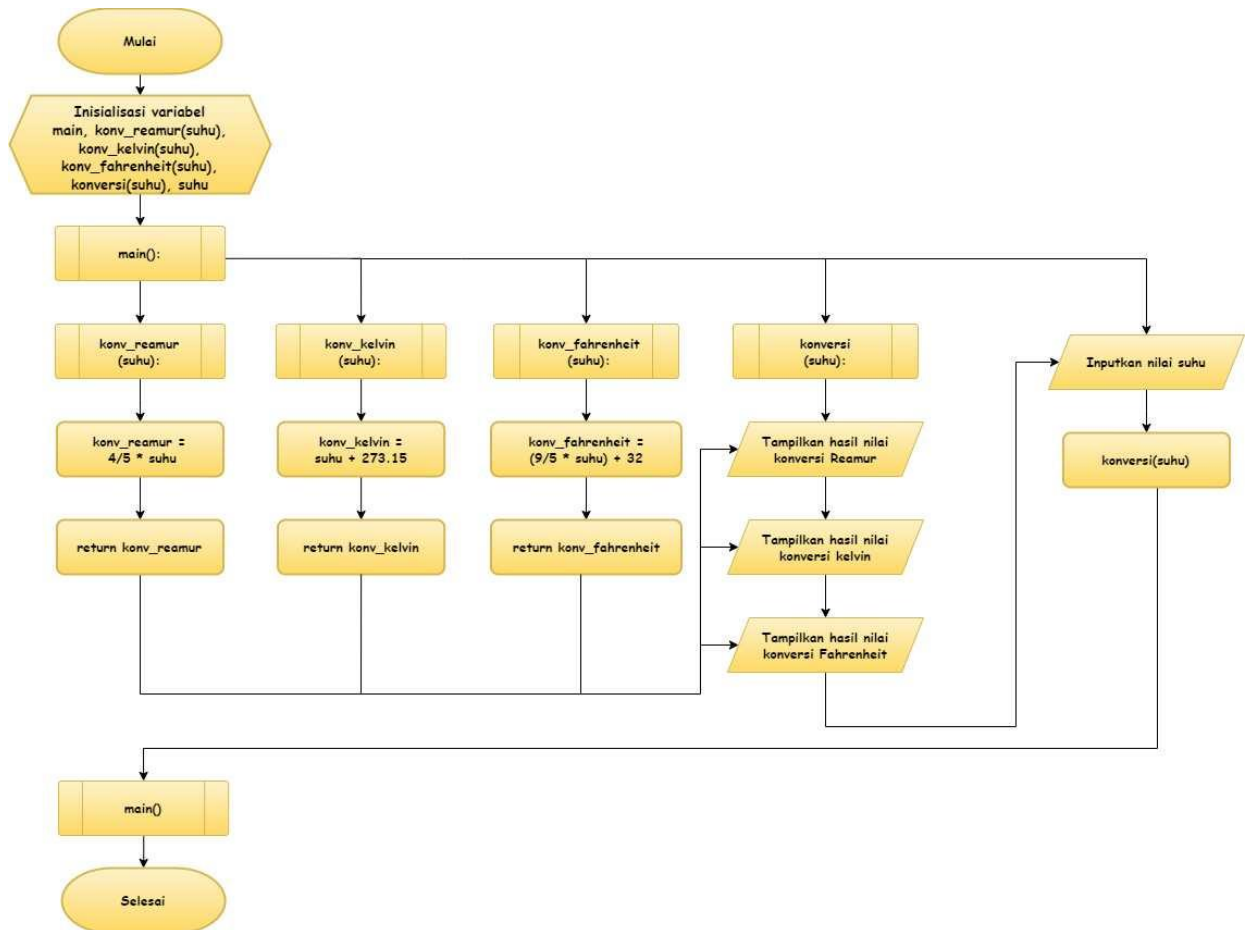


A. Python Function

1. Mengkonverensi Suhu Menggunakan Function

➤ Flowchart



➤ Kode Script Python

A. Python Function

1. Mengkonversi Suhu Menggunakan Function

```
In [2]: def main():  
  
    def konv_reamur(suhu):  
        konv_reamur = 4/5 * suhu  
        return konv_reamur  
  
    def konv_kelvin(suhu):  
        konv_kelvin = suhu + 273.15  
        return konv_kelvin  
  
    def konv_fahrenheit(suhu):  
        konv_fahrenheit = (9/5 * suhu) + 32  
        return konv_fahrenheit  
  
    def konversi(suhu):  
        print("Di peroleh hasil konversi suhu, dari ", suhu, "°C, ke Reamur sebesar : ", konv_reamur(suhu), "°R")  
        print("Di peroleh hasil konversi suhu, dari ", suhu, "°C, ke Kelvin sebesar : ", konv_kelvin(suhu), "°K")  
        print("Di peroleh hasil konversi suhu, dari ", suhu, "°C, ke Fahrenheit sebesar : ", konv_fahrenheit(suhu), "°F")  
  
        suhu = eval(input("Masukkan nilai suhu yang akan di konversi dalam satuan °C : "))  
        konversi(suhu)  
    main()
```

```
Masukkan nilai suhu yang akan di konversi dalam satuan °C : 90  
Di peroleh hasil konversi suhu, dari 90 °C, ke Reamur sebesar : 72.0 °R  
Di peroleh hasil konversi suhu, dari 90 °C, ke Kelvin sebesar : 363.15 °K  
Di peroleh hasil konversi suhu, dari 90 °C, ke Fahrenheit sebesar : 194.0 °F
```

Penjelasan Deskriptif :

➤ **def main():**

Define function di pergunakan untuk mendefinisikan dan menggabungkan suatu fungsi mengkonversi suhu menggunakan function yang di dalam program ini terdapat beberapa define function lain.

➤ **def konv_reamur(suhu):**

Def konv_reamur di pergunakan untuk menyatakan suatu program mengkonversi suhu ke dalam satuan reamur, parameter formal suhu.

➤ **konv_reamur = 4/5 * suhu**

Di pergunakan sebagai variabel untuk menuliskan rumus mengubah suhu ke dalam satuan reamur. Dengan rumus, 4/5 di kali dengan nilai suhu yang telah di inputkan di variabel suhu.

➤ **return konv_reamur**

Return di pergunakan untuk menampilkan hasil nilai akhir suhu reamur.

➤ **def konv_kelvin(suhu):**

Def konv_kelvin di pergunakan untuk menyatakan suatu program mengkonversi suhu ke dalam satuan kelvin, parameter formal suhu.

➤ **konv_kelvin = suhu + 273.15**

Di pergunakan sebagai variabel untuk menuliskan rumus untuk mengubah suhu ke dalam satuan kelvin. Dengan rumus, nilai suhu yang telah di inputkan di variabel suhu di tambah dengan 273.15.

➤ `return konv_kelvin`

Return di pergunakan untuk menampilkan hasil nilai akhir suhu kelvin

➤ `def konv_fahrenheit(suhu)`

Def konv_fahrenheit di pergunakan untuk menyatakan suatu program mengkonversi suhu ke dalam satuan fahrenheit , parameter formal suhu.

➤ `konv_fahrenheit = (9/5 * suhu) + 32`

Di pergunakan sebagai variabel untuk menuliskan rumus mengubah suhu ke dalam satuan fahrenheit. Dengan rumus, 9/5 di kali nilai suhu yang telah di inputkan di variabel suhu di tambah dengan 32.

➤ `return konv_fahrenheit`

Return di sini di pergunakan untuk menampilkan hasil nilai akhir suhu fahrenheit.

➤ `def konversi(suhu):`

Def konversi di pergunakan untuk menyatakan dan menampilkan hasil nilai dari suhu yang telah di konversi ke dalam satuan reamur, kelvin, dan fahrenheit.

➤ `print("Di peroleh hasil konversi suhu, dari", suhu, "°C", ke Reamur sebesar : " konv_reamur(suhu), "°R")`

Pada program ini di tampilkan di peroleh nilai suhu celcius yang telah di ubah ke dalam satuan reamur sebesar 72,0°R.

➤ `print("Di peroleh hasil konversi suhu, dari", suhu, "°C, ke Kelvin sebesar : ", konv_kelvin(suhu), "°K")`

Pada program ini di tampilkan di peroleh nilai suhu celcius yang telah di ubah ke dalam satuan kelvin sebesar 363.15°K.

