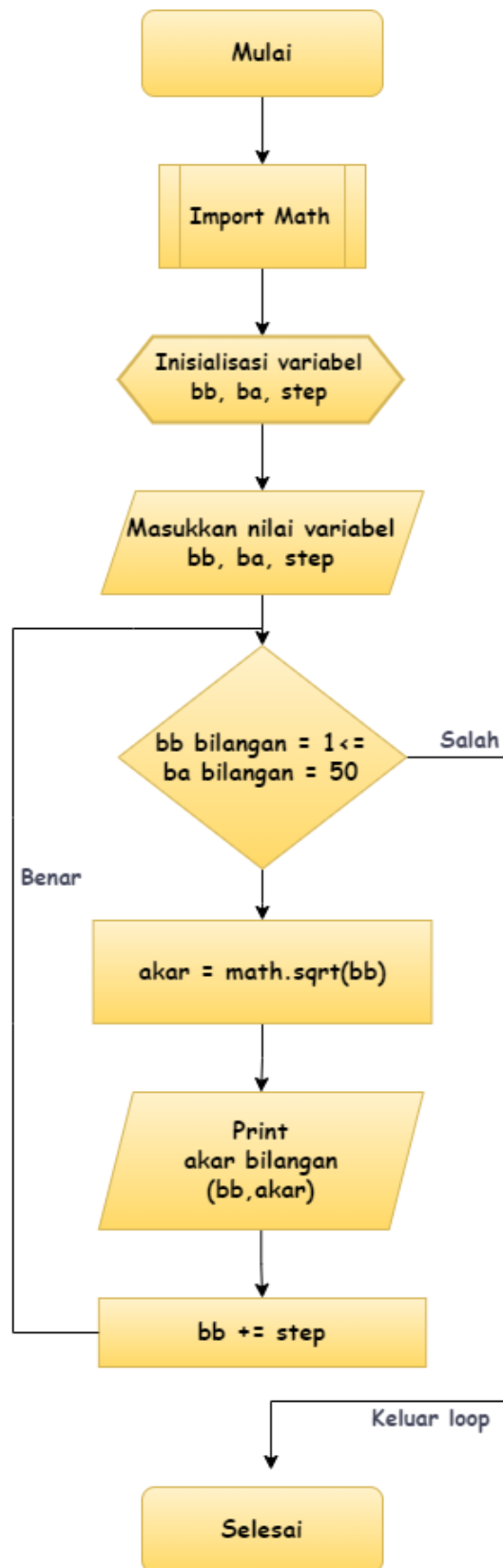
✓ `print("Hasil pangkat 3 dari", n, "=", n**3)`

Fungsi print di sini di pergunakan untuk menampilkan hasil dari bilangan dari 1-100 dengan step 1 setelah di pangkatkan dengan 3 sesuai dengan apa yang telah di tulis di rumus sel atasnya.

Dan di peroleh hasil, contohnya Hasil pangkat 3 dari 1 = 1.

2. Me-looping Akar Bilangan, mulai dari 1 s/d 50.

➤ Flowchart



➤ Kode Script Python

```
2. Me-looping Akar Bilangan

n [3]: import math
bb = int(input("Masukkan batas bawah : "))
ba = int(input("Masukkan batas atas : "))
step = int(input("Masukkan step : "))
while bb <= ba:
    akar=math.sqrt(bb)
    print("Hasil akar dari", bb, "=", akar)
    bb += step

Masukkan batas bawah : 1
Masukkan batas atas : 50
Masukkan step : 1
Hasil akar dari 1 = 1.0
Hasil akar dari 2 = 1.4142135623730951
Hasil akar dari 3 = 1.7320508075688772
Hasil akar dari 4 = 2.0
Hasil akar dari 5 = 2.23606797749979
Hasil akar dari 6 = 2.449489742783178
Hasil akar dari 7 = 2.6457513110645907
Hasil akar dari 8 = 2.8284271247461903
Hasil akar dari 9 = 3.0
Hasil akar dari 10 = 3.1622776601683795
Hasil akar dari 11 = 3.3166247903554
.....
```

Penjelasan Deskriptif :

✓ `import math`

`import math` di sini di pergunakan untuk menggunakan dan memperluas modul matematika di dalam python.

✓ `bb = int(input("Masukkan batas bawah : "))`

Fungsi integer di sini di pergunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di pergunakan untuk memberikan nilai kepada variabel `bb` (batas bawah). Dengan variabel `bb` (batas bawah) di berikan nilai sebesar 1.

