

Pemrograman Dasar dan Pengantar Algoritma. (Python Edition)

By Abdan Hafidz

Kenalan Dikit

Website : <https://abdanhafidz.com>

Linkedin : <https://www.linkedin.com/in/abdan-hafidz/>

Github : github.com/abdan_hafidz

List of Contents

- *Pengenalan Konsep Pemrograman*
- *Syntax Dasar Python: Variabel & Tipe Data, Masukan & Keluaran, Percabangan (if-else), Perulangan for – while, Subprogram*
- *Pengenalan Algoritma*
- *Problem Solving Soal Pemrograman Dasar*

Disclaimer

- *Kalau aku kecepetan bilang*
- *Kalau **koneksiku putus – putus, pinjam dulu seratus***

01

Konsep Pemrograman

Pada sub-materi ini anda diminta untuk memaksimalkan imajinasi anda, karena kali ini kita akan belajar dengan cara yang sedikit berbeda, materi akan dihubungkan kepada studi kasus di dunia nyata dan semoga anda lebih mudah memahaminya

Kenapa harus belajar Pemrograman?

Pake Nanya

Yaudah Kita
Pelajarin
aja ya Pelan Pelan

Apa itu Pemrograman?

Coba kamu bayangkan ada Seorang Ibu yang memberikan tugas tertulis di pintu kulkas kepada anaknya Bernama Ani seperti di bawah ini :

Buat Sweetie Pookie Bear Sayangnya Mamah, Ani :

Mamah liburan bentar ya ke Jogja kamu jaga Rumah.
Setiap 2 Minggu sekali Rumah disapu, Cuci Pakaian
deterjennya 2 liter / 1kg baju ya

Anak Lagi
Nganggur gaada
kerjaan.

Anak Lagi Push
Emel / Ngerjain
Sesuatu



Aku Diam

Waktunya Disuruh
Suruh

Apa itu Pemrograman?

Dari Contoh di atas kita dapat menilai bahwa Ibunya Ani :

- Memberikan perintah kepada Ani untuk melakukan suatu pekerjaan
- Pekerjaan yang dilakukan dapat berulang kali

Setiap 2 Minggu sekali Rumah disapu,



Apa itu Pemrograman?

Dari kasus di atas yang dilakukan Ibu kepada Ani disebut sebagai pemrograman.

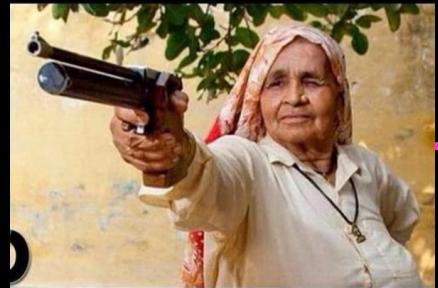
Pemrograman adalah proses menulis suatu program yang terdiri dari sekumpulan instruksi atau perintah untuk melakukan pekerjaan tertentu

```
procedure lakukan_pekerjaan:  
  Sapu_rumah(2x sehari);  
  cuci_pakaian({  
    detergen = 2 liter x berat baju  
  })
```

Apa itu Pemrograman?

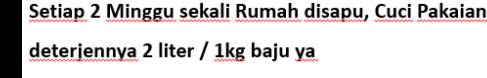
Dari kasus di atas yang dilakukan Ibu kepada Ani disebut sebagai **pemrograman**.

Hanya saja dalam konteks Teknologi Ibu itu adalah kita / programmer (orang yang menulis program) lalu Ani adalah Komputer.



Apa itu Pemrograman?

Dalam melakukan pekerjaannya Ani pasti bisa melaksanakan pekerjaannya dengan baik jika

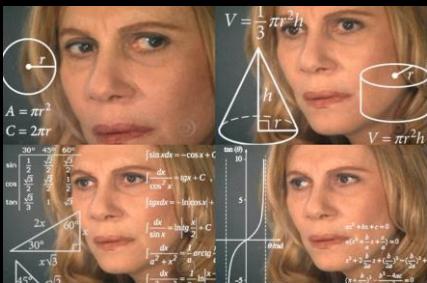
- Ada parameter pekerjaan yang jelas -> 
Setiap 2 Minggu sekali Rumah disapu, Cuci Pakaian
deterjennya 2 liter / 1kg baju ya
- Sehingga pemrograman bisa disebut efektif jika ada parameter yang jelas

Bagaimana Cara Memprogram?