➤ `print("Di peroleh hasil konversi suhu, dari", suhu, "°C, ke Fahrenheit sebesar : ", konv_fahrenheit(suhu) "°F")`

Pada program ini di tampilkan di peroleh nilai suhu celcius yang telah di ubah ke dalam satuan fahrenheit sebesar 194,0°F.

➤ `suhu = eval(input("Masukkan nilai suhu yang akan di konversi dalam satuan °C : "))`

Suhu di sini di pergunakan untuk menginputkan besarnya nilai suhu yang akan di konversi dalam satuan celcius dengan tipe eval (bilangan float maupun bulat). Pada program ini di inputkan nilai suhu yang akan di konversi sebesar 90°C.

➤ konversi(suhu)

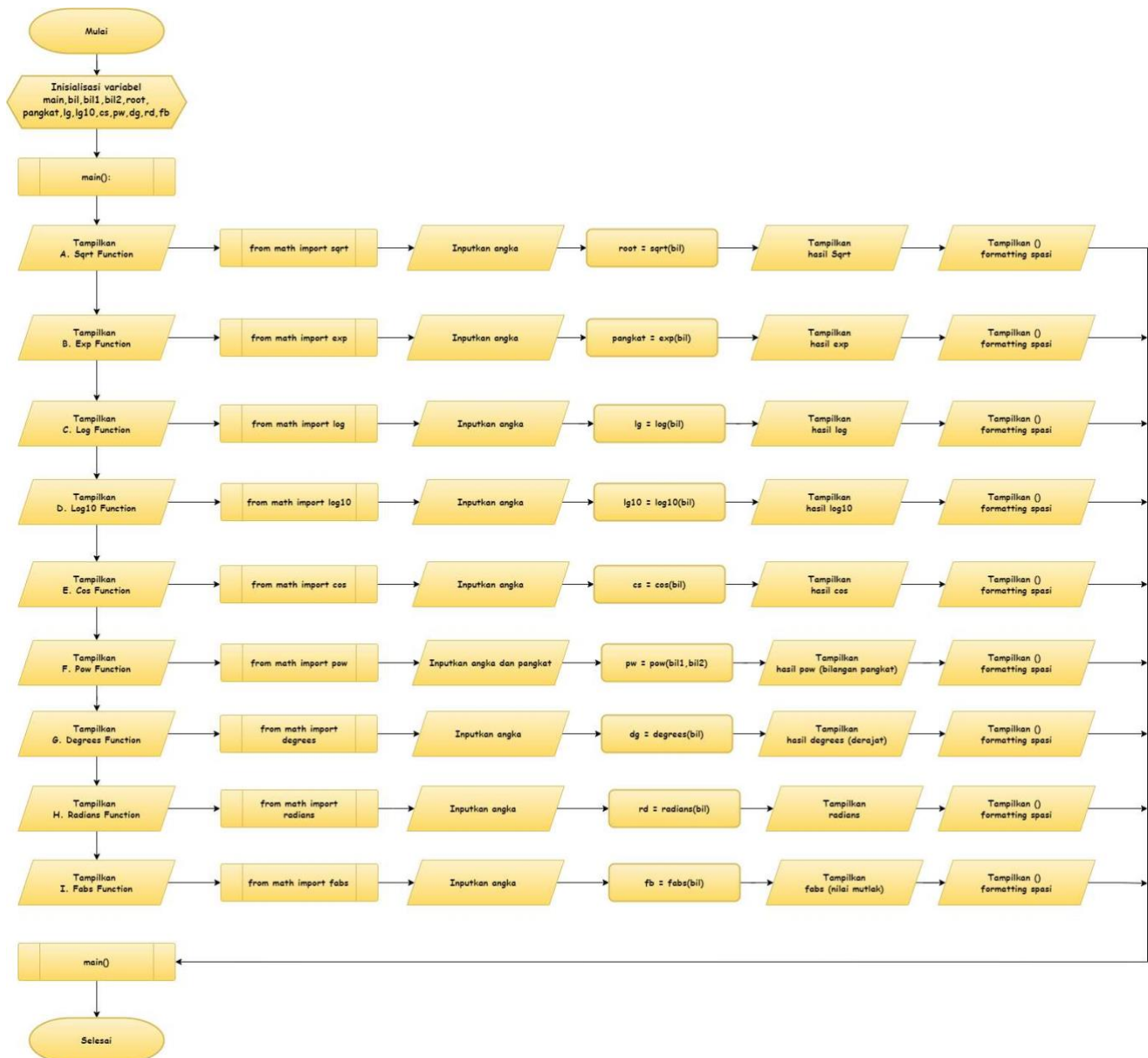
Konversi suhu di sini di pergunakan untuk memanggil hasil dari bagian deg konversi(suhu).

➤ main()

Main di sini di pergunakan untuk pemanggilan seluruh define function sesuai apa yang telah di tuliskan di atas.

2. Menggunakan Seluruh Function Pada Module Math

➤ Flowchart



➤ Kode Script Python

2. Menggunakan Seluruh Function Pada Module Math

```
In [1]: def main():  
  
    print("A. Sqrt Function")  
    from math import sqrt  
    bil = eval(input("Masukkan angka : "))  
    root = sqrt(bil)  
    print("Di peroleh hasil akar dari bilangan", bil, "sebesar : ", root)  
    print()  
  
    print("B. Exp Function")  
    from math import exp  
    bil = eval(input("Masukkan angka : "))  
    pangkat = exp(bil)  
    print("Di peroleh hasil pengembalian yang di pangkatkan dari bilangan", bil, "sebesar : ", pangkat)  
    print()  
  
    print("C. Log Function")  
    from math import log  
    bil = eval(input("Masukkan angka : "))  
    lg = log(bil)  
    print("Di peroleh hasil log dari bilangan", bil, "sebesar : ", lg)  
    print()  
  
    print("D. Log10 Function")  
    from math import log10
```

```
    print("D. Log10 Function")  
    from math import log10  
    bil = eval(input("Masukkan angka : "))  
    lg10 = log10(bil)  
    print("Di peroleh hasil log10 dari bilangan", bil, "sebesar : ", lg10)  
    print()  
  
    print("E. Cos Function")  
    from math import cos  
    bil = eval(input("Masukkan angka : "))  
    cs = cos(bil)  
    print("Di peroleh hasil cos dari bilangan", bil, "sebesar : ", cs)  
    print()  
  
    print("F. Pow Function")  
    from math import pow  
    bil1 = eval(input("Masukkan angka : "))  
    bil2 = eval(input("Masukkan pangkat : "))  
    pw = pow(bil1, bil2)  
    print("Di peroleh hasil dari bilangan", bil1, "pangkat", bil2, "sebesar : ", pw)  
    print()  
  
    print("G. Degrees Function")  
    from math import degrees  
    bil = eval(input("Masukkan angka : "))  
    dg = degrees(bil)  
    print("Di peroleh hasil konverensi", bil, "radian, sebesar : ", dg, "derajat")  
    print()
```

```
    print("H. Radians Function")  
    from math import radians  
    bil = eval(input("Masukkan angka : "))  
    rd = radians(bil)  
    print("Di peroleh hasil konverensi", bil, "derajat, sebesar : ", rd, "radian")  
    print()  
  
    print("I. Fabs Function")  
    from math import fabs  
    bil = eval(input("Masukkan angka : "))  
    fb = fabs(bil)  
    print("Di peroleh hasil nilai mutlak dari bilangan", bil, "sebesar : ", fb)  
    print()  
  
main()
```

```
A. Sqrt Function  
Masukkan angka : 100  
Di peroleh hasil akar dari bilangan 100 sebesar : 10.0  
  
B. Exp Function  
Masukkan angka : 65  
Di peroleh hasil pengembalian yang di pangkatkan dari bilangan 65 sebesar : 1.6948892444103338e+28  
  
C. Log Function  
Masukkan angka : 14  
Di peroleh hasil log dari bilangan 14 sebesar : 2.6390573296152584
```

```

D. Log10 Function
Masukkan angka : 86
Di peroleh hasil log10 dari bilangan 86 sebesar : 1.9344984512435677

E. Cos Function
Masukkan angka : 0
Di peroleh hasil cos dari bilangan 0 sebesar : 1.0

F. Pow Function
Masukkan angka : 3
Masukkan pangkat : 5
Di peroleh hasil dari bilangan 3 pangkat 5 sebesar : 243.0

G. Degrees Function
Masukkan angka : 50
Di peroleh hasil konverensi 50 radian, sebesar : 2864.7889756541163 derajat

H. Radians Function
Masukkan angka : 90
Di peroleh hasil konverensi 90 derajat, sebesar : 1.5707963267948966 radian

I. Fabs Function
Masukkan angka : -99
Di peroleh hasil nilai mutlak dari bilangan -99 sebesar : 99.0