✓ `ba = int(input("Masukkan batas atas : "))`

Fungsi integer di sini di pergunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di pergunakan untuk memberikan nilai kepada variabel `ba` (batas atas). Dengan variabel `ba` (batas atas) di berikan nilai sebesar 50.

✓ `step = int(input("Masukkan step : "))`

Fungsi integer di sini di pergunakan untuk menampilkan angka secara langsung tanpa adanya titik. Fungsi input di pergunakan untuk

memberikan nilai kepada variabel step. Dengan variabel step di berikan nilai sebesar 1.

✓ `while bb <= ba:`

While di pergunakan untuk proses perulangan suatu blok kode program selama sebuah kondisi terpenuhi. Dengan kondisi yang di maksudkan di sini adalah ketika bb (batas bawah) lebih kecil sama dengan ba (batas atas). Maka, pada kode script python di atas akan terjadi proses pengulangan hasil akar dari angka 1-50.

✓ `akar = math.sqrt(bb)`

Akar di pergunakan untuk menuliskan rumus yang akan di gunakan untuk menghitung bilangan akar dari 1-50 dengan step 1 dengan perluasan modul matematika di atas.

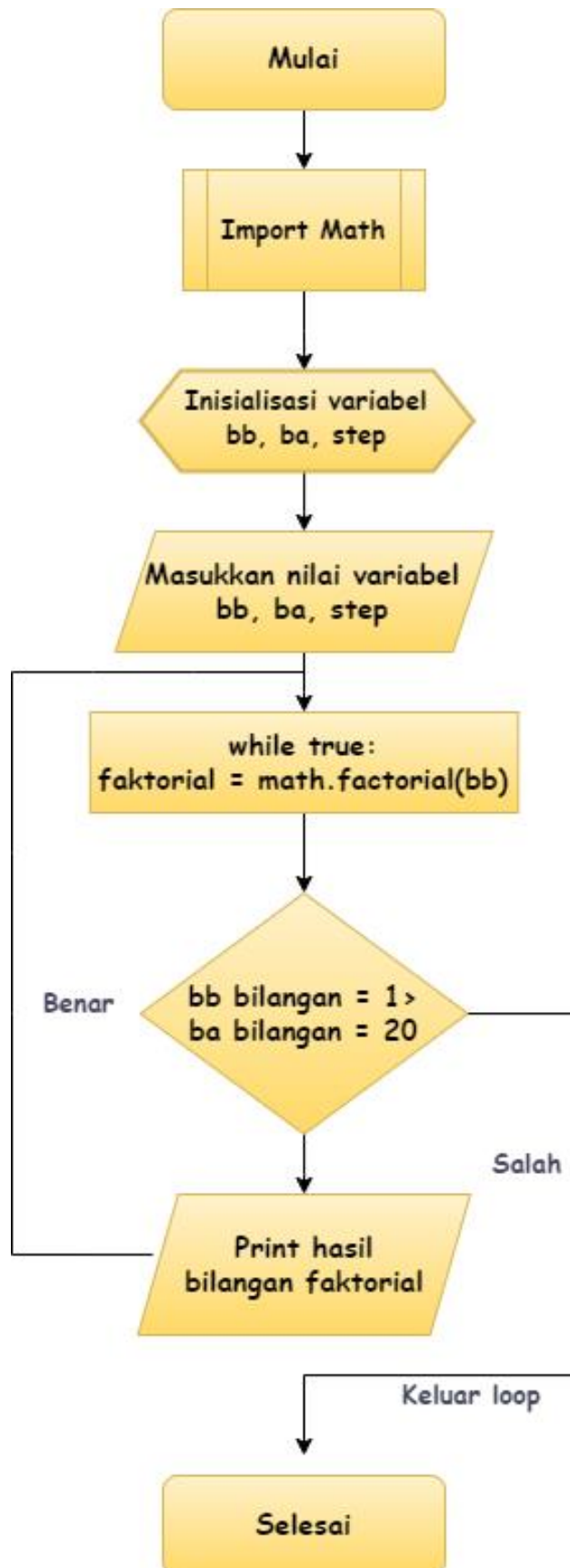
✓ `print("Hasil akar dari", bb, "=", akar)`

Print di pergunakan untuk menampilkan hasil dari akar bilangan 1-50 dengan step 1 sesuai yang telah di tuliskan pada sel di atasnya. Dan di peroleh hasil, contohnya Hasil akar dari 1 = 1.0.

✓ `bb += step`

Bb += step di sini di pergunakan untuk menuliskan pernyataan atau menuliskan hasil yang di peroleh dan di print, kemudian proses akan mengalami perulangan yakni bb += step jika kondisi tersebut masih bisa memenuhi pernyataan while bb <= ba maka proses akan terus di jalankan.