Jika dilihat dari yang dilakukan ibu tadi :

Ibu Menulis Perintah (**Coding**)

Proses menulis ini kita sebut sebagai
Coding

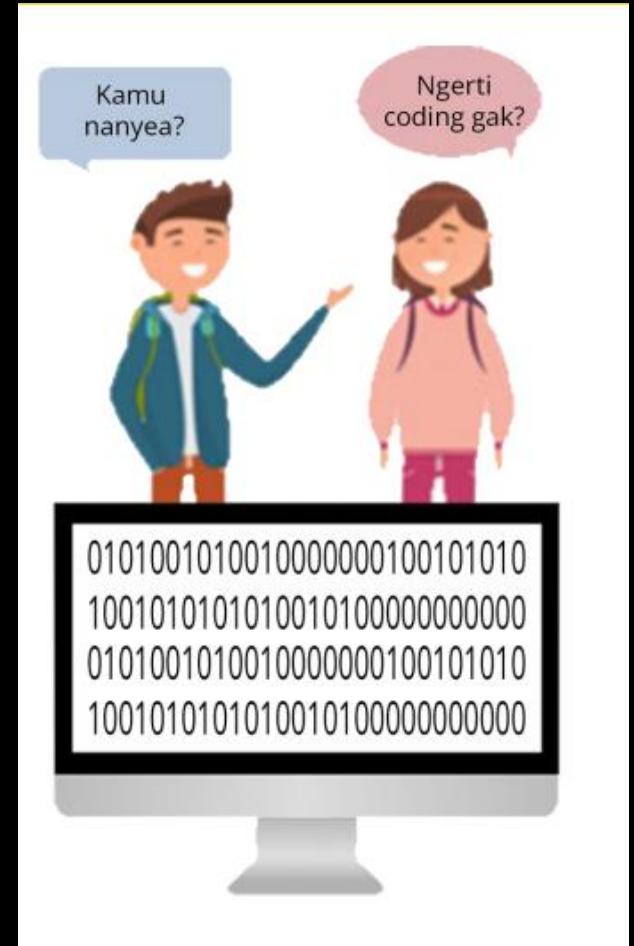


Si Ani membaca dan memahami
Perintah ibu (**Compiling**)

Proses memahami perintah oleh Ani
disebut sebagai **Compiling**

Coding

Pada dasarnya, komputer tidak mengerti Bahasa Manusia sehingga untuk memerintahkan computer melakukan sesuatu, Manusia harus menuliskan perintahnya dalam Bahasa Pemograman Komputer. Proses penulisan itulah yang disebut Coding.



Coding Vs Programming

Menjadi topik yang lumayan hangat dibicarakan.

Coding dan Programming sebenarnya adalah dua hal yang tidak sama

Karena sebenarnya Coding adalah sebagian dari proses Programming dan ini berarti Coding belum tentu programming

Coding Vs Programming

Kumpulan perintah bisa disebut sebagai program dan menjadi bagian dari Programming jika :

- Jika di dalamnya terdapat parameter pengukuran yang jelas
- Perintah bisa dilaksanakan berulang kali (dinamis)
- Perintah biasanya diintegrasikan dengan Formula, Rumus, atau Permodelan Matematis / Logika.

Bagaimana Cara Memprogram?

Untuk memprogram kita membutuhkan :

- **Dalam Proses Coding**
 1. Kita membutuhkan Bahasa yang sesuai dengan kebutuhan program kita.
 2. Kode ditulis dalam format text/script dan membutuhkan media menulis kode dan ini disebut sebagai **[IDE*]: Text Editor, Tools, Environment**

Bagaimana Cara Memprogram?

Untuk memprogram kita membutuhkan :

- **Dalam Proses Compiling**

1. Kita membutuhkan Compiler. Yaitu suatu mesin yang membaca perintah yang kita tulis lalu mengeksekusinya. Juga akan ditemui *Interpreter (Pemroses Lain)* tapi sejatinya bekerja dengan fungsi yang sama. Ibaratkan otak pada manusia.

Bahasa Pemrograman

Bahasa adalah standar format perintah yang kita berikan. Seperti manusia dalam Pemrograman Bahasa Juga ada banyak. Variasi Bahasa Pemrograman dikarenakan setiap jenis program memiliki kebutuhan yang berbeda – beda dalam penulisan perintah.