```

Penjelasan Deskriptif :

- `def main():`
Define function di pergunakan untuk mendefinisikan suatu fungsi. Pada program ini def main di pergunakan untuk mendefinisikan suatu fungsi penggunaan seluruh function pada module math.
- `from math import sqrt`
From math di pergunakan untuk pemanggilan modul math dan mengimport sqrt untuk mencari akar kuadrat.
- `bil = eval(input("Masukkan angka : "))`
Bil di pergunakan sebagai variabel untuk menginputkan angka dengan tipe eval (bilangan float ataupun bulat).
- `root = sqrt(bil)`
Root di pergunakan sebagai variabel untuk mencari nilai akar kuadrat.
Pada program ini di tampilkan di inputkan angka sebesar 100 dan di peroleh hasil sebesar 10,0.
- `print()`
Print di pergunakan untuk menampilkan formatting print spasi.
- `from math import exp`
From math di pergunakan untuk pemanggilan module math dan mengimport exp untuk menampilkan bilangan eksponensial dari angka.
- `pangkat = exp(bil)`
Pangkat di pergunakan sebagai variabel untuk mencari bilangan eksponensial atau hasil pengembalian dari angka.

Pada program ini di tampilkan di inputkan angka sebesar 65 dan di peroleh hasil sebesar 1.694889244410338e+28.

➤ `from math import log`

From math di pergunakan untuk pemanggilan module math dan mengimport log untuk pengembalian logaritma natural dari angka.

➤ `lg = log(bil)`

Lg di pergunakan sebagai variabel untuk mencari nilai log dari suatu angka. Pada program ini di tampilkan di inputkan angka sebesar 14 dan di peroleh hasil sebesar 2.63905732962584.

➤ `from math import log10`

From math di sini di pergunakan untuk pemanggilan module math dan mengimport log10 untuk pengembalian basis 10 dari logaritma angka.

➤ `lg10 = log10(bil)`

Lg10 di pergunakan sebagai variabel untuk mencari nilai log10 dari suatu angka.

Pada program ini di tampilkan di inputkan angka sebesar 86 dan di peroleh hasil sebesar 1.9344984512435677.

➤ `from math import cos`

From math di pergunakan untuk pemanggilan module math dan mengimport cos untuk menghitung nilai cosinus dari angka.

➤ `cs = cos(bil)`

Cs di pergunakan sebagai variabel untuk mencari nilai cosinus dari suatu angka.

Pada program ini di tampilkan di inputkan angka sebesar 0 dan di peroleh hasil sebesar 1.0.

➤ `from math import pow`

From math di pergunakan untuk pemanggilan module math dan mengimport pow untuk mendapatkan nilai pangkat dari angka.

➤ `pw = pow(bil1, bil2)`

Pw di sini di pergunakan sebagai variabel untuk mencari nilai pow dari suatu angka.

Pada program ini di tampilkan di inputkan angka sebesar 3 dan pangkat sebesar 5 di peroleh hasil sebesar 343.0.

➤ `from math import degrees`

From math di sini di pergunakan untuk pemanggilan module math dan mengimport degrees untuk mengkonversi sudut dari radian ke derajat.

➤ `dg = degrees(bil)`

Dg di sini di pergunakan sebagai variabel untuk mencari nilai degrees (derajat) dari suatu angka.

Pada program ini di tampilkan di inputkan angka sebesar 50 dan di peroleh hasil sebesar 2864.788975641163.

➤ `from math import radians`

From math di pergunakan untuk pemanggilan module math dan mengimport radians untuk mengkonversi sudut dari derajat ke radians.

➤ `rd = radians(bil)`

Rd di pergunakan sebagai variabel untuk mencari radians.

Pada program ini di tampilkan di inputkan angka sebesar 90 dan di peroleh hasil sebesar 1.5707963267948966.

➤ `from math import fabs`

From math di pergunakan untuk pemanggilan module math dan mengimport fabs untuk mencari nilai mutlak.

➤ `fb = fabs(bil)`

Fb di pergunakan untuk mencari nilai mutlak dari suatu angka.

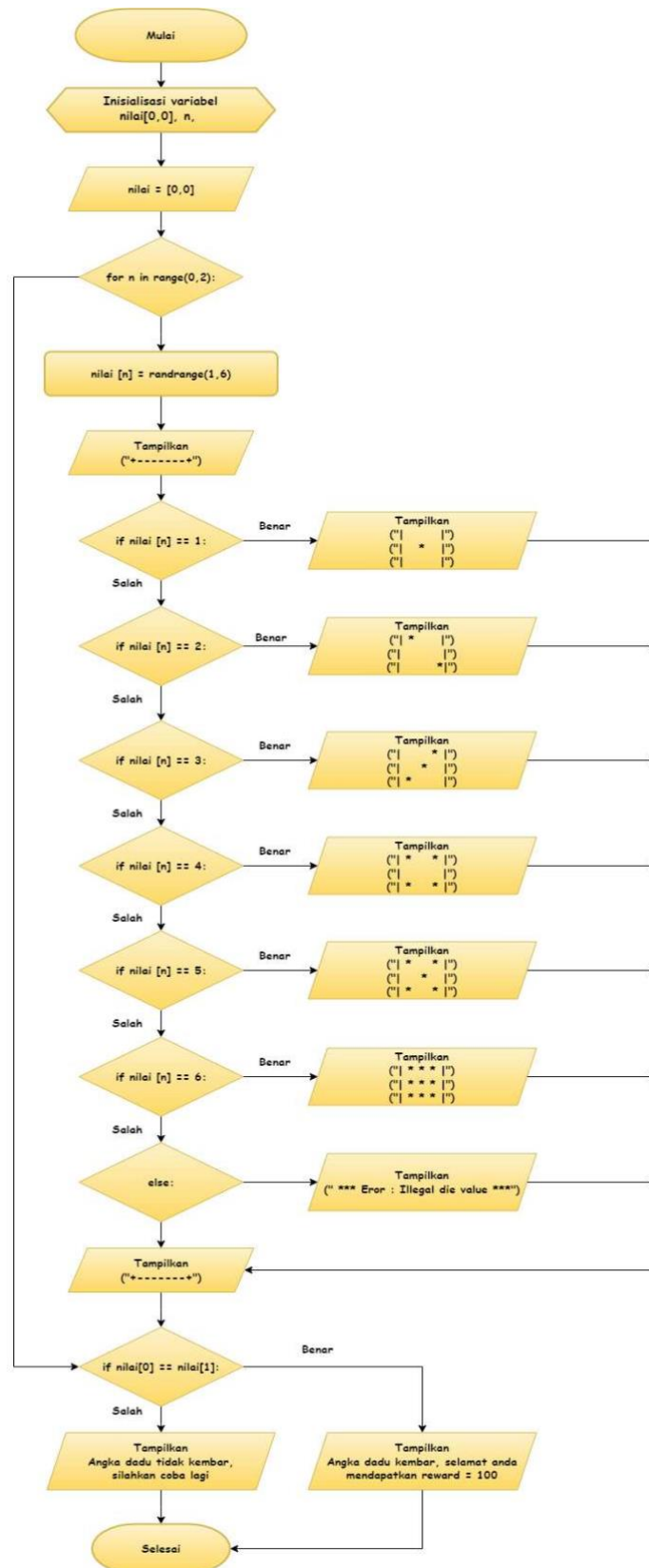
Pada program ini di tampilkan di inputkan angka sebesar -99 dan di peroleh hasil sebesar 99.0.

➤ `main()`

Main di pergunakan untuk pemanggilan fungsi sesuai apa yang telah di tuliskan.