3. Me-looping Faktorial Bilangan, mulai dari 1-20
➤ Flowchart



➤ Kode Script Python

```
3. Me-looping Faktorial Bilangan

In [3]: import math
bb = int(input("Masukkan batas bawah : "))
ba = int(input("Masukkan batas atas : "))
step = int(input("Masukkan step : "))
while True :
    faktorial = math.factorial(bb)
    print("Hasil faktorial dari" f" {bb}! = {faktorial}")
    bb = bb + step
    if(bb > ba):
        break

Masukkan batas bawah : 1
Masukkan batas atas : 20
Masukkan step : 1
Hasil faktorial dari 1! = 1
Hasil faktorial dari 2! = 2
Hasil faktorial dari 3! = 6
Hasil faktorial dari 4! = 24
Hasil faktorial dari 5! = 120
Hasil faktorial dari 6! = 720
Hasil faktorial dari 7! = 5040
Hasil faktorial dari 8! = 40320
Hasil faktorial dari 9! = 362880
```

Penjelasan Deskriptif :

✓ `import math`

Import math di sini di digunakan untuk menggunakan dan memperluas modul matematika dalam python.

✓ `bb = int(input("Masukkan batas bawah :"))`

Integer di sini di digunakan untuk menampilkan angka tanpa adanya titik. Input di sini di digunakan untuk memberikan nilai kepada variabel bb (batas bawah). Dan di berikan nilai sebesar 1.

✓ `ba = int(input("Masukkan batas atas :"))`

Integer di sini di digunakan untuk menampilkan angka tanpa adanya titik. Input di sini di digunakan untuk memberikan nilai kepada variabel ba (batas atas). Dan di berikan nilai sebesar 20.

✓ `step = int(input("Masukkan step :"))`

Integer di sini di digunakan untuk menampilkan angka tanpa adanya titik. Input di sini di digunakan untuk memberikan nilai kepada variabel step. Dan di berikan nilai sebesar 1.

✓ `while True:`

While true berfungsi untuk melakukan perulangan dari suatu program selama masih sesuai dengan apa yang telah di tuliskan di sel atasnya.

✓ `faktorial = math.factorial(bb)`

Faktorial di sini di pergunakan untuk menuliskan rumus yang akan di pergunakan untuk menghitung hasil faktorial dari bilangan 1-20 dengan step 1 dengan menggunakan modul perluasan matematika di atas.

✓ `print("Hasil faktorial dari" f" {bb}!= {faktorial}")`

Print di sini di pergunakan untuk menampilkan hasil faktorial dengan menggunakan rumus faktorial sesuai apa yang telah di tuliskan pada sel atasnya.

✓ `bb = bb + step`

Bb = bb + step ini di pergunakan untuk menuliskan pernyataan. Dengan di maksudkan setelah bb (batas bawah) yang terakhir di eksekusi masih belum memenuhi ba (batas atas) maka prosess pengulangan masih akan terus berjalan.

✓ `if(bb > ba):`

If(bb > ba) di sini di pergunakan untuk menuliskan kondisi jika di peroleh bb (batas bawah) dari hasil di atas masih kurang dari ba (batas atas) maka perulangan masih akan terus di jalankan. Namun, jika di peroleh kondisi bb (batas bawah) lebih besar daripada ba (batas atas) maka perulangan akan di hentikan.