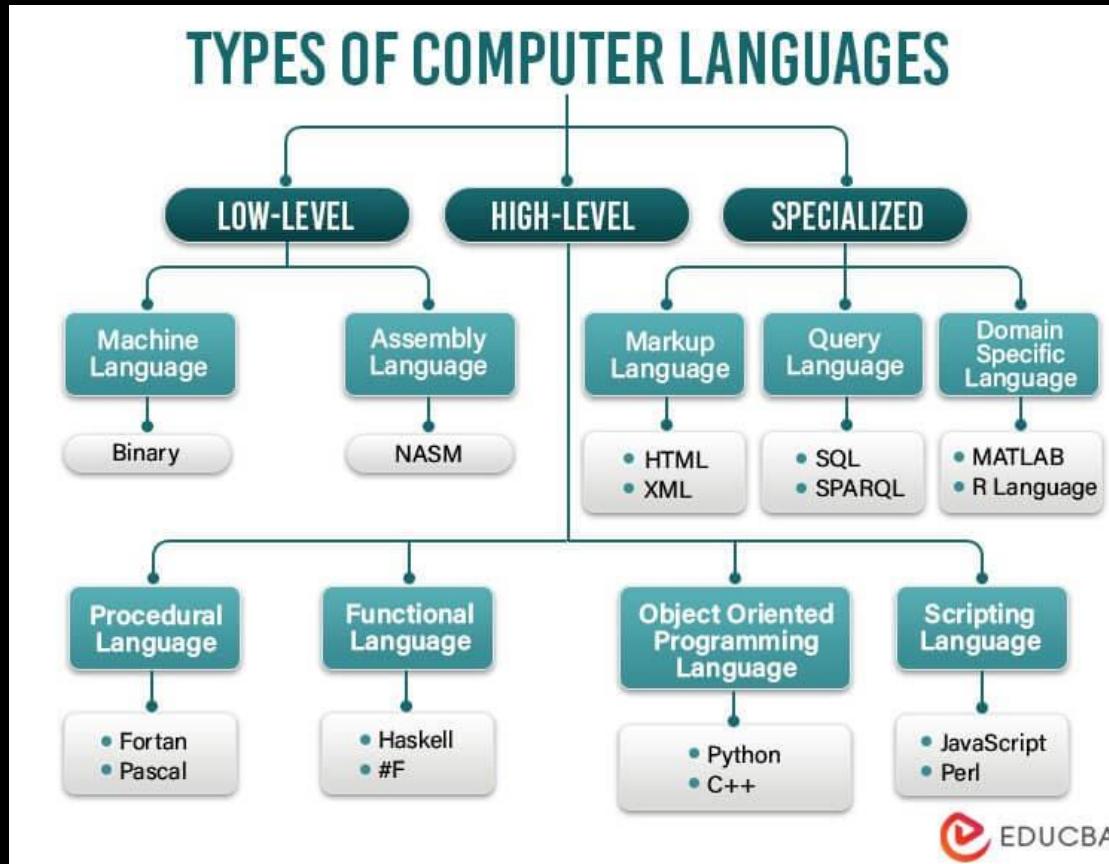
Inggris	I don't care
Indonesian	Bodo amat
Jawa	Luweh

Bahasa Pemrograman

Walaupun variasi Bahasa yang banyak. Tapi sejatinya setiap Bahasa Pemrograman selalu memiliki tujuan perintah yang serupa. Terkadang yang menjadi pembeda setiap Bahasa Pemrograman adalah :