3. Memberikan Reward Jika 2 Angka Dadu Kembar

➤ Flowchart



➤ Kode Script Python

```
3. Memberikan Reward Jika 2 Angka Dadu Kembar

n [15]: from random import randrange

nilai = [0,0]
for n in range(0,2):
    nilai [n] = randrange(1,6)
    print("+-----+")

    if nilai [n] == 1:
        print(" * * ")
        print(" * * ")
        print(" * * ")
    elif nilai [n] == 2:
        print(" * * ")
        print(" * * ")
        print(" * * ")
    elif nilai [n] == 3:
        print(" * * ")
        print(" * * ")
        print(" * * ")
    elif nilai [n] == 4:
        print(" * * ")
        print(" * * ")
        print(" * * ")
```

```
        print(" * * ")
        print(" * * ")
        print(" * * ")
    elif nilai [n] == 6:
        print(" * * ")
        print(" * * ")
        print(" * * ")
    else:
        print(" *** Error : Illegal die Value ***")
        print("+-----+")

if nilai[0] == nilai[1]:
    print("Angka dadu kembar, selamat anda mendapatkan reward = 100")
else:
    print("Anda dadu tidak kembar, silahkan coba lagi")

+-----+
| *      |
+-----+
+-----+
| *      |
+-----+
+-----+

Angka dadu kembar: selamat anda mendapatkan reward = 100
```

Penjelasan Deskriptif :

➤ `from random import randrange`

From random di pergunakan untuk penggunaan module random dan mengimport randrange untuk memilih secara acak.

➤ `nilai = [0,0]`

Nilai di pergunakan sebagai variabel untuk menampung dadu.

➤ `for n in range(0,2):`

For n di pergunakan untuk melakukan loop for untuk menampilkan dadu sebanyak 2 kali.

➤ `nilai [n] = randrange(1,6)`

Nilai [n] di pergunakan untuk mengatur angka dadu pertama dan kedua dalam rentang 1-6 menggunakan randrange.

➤ `print("+-----+")`

Print di pergunakan untuk menampilkan formatting print tersebut.

➤ `if nilai [n] == 1:`

```
print("|  |")
```

```
print("| * |")
```

```
print("|  |")
```

`elif nilai [n] == 6:`

If elif di pergunakan untuk menampilkan hasil dadu sesuai apa

yang telah di peroleh dari hasil randrange untuk dadu pertama dan kedua.

➤ `else :`

```
print("*** Error : Illegal die value ***")
```

Jika kondisi if elif tidak terpenuhi akan di tampilkan kondisi tersebut.

➤ `if nilai [0] = nilai [1]`

If di pergunakan untuk memberikan kondisi jika dadu pertama dan kedua sama.

➤ `print("Angka dadu kembar, selamat anda mendapatkan reward = 100")`

Maka, akan di tampilkan Angka dadu kembar, selamat anda mendapatkan reward 100.

➤ `else:`

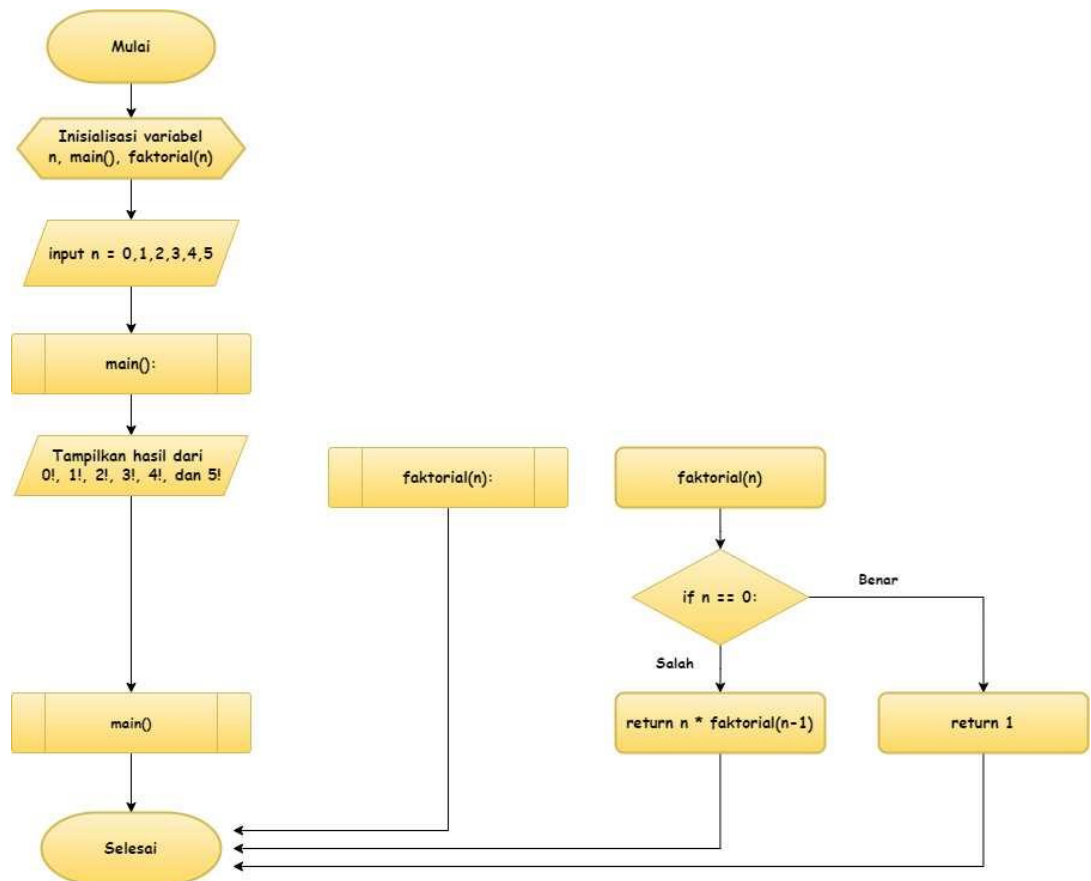
```
print("Angka dadu tidak kembar, silahkan coba lagi")
```

Else di pergunakan untuk memberikan kondisi jika if tidak terpenuhi. Akan di tampilkan tulisan angka dadu tidak kembar, silahkan coba lagi

B. Python Recursion (Rekursi)

1. Bilangan Faktorial

➤ Flowchart



➤ Kode Script Python

B. Python Recursion (Rekursi)

1. Bilangan Faktorial

```
In [5]: def faktorial(n):
        if n == 0:
            return 1
        else:
            return n * faktorial(n-1)

        def main():
            print("Di peroleh hasil 0! : ", faktorial(0))
            print("Di peroleh hasil 1! : ", faktorial(1))
            print("Di peroleh hasil 2! : ", faktorial(2))
            print("Di peroleh hasil 3! : ", faktorial(3))
            print("Di peroleh hasil 4! : ", faktorial(4))
            print("Di peroleh hasil 5! : ", faktorial(5))

        main()

Di peroleh hasil 0! : 1
Di peroleh hasil 1! : 1
Di peroleh hasil 2! : 2
Di peroleh hasil 3! : 6
Di peroleh hasil 4! : 24
Di peroleh hasil 5! : 120
```

Penjelasan Deskriptif :

➤ `def faktorial(n):`

Define function di pergunakan untuk menyatakan atau mendefinisikan suatu fungsi. Pada program ini di definisikan suatu define function faktorial dengan parameter formal n.

➤ `if n == 0:`

If di pergunakan untuk memberikan kondisi jika $n = 0$.

➤ `return 1`

Maka, akan di tampilkan hasil faktorial sebesar 1.

➤ `else:`

Else di pergunakan untuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `return n * faktorial(n-1)`

Maka, akan di lakukan perhitungan faktorial dengan n di kali faktorial n-1.

➤ `def main():`

Pada program ini def main di pergunakan untuk menampilkan hasil faktorial.

➤ `print("Di peroleh hasil 0! : ", faktorial(0)) ...`

Print di pergunakan untuk menampilkan hasil dari bilangan faktorial. Dengan proses pada 0 faktorial keadaan ini memenuhi kondisi if, yakni $n == 0$, maka akan di tampilkan hasil 1.

Dan pada 1-5 keadaan ini memenuhi kondisi else dan untuk mencari hasil akan di proses menggunakan $n * \text{faktorial}(n-1)$.