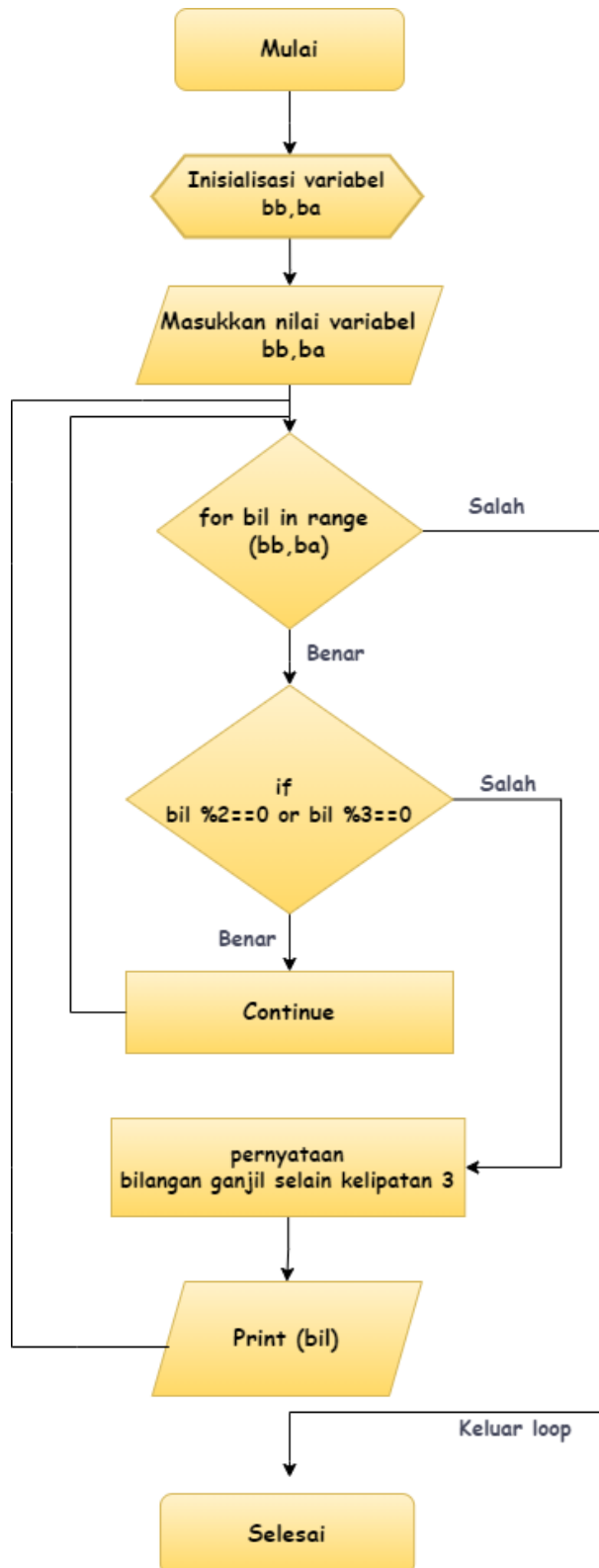
✓ `break`

Break di sini di pergunakan untuk memberhentikan suatu kondisi perulangan ketika di peroleh bb (batas bawah) lebih besar di bandingkan ba (batas atas). Dengan rincian bahwa ba dalam jangkauan bb,ba

B. Struktur Perulangan Bersarang

1. Menampilkan Bilangan Ganjil Selain Kelipatan 3

➤ Flowchart



➤ Kode Script Python

```
B. Struktur Perulangan Bersarang

1. Menampilkan Bilangan Ganjil Selain Kelipatan 3

In [1]: bb = int(input("Masukkan batas bawah : "))
        ba = int(input("Masukkan batas atas : "))
        for bil in range (bb,ba):
            if bil %2==0 or bil %3==0:
                continue
            print(bil)

Masukkan batas bawah : 1
Masukkan batas atas : 15
1
5
7
11
13
```

Penjelasan Deskriptif :

✓ `bb = int(input("Masukkan batas bawah : "))`

Integer di sini di pergunakan untuk menampilkan angka tanpa adanya titik. Input di sini di pergunakan untuk memberikan nilai kepada variabel bb (batas bawah). Dan di berikan nilai sebesar 15.

✓ `ba = int(input("Masukkan batas atas : "))`

Integer di sini di pergunakan untuk menampilkan angka tanpa adanya titik. Input di sini di pergunakan untuk memberikan nilai kepada variabel ba (batas atas). Dan di berikan nilai sebesar 15.

✓ `for bil in range (bb,ba)`

For di sini di pergunakan untuk bil (bilangan) dalam jangkauan bb,ba. Dengan rincian bahwa bil dalam jangkauan batas bawah 1 dan batas atas 15.

✓ `if bil %2==0 or bil %3==0:`

If di sini di pergunakan untuk memberikan syarat atau kondisi untuk bilangan yang akan di tampilkan. Dengan syarat jika bilangan akan habis di bagi di bagi dua dan habis di bagi 3. Maka, bilangan tersebut tidak akan di tampilkan.