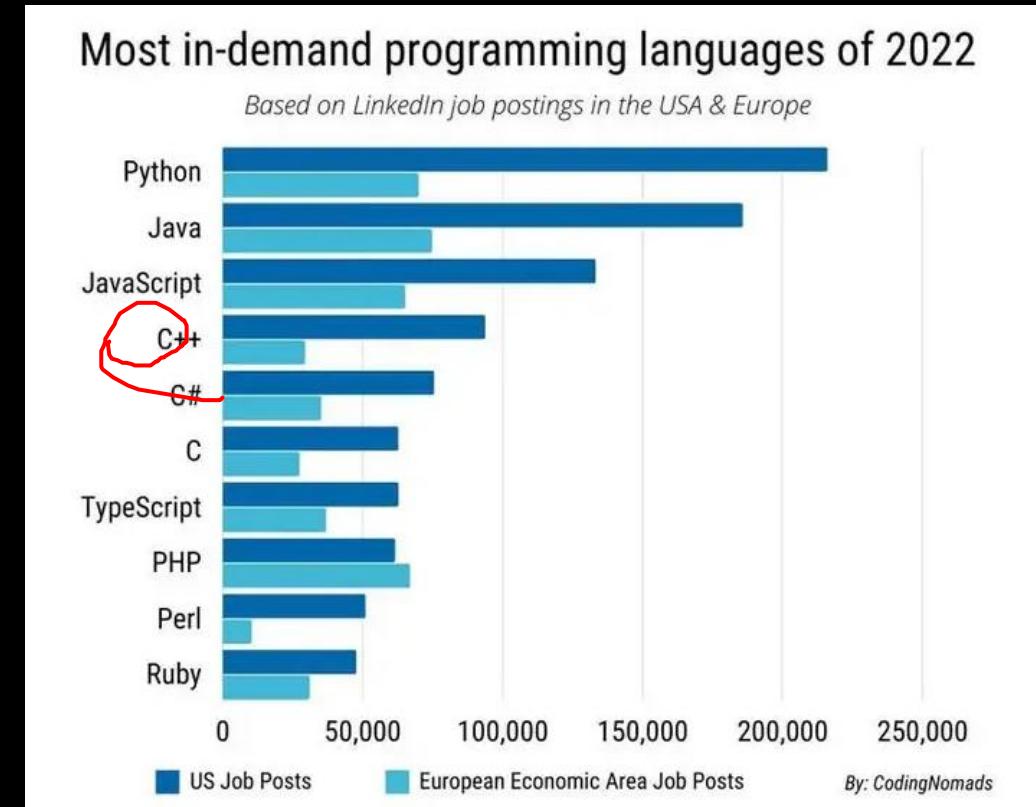
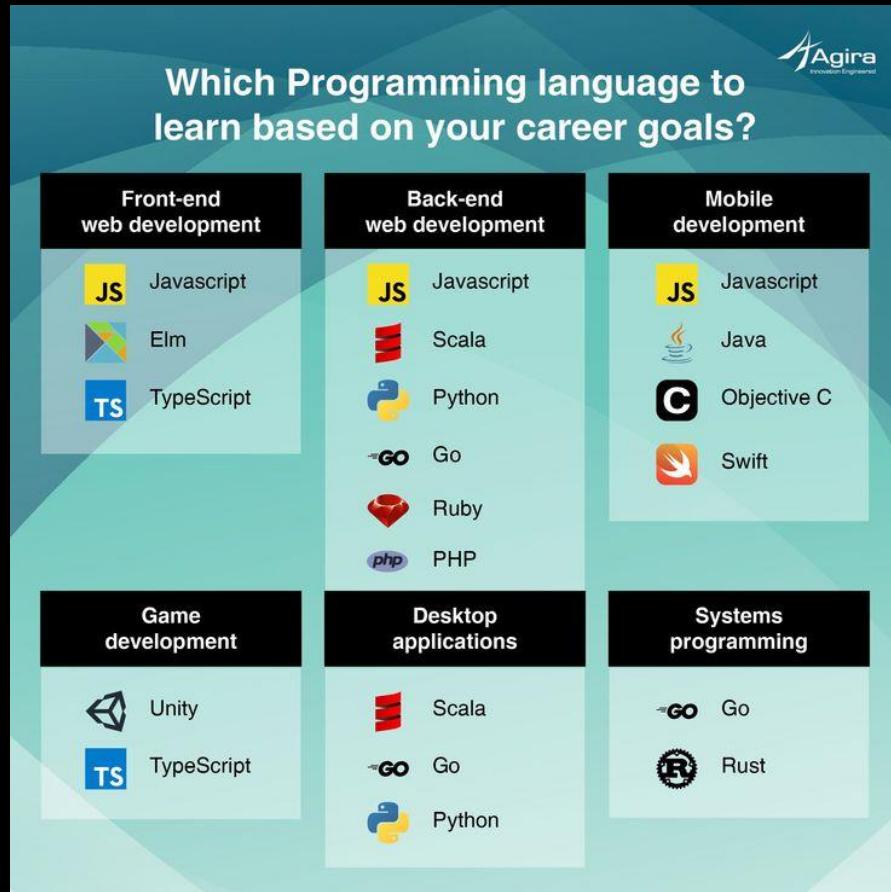
- Kemudahan penulisan
- Kecepatan Waktu Proses
- Alokasi Memori Komputer