Sebagai contoh pada 1!, akan di proses dengan n sebesar 1 dikali faktorial 1-1, yakni 1 di kali satu, maka di dapatkan hasil sebesar 1.

Hal tersebut akan di ulangi hingga hasil dari 5!.

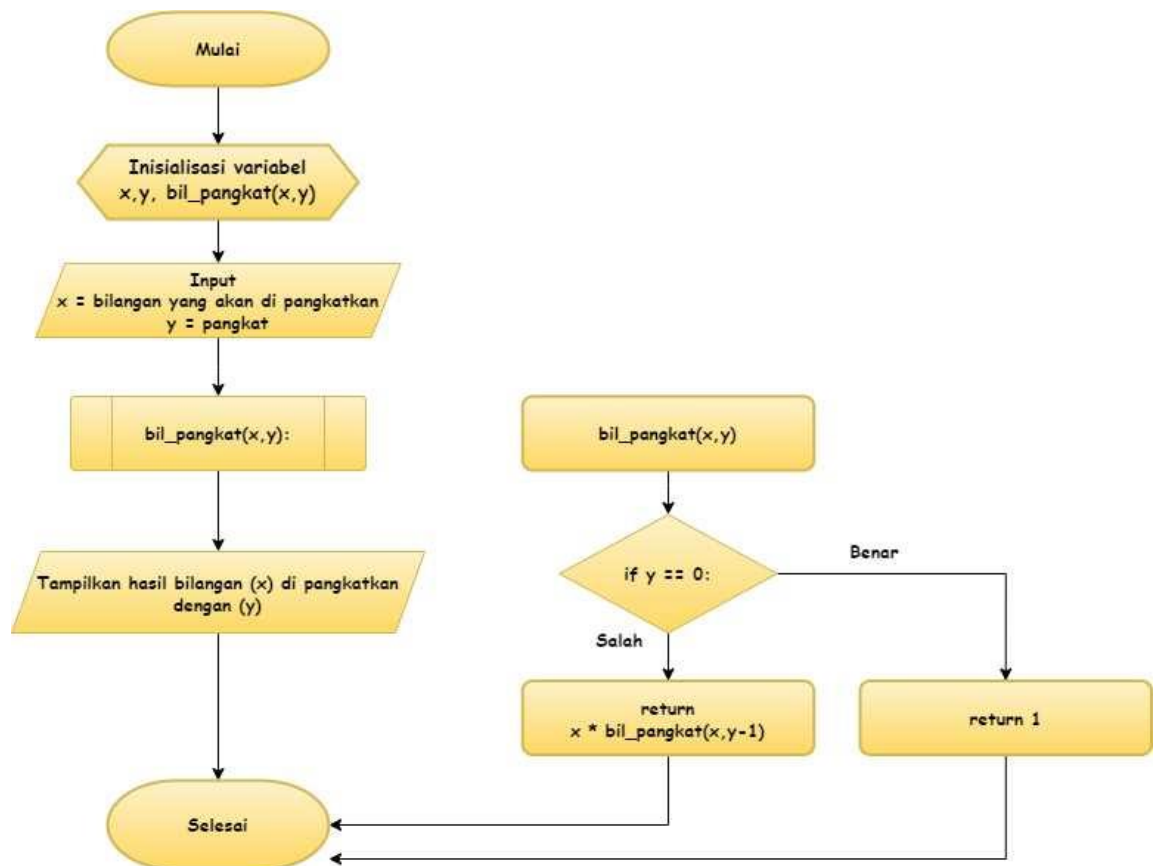
Di peroleh hasil $0! = 1$, $1! = 1$, $2! = 2$, $3! = 6$, $4! = 24$, dan $5! = 120$.

➤ `main()`

Main di pergunakan untuk pemanggilan fungsi sesuai apa yang telah di tuliskan di atas.

2. Bilangan Berpangkat

➤ Flowchart



➤ Kode Script Python

```
2. Bilangan Berpangkat

In [9]: def bil_pangkat(x,y):
        if y == 0:
            return 1
        else:
            return x * bil_pangkat(x,y-1)

        x = int(input("Masukkan bilangan yang akan di pangkatkan : "))
        y = int(input("Masukkan pangkat : "))

        print("Di peroleh hasil bilangan berpangkat dari", x, "di pangkatkan dengan ", y, "sebesar : ", bil_pangkat(x,y))

Masukkan bilangan yang akan di pangkatkan : 3
Masukkan pangkat : 5
Di peroleh hasil bilangan berpangkat dari 3 di pangkatkan dengan 5 sebesar : 243
```

Penjelasan Deskriptif :

➤ `def bil_pangkat(x,y)`

Define function di pergunakan untuk menyatakan atau mendefinisikan suatu fungsi. Pada program ini di definisikan suatu define function bil_pangkat dengan parameter formal x,y.

➤ `if y == 0:`

If di pergunakan untuk memberikan kondisi jika $y = 0$.

➤ `return 1`

Maka, akan di tampilkan angka sebesar 1.

➤ `else:`

Else di pergunakan untuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `return x * bil_pangkat(x,y-1)`

Maka, akan di lakukan perhitungan bil_pangkat dengan x (bilangan) dikali dengan angka,pangkat-1 hingga pangkat dikurang 1 memiliki hasil 0.

➤ `x = int(input("Masukkan bilangan yang akan di pangkatkan : "))`

X di pergunakan sebagai variabel untuk menginputkan angka dengan tipe integer (bilangan bulat).

➤ `y = int(input("Masukkan pangkat : "))`

Y di pergunakan sebagai variabel untuk menginputkan pangkat.

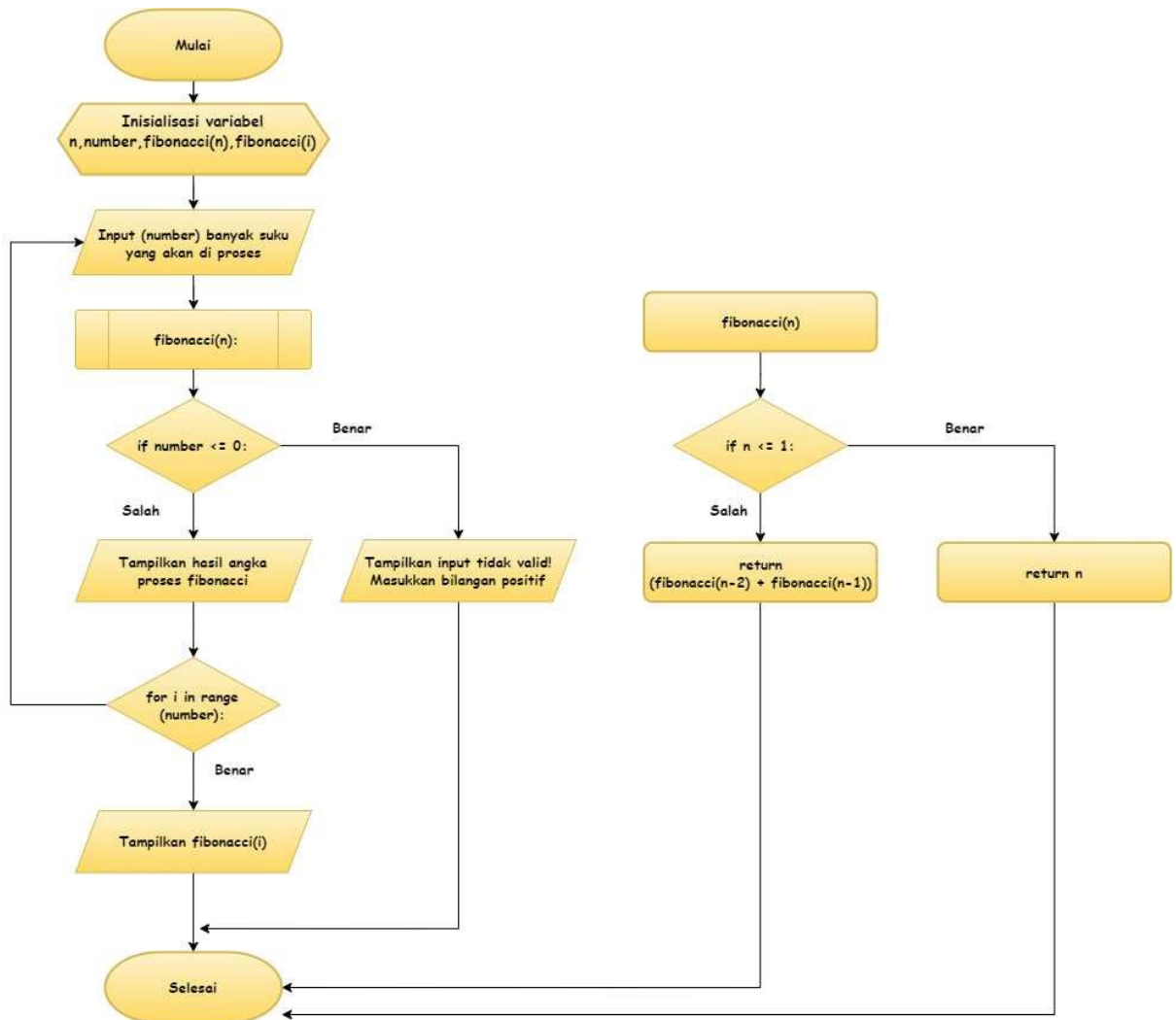
➤ `print("Di peroleh hasil bilangan berpangkat dari", x, " di pangkatkan dengan", y, "sebesar : ", bil_pangkat(x,y))`