✓ `continue`

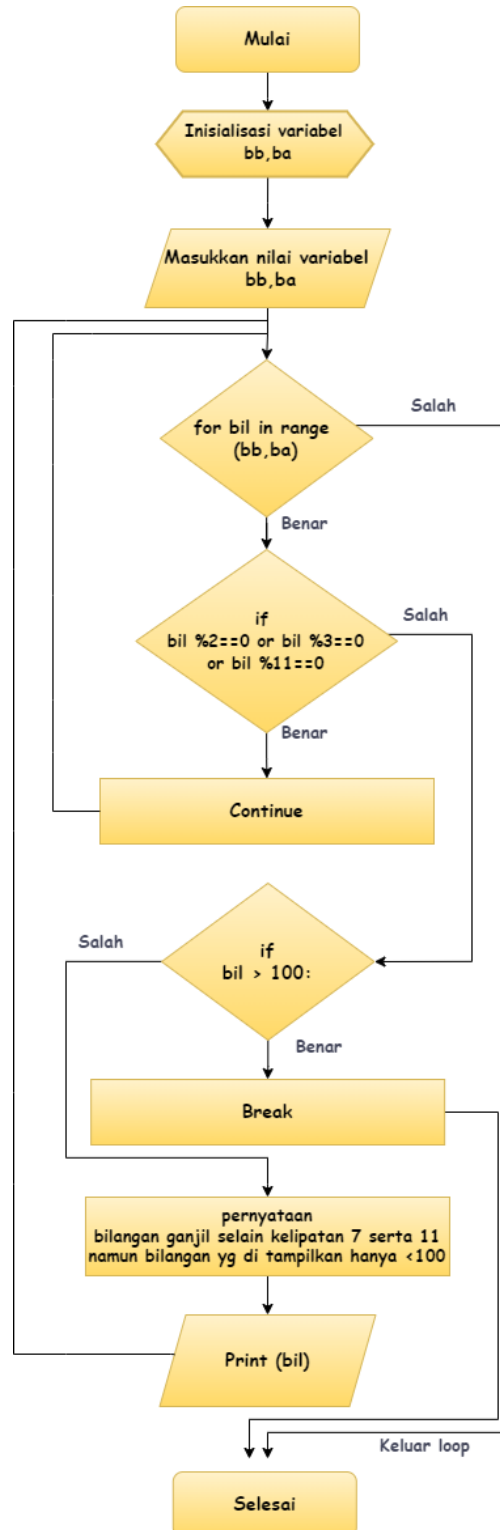
Continue di pergunakan untuk mengarahkan ke proses berikutnya dengan syarat jika kondisi if terpenuhi maka continue akan di jalankan dengan lompat ke iterasi selanjutnya.

✓ `print(bil)`

Print di pergunakan untuk menampilkan bilangan-bilangan yang telah di proses di sel atasnya, yakni akan menampilkan bilangan ganjil selain kelipatan 3.

2. Menampilkan Bilangan Ganjil Selain Kelipatan 7 serta 11, namun Bilangan yang Ditampilkan Hanya Dibatasi <100

➤ Flowchart



➤ Kode Script Python

```
2. Menampilkan Bilangan Ganjil selain Kelipatan 7 serta 11, namun Bilangan yang Ditampilkan Hanya Dibatasi <100

In [2]: bb = int(input("Masukkan batas bawah : "))
        ba = int(input("Masukkan batas atas : "))
        for bil in range (bb,ba):
            if bil %2==0 or bil %7==0 or bil %11==0:
                continue
            if bil > 100:
                break
            print(bil)

Masukkan batas bawah : 1
Masukkan batas atas : 250
1
3
5
9
13
15
17
19
23
25
--
```

Penjelasan Deskriptif :

✓ `bb = int(input("Masukkan batas bawah :"))`

Integer di sini di pergunakan untuk menampilkan angka tanpa adanya titik. Input di sini di pergunakan untuk memberikan nilai kepada variabel bb (batas bawah). Dan di berikan nilai sebesar 1.

✓ `ba = int(input("Masukkan batas atas :"))`

Integer di sini di pergunakan untuk menampilkan angka tanpa adanya titik. Input di sini di pergunakan untuk memberikan nilai kepada variabel ba (batas atas). Dan di berikan nilai sebesar 250.

✓ `for bil in range (bb,ba)`

For di pergunakan untuk bil (bilangan) dalam jangkauan bb, ba. Dengan rincian bahwa bil dalam jangkauan batas bawah 1 dan batas atas 250.

✓ `if bil %2==0 or bil %7==0 or bil %11==0:`

If di sini di pergunakan untuk memberikan syarat atau kondisi untuk bilangan yang akan di tampilkan. Dengan syarat jika bilangan akan habis di bagi 2, habis di bagi 7, dan habis di bagi 11. Maka, bilangan tersebut tidak akan di tampilkan.

✓ `continue`

Continue di pergunakan untuk mengarahkan ke proses berikutnya dengan syarat jika kondisi if terpenuhi maka continue akan di jalankan dengan lompat ke iterasi selanjutnya.