Bahasa Pemrograman

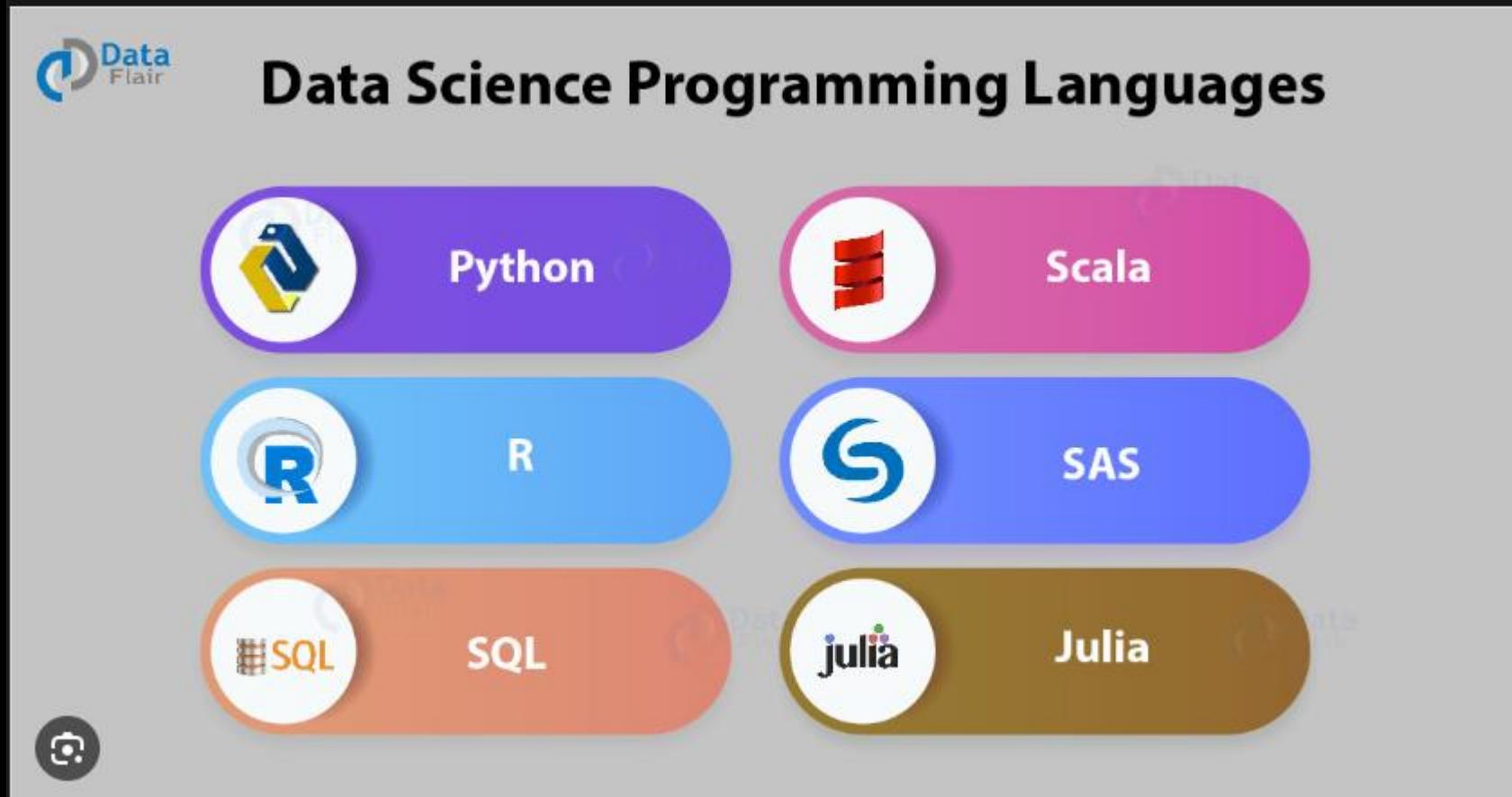


Semakin tinggi level bahasanya, semakin mendekati Bahasa manusia dan mudah dipahami.