Print di pergunakan untuk menampilkan hasil bilangan berpangkat. Pada program ini di inputkan bilangan sebesar 3 dan pangkat sebesar 5 dan di peroleh hasil sebesar 243.

Dengan perhitungan di lakukan return di kondisi else hingga y (pangkat) hingga $y-1 = 0$ dan di peroleh proses $3 * 3 * 3 * 3 * 3$ dan di peroleh hasil sebesar 243.

3. Deret Fibonacci

➤ Flowchart



➤ Kode Script Python

```
3. Deret Fibonacci

In [3]: def fibonacci(n):
        if n <= 1:
            return n
        else:
            return (fibonacci(n-2) + fibonacci(n-1))

        number = int(input("Masukkan banyak suku yang akan di proses : "))

        if number <= 0:
            print("Input tidak valid! Silahkan masukkan bilangan positif")
        else:
            print("Di peroleh hasil angka dari proses fibonacci : ")
            for i in range(number):
                print(fibonacci(i))
```

```
Masukkan banyak suku yang akan di proses : 10
Di peroleh hasil angka dari proses fibonacci :
0
1
1
2
3
5
8
13
21
34
```

Penjelasan Deskriptif :

➤ `def fibonacci(n):`

Define function di digunakan untuk menyatakan atau mendefinisikan suatu fungsi. Pada program ini di definisikan suatu define function fibonacci dengan parameter formal n.

➤ `if n <= 1:`

If di digunakan untuk kondisi jika n kurang dari sama dengan 1.

➤ `return n`

Maka, akan di tampilkan angka sebesar n atau sebesar (angka itu sendiri).

➤ `else :`

Else di digunakan ntuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `return(fibonaacci(n-2) + fibonacci (n-1))`

Maka, akan di lakukan perhitungan fibonacci dengan rumus pertambahan angka fibonacci dikurang dua suku sebelumnya di tambah dengan angka fibonacci satu suku sebelumnya.

➤ `number = int(input("Masukkan banyak suku yang akan di proses : "))`

Number di digunakan sebagai variabel untuk menginputkan banyak suku yang akan di proses dengan tipe integer (bilangan bulat). Pada program ini di inputkan sebanyak 10 suku.

➤ `if number <= 0:`

If di digunakan untuk memberikan kondisi jika suku yang di inputkan kurang dari atau sama dengan 0.

➤ `print("Input tidak valid! Silahkan masukkan bilangan positif")`

Maka, akan di tampilkan input tidak valid! Silahkan masukkan bilangan positif.

➤ `else:`

Else di pergunakan untuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `print("Di peroleh hasil angka dari proses Fibonacci : ")`

Maka, akan di tampilkan hasil angka dari proses fibonacci.

➤ `for i in range(number)`

Dengan melakukan looping for dalam rentang number (10 suku).

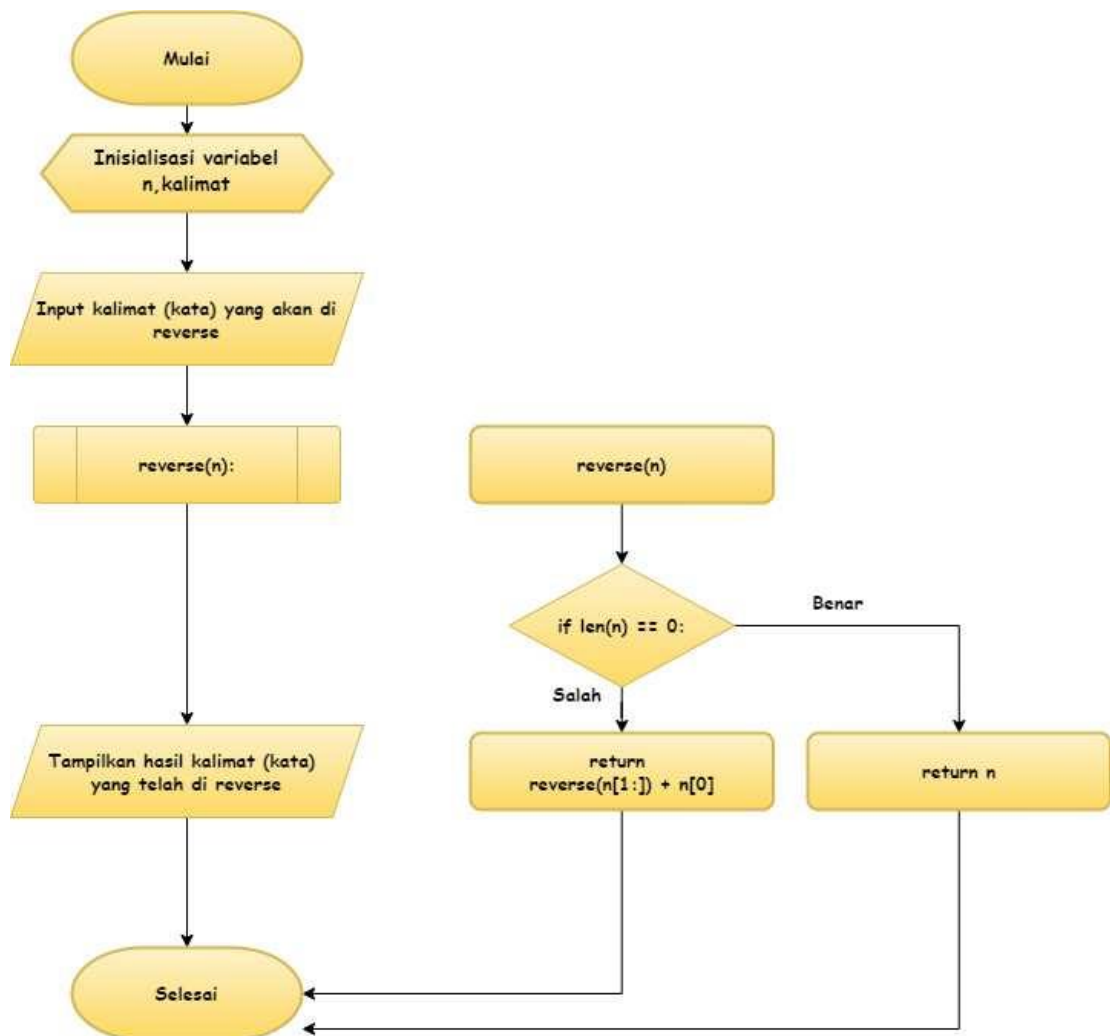
➤ `print((fibonacci)(i))`

Print di pergunakan untuk menampilkan angka-angka yang di peroleh dari proses Fibonacci sebanyak 10 suku. Dengan contoh perhitungan pada angka ke 4 di peroleh dari hasil perhitungan angka 2 di tambah dengan hasil perhitungan angka ke 3.

Dan di peroleh angka-angka fibonacci sebanyak 10 suku, yakni 0,1,1,2,3,5,8,13,21,34.