✓ `if bil > 100:`

If di pergunakan untuk memberikan syarat atau kondisi bahwa bilangan yang akan di tampilkan hanya akan kurang dari 100.

✓ `break`

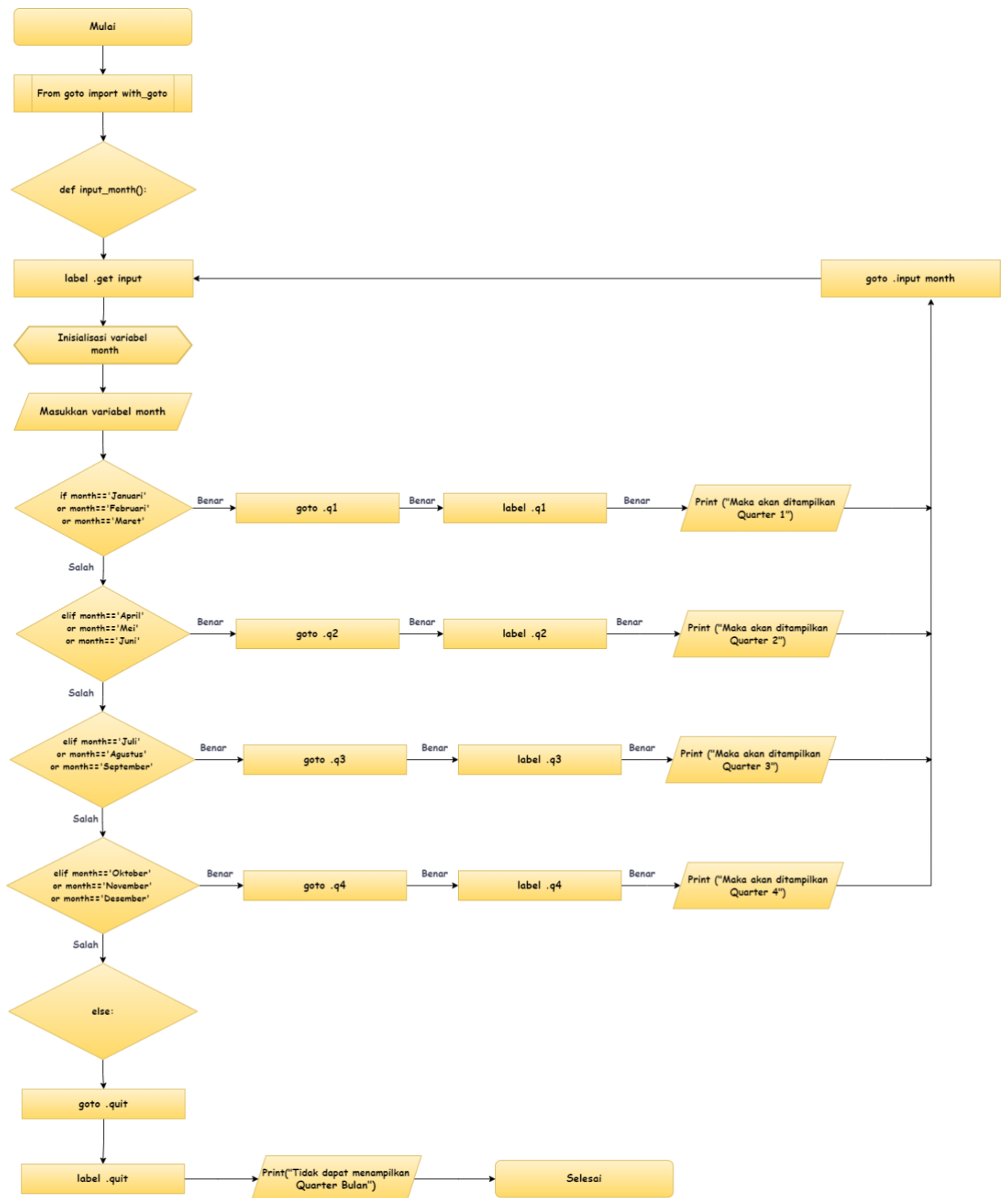
Break di pergunakan untuk jika kondisi dalam if kurang dari 100 telah terpenuhi. Maka, proses pengulangan akan di hentikan.

✓ `print(bil)`

Print di pergunakan untuk menampilkan bilangan-bilangan yang telah di proses di sel atasnya, yakni akan menampilkan bilangan ganjil selain kelipatan 7 dan 11, dan bilangan yang akan di tampilkan akan kurang dari 100.