Bahasa Pemrograman



Bahasa Pemrograman



Bahasa Pemrograman



Created With Love By : Abdan Hafidz

Private Usage Only

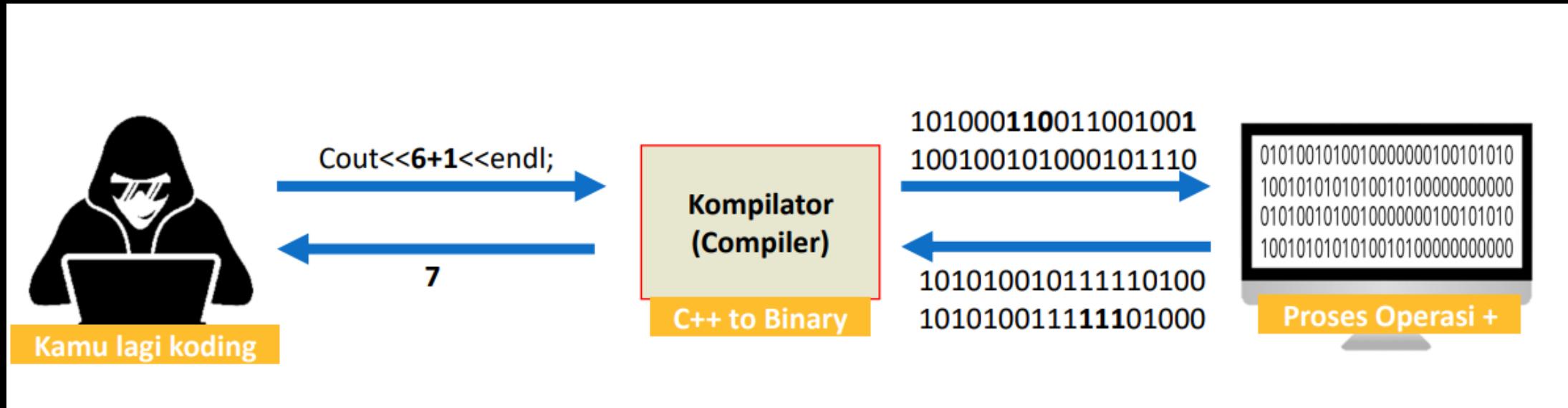
Bahasa Pemrograman

Bahasa yang kita gunakan untuk pemrograman di RKA dominan Python, mungkin Sebagian ketemu Javascript.

Compiler

- Syntax-syntax tiap Bahasa pemograman telah ditetapkan, sehingga akan memudahkan Kompilator mengerti kode yang kita tulis.
- Nantinya, Bahasa pemograman yang telah kita tulis akan diterjemahkan kedalam sebuah Kompilator supaya dapat dimengerti dan dieksekusi oleh Komputer.
- Compiler = program komputer yang dapat menerjemahkan bahasa pemrograman tingkat tinggi ke bahasa mesin