4. Membalikkan String

➤ Flowchart



➤ Kode Script Python

```
4. Membalikkan String

In [6]: def reverse(n):
        if len(n) == 0:
            return n
        else:
            return reverse(n[1:]) + n[0]

        kalimat = str(input("Masukkan kata yang akan di reverse : "))
        print("Di dapatkan hasil kata yang telah di reverse : ", reverse(kalimat))

        Masukkan kata yang akan di reverse : halo
        Di dapatkan hasil kata yang telah di reverse :  olah
```

Penjelasan Deskriptif :

➤ `def reverse(n):`

Define function di pergunakan untuk menyatakan atau mendefinisikan suatu fungsi. Pada program ini di definisikan suatu define function reverse dengan parameter formal n.

➤ `if len(n) == 0:`

If di pergunakan untuk memberikan kondisi jika panjang string yang di inputkan sama dengan 0.

➤ `return n`

Maka, akan di tampilkan string n atau (string itu sendiri).

➤ `else:`

Else di pergunakan untuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `return reverse(n[1:]) + n[0]`

Maka, akan di lakukan reverse (pembalikan string) dengan proses mengambil indeks setelah 0 hingga terakhir.

➤ `kalimat = str(input("Masukkan kata yang akan di reverse : "))`

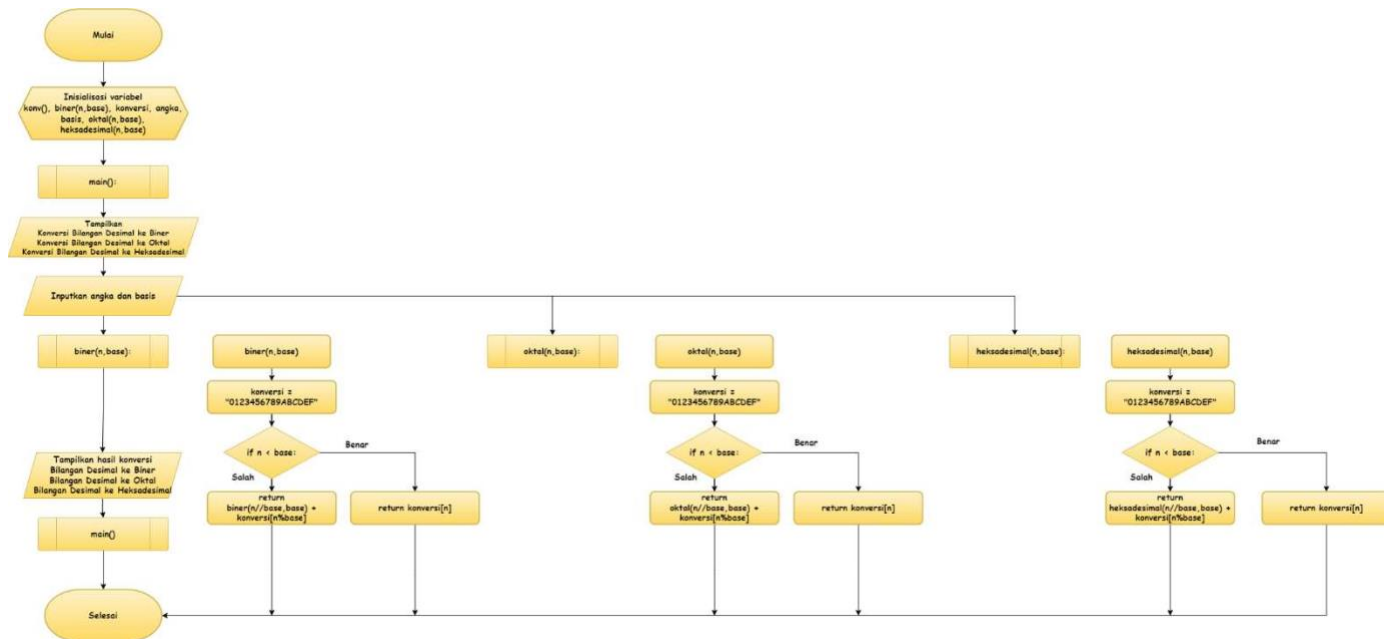
Kalimat di pergunakan sebagai variabel untuk menginputkan kata yang akan di reverse. Pada program ini di inputkan kata halo.

➤ `print("Di dapatkan hasil kata yang telah di reverse : ", reverse(kalimat))`

Print di sini di pergunakan untuk menampilkan kata yang telah di lakukan proses reverse. Dan di peroleh hasil kata halo yang telah di reverse menjadi olah.

5. Konversi Bilangan Desimal ke Bilangan Biner, Oktal, Heksadesimal

➤ Flowchart



➤ Kode Script Python

5. Konversi Bilangan Desimal Ke Bilangan Biner, Oktal, Hexadesimal

```
In [3]: def konv():
        print("Konversi Bilangan Desimal ke Bilangan Biner")
        def biner(n, base):
            konversi = "0123456789ABCDEF"
            if n < base:
                return konversi[n]
            else:
                return biner(n//base, base) + konversi[n%base]
        angka = int(input("Masukkan angka yang akan di konversi : "))
        basis = int(input("Masukkan basis : "))
        print("Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Biner, yakni : ", biner(angka, basis))
        print()

        print("Konversi Bilangan Desimal ke Bilangan Oktal")
        def oktal(n, base):
            konversi = "0123456789ABCDEF"
            if n < base:
                return konversi[n]
            else:
                return oktal(n//base, base) + konversi[n%base]
        angka = int(input("Masukkan angka yang akan di konversi : "))
        basis = int(input("Masukkan basis : "))
        print("Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Oktal, yakni : ", oktal(angka, basis))
        print()
```

```

print("Konversi Bilangan Desimal ke Bilangan Heksadesimal")
def heksadesimal(n,base):
    konversi = "0123456789ABCDEF"
    if n < base :
        return konversi[n]
    else:
        return heksadesimal(n//base,base) + konversi[n%base]
angka = int(input("Masukkan angka yang akan di konversi : "))
basis = int(input("Masukkan basis : "))
print("Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Heksadesimal, yakni : ", heksadesimal(angka,basis))
print()

konv()

Konversi Bilangan Desimal ke Bilangan Biner
Masukkan angka yang akan di konversi : 60
Masukkan basis : 2
Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Biner, yakni : 111100

Konversi Bilangan Desimal ke Bilangan Oktal
Masukkan angka yang akan di konversi : 60
Masukkan basis : 8
Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Oktal, yakni : 74

Konversi Bilangan Desimal ke Bilangan Heksadesimal
Masukkan angka yang akan di konversi : 60
Masukkan basis : 16
Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Heksadesimal, yakni : 3C

```

Penjelasan Deskriptif :

➤ **def konv():**

Pada program ini di buat define function konv (konversi) untuk mengkonversi angka desimal ke biner, oktal, dan heksadesimal.

➤ **def biner(n,base):**

Pada program ini di buat define function biner untuk mengkonversi bilangan desimal ke bilangan biner dengan parameter formal n, base.

➤ **konversi = "0123456789ABCDEF"**

Konversi sebagai variabel untuk mendeklarasikan atau menuliskan konversi untuk bilangan desimal ke bilangan biner.

➤ **if n < base:**

If di pergunkan untuk memberikan kondisi jika n (angka) kurang dari base (basis).

➤ **return konversi[n]:**

Maka, akan di return atau di kembalikan pada variabel konversi.

➤ **else:**

Else di pergunkan untuk memberikan kondisi jika if tidak terpenuhi.