3. Menampilkan Kuartar Bulan

➤ Flowchart



➤ Kode Script Python

3. Menampilkan Kuarter Bulan

```
In [*]: from goto import with_goto

@with_goto
def input_month():

    label .getinput
    month = str(input("Masukkan nama bulan : "))

    if month=='Januari' or month=='Februari' or month=='Maret':
        goto .q1
    elif month=='April' or month=='Mei' or month=='Juni':
        goto .q2
    elif month=='Juli' or month=='Agustus' or month=='September':
        goto .q3
    elif month=='Oktober' or month=='November' or month=='Desember':
        goto .q4
    else:
        goto .quit

    label .q1
    print("Maka akan di tampilkan Quarter 1")
    goto .getinput
```

```
    label .q2
    print("Maka akan di tampilkan Quarter 2")
    goto .getinput

    label .q3
    print("Maka akan di tampilkan Quarter 3")
    goto .getinput

    label .q4
    print("Maka akan di tampilkan Quarter 4")
    goto .getinput

    label .quit
    print("Tidak dapat menampilkan Quarter Bulan")

input_month()

Masukkan nama bulan : Mei
Maka akan di tampilkan Quarter 2
```

Penjelasan Deskriptif :

- ✓ `from goto import with_goto`

From goto import with_goto di pergunakan untuk menggunakan goto dalam python.

- ✓ `@with_goto`

Pada program ini akan menggunakan goto dalam python.

- ✓ `def input_month()`

Define function di pergunakan untuk mendefinisikan suatu fungsi. Dalam program ini di definisikan suatu fungsi dari input month (input bulan).

- ✓ label .get input

```
month = str(input("Masukkan nama bulan : "))
```

Label di pergunakan untuk menginputkan bulan yang di pilih dengan di tampung dulu.

String di gunakan untuk operasi input huruf. Input di pergunakan untuk memberikan inputan ke variabel month. Dan di berikan inputan nama bulan Mei di variabel month.

- ✓

```
if month=='Januari' or month=='Februari' or month=='Maret':  
    goto .q1
```

If di pegunakan untuk menentukan tindakan jika di inputkan nama bulan Januari, Februari, Maret pada variabel month maka proses akan di bawa ke label q1.

- ✓

```
elif month=='April' or month=='Mei' or month=='Juni':  
    goto .q2  
  
elif month=='Juli' or month=='Agustus' or month=='September':  
    goto .q3  
  
elif month=='Oktober' or month=='November' or month=='Desember':  
    goto .q4
```

Elif di pergunakan untuk menampilkan suatu kondisi yang sesuai dengan statement yang telah di inputkan pada bulan. Jika di inputkan bulan Mei maka proses akan di bawa ke label q2.

- ✓

```
else:  
    goto .quit
```

Else di pergunakan untuk pilihan terakhir jika nama bulan yang di inputkan pada input month tidak ada yang sesuai dengan kondisi if maupun elif. Maka proses akan di bawa ke label quit.

- ✓ label .q1

```
print("Maka akan di tampilkan Quarter 1")
```

goto .getinput

hingga seterusnya sampai :