Compiler

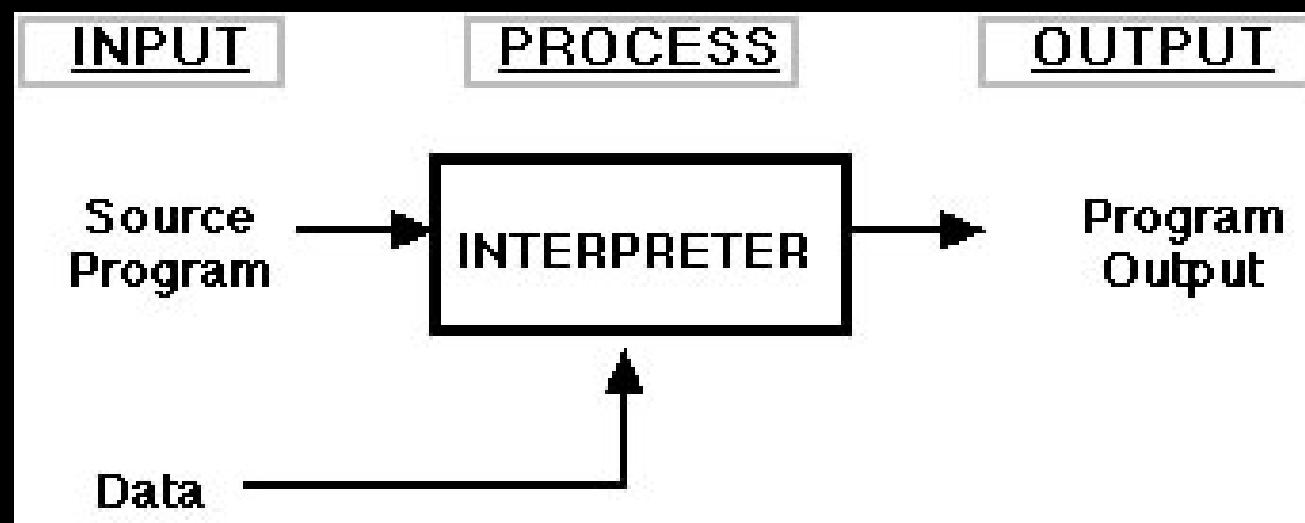


Compiler

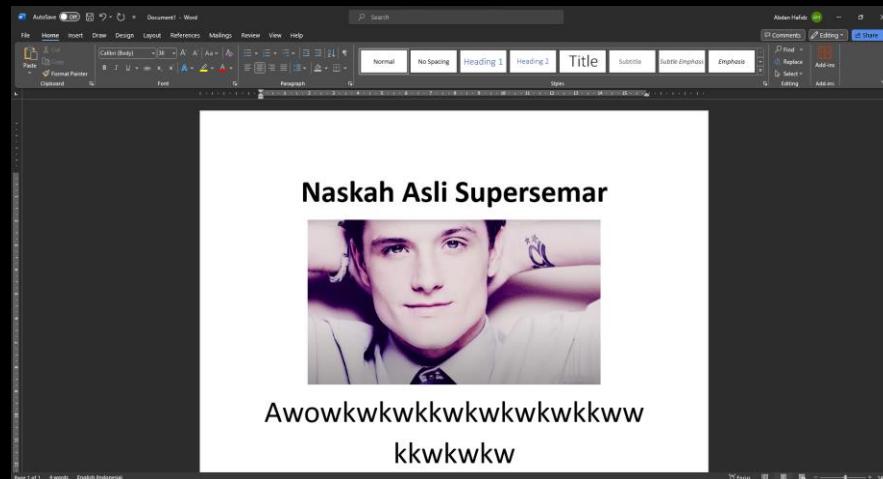
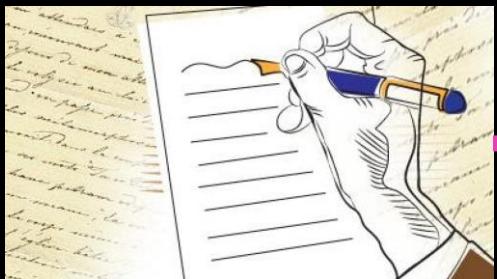
- Tapi sayangnya Python bukan Compiler Based
- Ini yang membuat hal yang bisa kita ghibahin dari si Python yaitu (LELET)

Interpreter

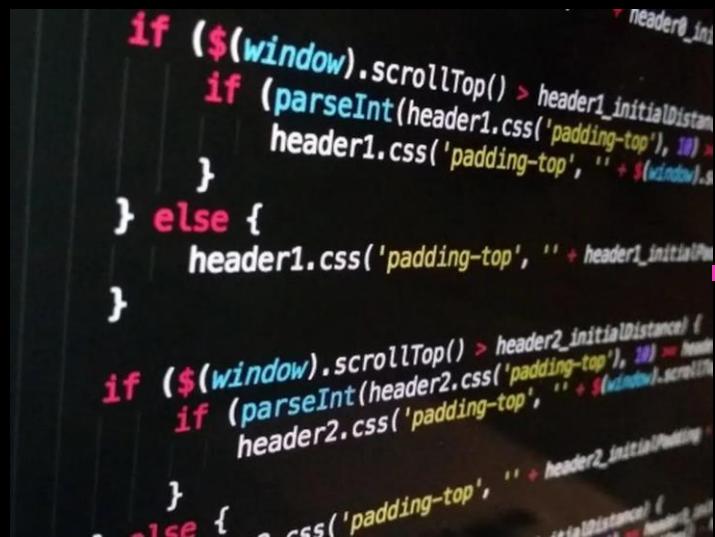
- Python memiliki mekanisme eksekusi program yaitu melalui proses interpreting.
- Di mana program yang sudah kita tulis akan langsung dibaca dalam bentuk binary oleh komputer



IDE (Integrated Development Env.)



IDE (Integrated Development Env.)



02

Mulailah Memprogram

Pada sub-materi ini diharapkan anda memperhatikan seluruh tutorial yang diberikan. Pembelajaran bersifat runut dan setiap materi saling berkaitan satu sama lain. Jika anda tertinggal Sebagian materi maka anda akan sulit lanjut ke materi berikutnya.

***JIKA ADA HAL YANG DIBINGUNGKAN MOHON
LANGSUNG DITANYAKAN!***



Perhatikan Lalu Praktikkan!

*Gak boleh gak focus, gak boleh gak dipraktikkan, gak boleh gak
paham*

Persiapan dan Instalasi

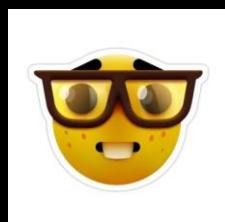
1. IDE/Compiler/Text Editor



With VSCode :

<https://code.visualstudio.com/docs/python/python-tutorial>

(Agak Ribet)



With Jupyter Notebook :

<https://jupyter.org/install> (Lumayan Mudah)

With Google Colab :



<https://colab.google/>

(Turu Deck)

Lanjut

AYO NGODINGG!!!