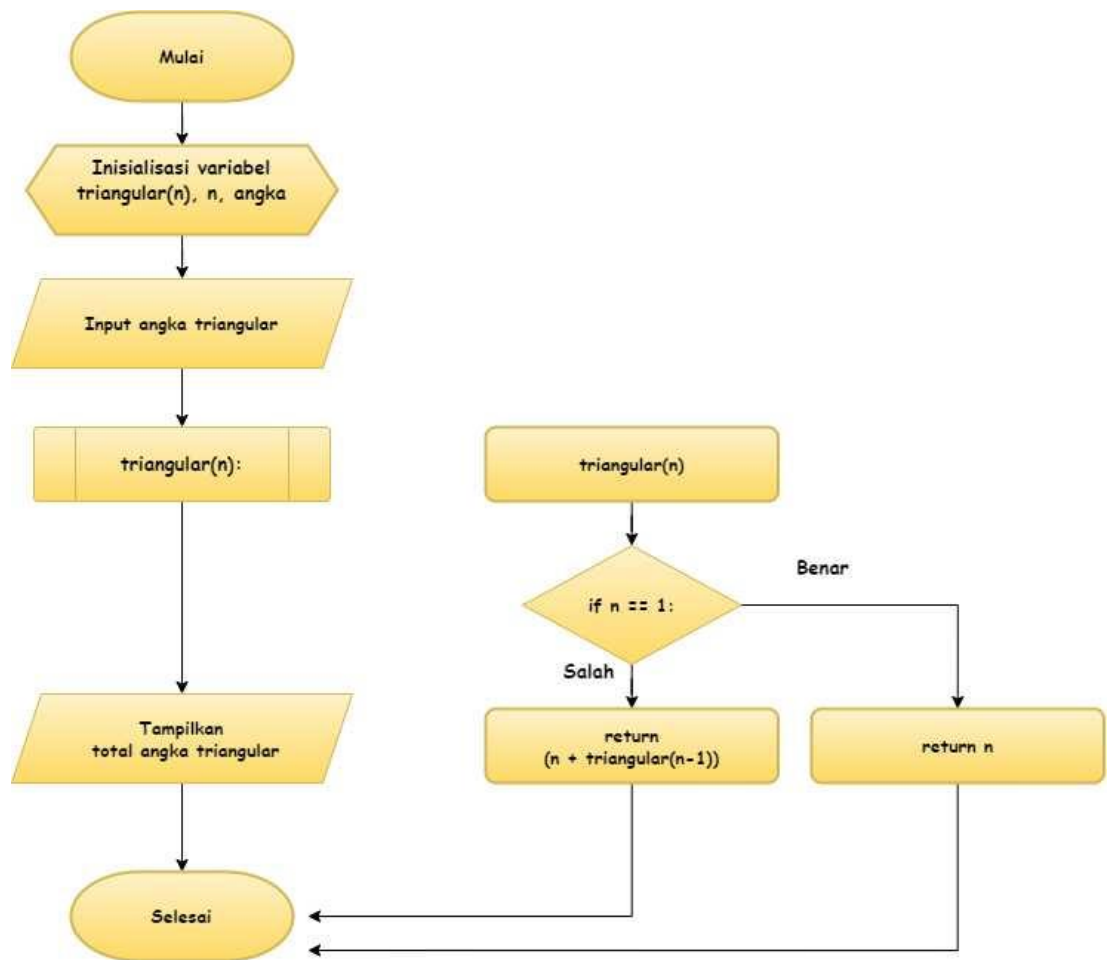
➤ **return biner(n//base,base) + konversi[n%base]**

Maka, akan di lakukan perhitungan desimal to biner dengan proses ke define function n di bagi dengan base (mengembalikan angka float ke desimal) di tambah dengan konversi n di modulus dengan base. Dan ini berlaku pada, oktal dan heksadesimal.

- `angka = int(input("Masukkan angka yang akan di konversi : "))`
 Angka di sini di pergunakan sebagai variabel untuk menginputkan angka yang akan di konversi dengan tipe integer (bilangan bulat).
- `basis = int(input("Masukkan basis : "))`
 Basis di sini di pergunakan sebagai variabel untuk menginputkan basis dengan tipe integer (bilangan bulat).
- `print("Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Biner, yakni : ", biner(angka,basis))`
 Pada program ini di tampilkan di inputkan angka sebesar 60 dan di inputkan basis sebesar 2. Di dapatkan hasil konversi desimal ke biner, yakni 111100.
- `print()`
 Print di sini di pergunakan untuk menampilkan formatting print spasi.
- `def oktal(n,base):`
 Pada program ini di definisikan suatu define function oktal dengan parameter formal n (angka) dan base (basis).
- `print("Di peroleh hasil konversi dari Bilangan Desimal ke Bilangan Oktal, yakni : ", oktal(angka,basis))`
 Pada program ini di tampilkan di inputkan angka sebesar 60 dan basis sebesar 8. Dan di dapatkan hasil konversi desimal ke oktal, yakni 74.
- `def heksadesimal(n,base):`
 Pada program ini di definisikan suatu define function heksadesimal dengan parameter formal n (angka) dan base (basis).
- `print("Di peroleh hasil konversi dari Bilangan Desimal ke Heksadesimal, yakni : ", heksadesimal(angka,basis))`
 Pada program ini di tampilkan di inputkan angka sebesar 60 dan basis sebesar 16. Dan di dapatkan hasil konversi desimal ke heksadesimal, yakni 3C.
- `konv()`
 Konv di sini di pergunakan untuk pemanggilan fungsi sesuai apa yang telah di tuliskan di atas.

6. Triangular Numbers

➤ Flowchart



➤ Kode Script Python

```
6. Triangular Numbers

In [4]: def triangular(n):
        if n == 1:
            return n
        else:
            return (n + triangular(n-1))
        angka = int(input("Masukkan angka : "))
        print("Di peroleh total triangular numbers, yakni : ", triangular(angka))

Masukkan angka : 5
Di peroleh total triangular numbers, yakni : 15
```

Penjelasan Deskriptif:

➤ `def triangular(n):`

Define function di digunakan untuk menyatakan atau mendefinisikan suatu fungsi. Pada program ini di definisikan define function triangular dengan parameter formal n.

➤ `if n == 1:`

If di digunakan untuk memberikan kondisi jika n sama dengan 1.

➤ `return n`

Maka, akan di tampilkan atau di hasilkan n atau angka itu sendiri.

➤ `else:`

Else di digunakan untuk memberikan kondisi jika pernyataan if tidak terpenuhi.

➤ `return (n + triangular(n-1))`

Maka, akan di lakukan proses perhitungan triangular dengan n di tambah dengan n -1 hingga proses n -1 menghasilkan nilai sebesar 0.

➤ `angka = int(input("Masukkan angka : "))`

Angka di digunakan sebagai variabel untuk menginputkan angka yang akan di proses dengan proses triangular dengan tipe integer (bilangan bulat).

➤ `print("Di peroleh total triangular numbers, yakni : ", triangular(angka))`

Print di digunakan untuk menampilkan hasil total dari triangular numbers. Pada program ini di inputkan angka sebesar 5 dan di peroleh proses perhitungan $5 + 4 + 3 + 2 + 1$ dan di peroleh hasil sebesar 15.

7. Pembuktian Bahwa Faktorial dari $0! = 1$

No _____
Date _____

Pembahasan :

Dapat diketahui, Faktorial memiliki makna yang erat dengan permutasi dan kombinatori. Sehingga, untuk membuktikan bahwa Faktorial dari $0! = 1$ dapat didefinisikan :

$$n! = n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$$

Lalu, bagi kedua ruas dengan n , diperoleh :

$$\frac{n!}{n} = \frac{n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1}{n}$$

$$\frac{n!}{n} = (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$$

$$\frac{n!}{n} = (n-1)!$$

Balik kedua ruas, diperoleh :

$$(n-1)! = \frac{n!}{n}$$

Agar didapatkan hasil $0!$, substitusikan $n=1$, diperoleh :

$$(1-1)! = \frac{1!}{1}$$

$$(1-1)! = \frac{1!}{1}$$

$$0! = \frac{1!}{1}$$

$$0! = 1$$

Jadi, dapat dibuktikan bahwa hasil dari $0! = 1$.

8. Pembuktian Bahwa $1 \% 2 = 1$

No _____
Date _____

Pembahasan :

Dapat diketahui, modulus adalah sebuah operasi hitung yang menghasilkan sisa pembagian suatu bilangan terhadap bilangan lainnya. Untuk melakukan pembuktian bahwa $1 \% 2 = 1$.

Dapat dicari dengan kelipatan pembagi 2 yang sama dengan atau lebih kecil dari 1.

Kelipatan 2 = 0, 2, 4, ..., n

Gunakan $n=0$, diperoleh :

$$1 - 0 = ?$$

$$1 - 0 = 1 \quad (\text{kurangi kelipatan pembagi tertinggi dan divide})$$

Jadi, dapat dibuktikan bahwa $1 \% 2 = 1$.