label .quit

`print("Tidak dapat menampilkan Quarter Bulan")`

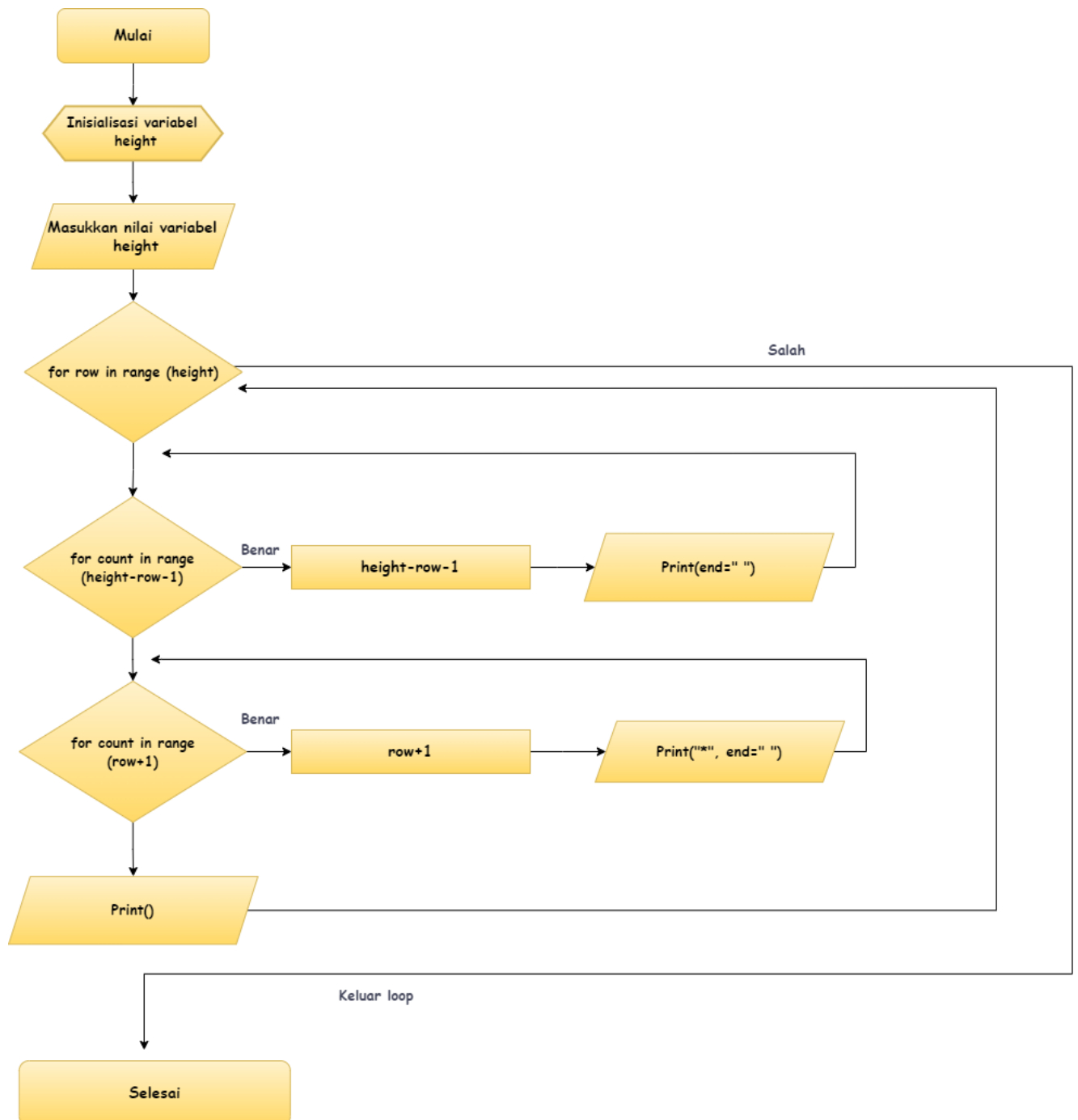
Label di sini di pergunakan untuk melanjutkan proses dari proses di atasnya. Jika, di inputkan bulan Mei maka proses akan di lanjutkan ke label .q2 dan akan di cetak Bulan Mei maka akan di tampilkan Quarter 2.

✓ `input_month()`

Input month di pergunakan untuk menginputkan bulan dan menampilkan hasil inputannya.

4. Menampilkan Tinggi Bintang

➤ Flowchart



➤ Kode Script Python

```
4. Menampilkan Tinggi Bintang

In [2]: height = eval(input("Masukkan tinggi bintang : "))
        for row in range(height):
            for count in range(height-row-1):
                print(end=" ")
            for count in range (row+1):
                print("*", end=" ")
            print()

Masukkan tinggi bintang : 5

*
 *
* *
* * *
* * * *
* * * * *
```

Penjelasan Deskriptif :

- ✓ `height = eval(input("Masukkan tinggi bintang : "))`
Eval di digunakan untuk menguraikan ekspresi dalam program. Input di digunakan untuk memberikan nilai kepada variabel height. Dan di berikan nilai kepada variabel height (tinggi bintang) sebesar 5.
- ✓ `for row in range(height):`
For row di digunakan untuk mengatur jumlah baris dalam jangkauan height (tinggi bintang).
- ✓ `for count in range(height-row-1):`
For count di sini di digunakan untuk menghitung jumlah kolom dan mengisi kolom menggunakan (*) dengan height-row-1.
- ✓ `print(end=" ")`
Print di sini di digunakan untuk menampilkan spasi jarak (*) hasil dari proses for count in range (height-row-1).
- ✓ `for count in range (row+1)`
For count di sini di digunakan untuk menghitung jumlah kolom dan mengisi kolom menggunakan (*) dengan row+1.
- ✓ `print("*", end=" ")`
Print di sini di digunakan untuk menampilkan (*) hasil dari proses di atas.
- ✓ `print()`

Print di sini di pergunakan untuk menampilkan bintang di tiap baris mulai 1 sampai 5 bintang sesuai dengan yang di tuliskan pada sel-sel di atasnya.