2.1

Struktur Python

Header Files , Namespace, Main Function

Struktur Dasar Python

Gaada Python EzPz, gaada
Struktur kek C, C++, atau C#

Langsung gas tulis kodanya aja

Struktur Dasar Python

Header Files / Library

```
import numpy → Header File
```

Dengan
menggunakan numpy
kita bisa

Struktur Dasar Python

Header Files

Tidak dengan Header Files

```
def goreng_ayam():
    panasin_minyak(suhu=80)
    cuci_ayam()
    masukkan_ke_wajan()
```



Struktur Dasar Python

Header Files

Dengan Header Files

```
import warteg_Bahari as wb  
wb.pesen_ayam_goreng();
```



2.2

Variabel & Tipe Data

Variable, Tipe Data Dasar (Integer (int,longint) , String, Character, Boolean, Double, Float)

Variabel

Variabel adalah istilah dalam dunia matematika, yang memetakan sebuah nama ke suatu nilai

```
namadepan = "Abdan"
```

```
namabelakang = "Hafidz"
```

```
Jumlahpacar = 0
```

```
namalengkap = namadepan + namabelakang
```

Variabel

Aturan penamaan variable :

1. Terdiri dari kombinasi karakter huruf, angka, dan underscore (_).
2. Tidak boleh dimulai dengan angka.
3. Huruf kapital dan huruf kecil dianggap berbeda. Artinya "a1" dan "A1" dianggap merupakan dua variabel yang berbeda.
4. Tidak boleh merupakan reserved word. Contoh reserved word pada python: *print*, *def*, *while*, *for*, atau *switch*.



Tipe Data

Tipe Data Manusia :

Data	Tipe Data Bagi Manusia
1, 2, 69, 30001288, 012736, 2.56, 3.14	Angka
Kucing, anjing, rubah	Huruf, kata
a, b, c, d, _, :, @	Huruf, karakter
Andi sedang belajar., Kamu lagi apa?,	Kalimat

Tipe Data

Tipe data dalam python biasanya akan ter-assign secara otomatis berdasarkan nilai yang kita berikan pada variable.

Ataupun kita bisa memberi tipe data berupa :

- **int** : bilangan bulat
- **str** : string -> Kalimat, kata,
- **float** : bilangan decimal
- **bool** : proposisi Boolean (T/F)
- **list, dictionary, dll** : Tipe data lanjutan

2.3

Assignment, Masukan dan Keluaran

Variable Declaration, Assignment, cin (Input), cout (Output)

Deklarasi Variabel

Gak perlu repot – repot declare langsung aja assign

Assignment

Assignment artinya memberikan nilai pada variabel

<namavariabel> = <nilai>

x = 9 → x bernilai 9 (x=9)

y = 5 → x bernilai 5 (x=5)

jumlah = x + y, kurang = x - y

Variabel jumlah akan bernilai hasil $x + y = 9 + 5 = 14$ dan kurang = $9 - 5 = 4$

Assignment

Assignment artinya memberikan nilai pada variabel

```
namadepan = "Abdan"
```

```
namabelakang = "Hafidz"
```

```
namalengkap = namadepan + " " + namabelakang  
namalengkap = "Abdan Hafidz"
```

spasi

Assignment

Assignment artinya memberikan nilai pada variabel

```
jumlahpacar = 0  
punyapacar = bool(jumlahpacar > 0)  
jomblo = True
```

*Karena jumlah pacar tidak >0
Maka variable punyapacar akan
bernilai false.*

Ini dinamakan sebagai proposisi

Assignment

Assignment artinya memberikan nilai pada variabel

```
A = 5, B = 2
```

```
X = float(A/B) X = 2.5 , karena tipe data X adalah float (Desimal)
```

```
C = 5, D = 2
```

```
Y = C // D Y = 2 , nilai dipaksa bulat dengan melakukan pembulatan ke bawah (floor) 2.5 -> 2 ,  
div
```

Keluaran

Hasil dari assignment dapat ditampilkan di layar menggunakan print()

```
namadepan = "Abdan"  
namabelakang = "Hafidz"  
namalengkap = namadepan + " " + namabelakang  
print("Halo")  
print(namalengkap)
```

*Output : Halo
Abdan Hafidz*

Keluaran

Penggabungan Output mendatar

```
nama = "Abdan Hafidz"  
usia = 18  
gender ="L"  
print("Halo : ", nama , "Usia : ", usia)
```

Output : Halo :Abdan Hafidz Usia :18

Keluaran

Penggabungan Output mendatar

```
nama = "Abdan Hafidz"  
usia = 18  
gender ="L"  
print("Halo",end="")  
print(nama,end="")  
print("Usia:",usia)
```

Output : Halo :Abdan Hafidz Usia :18

Keluaran

Penggabungan Output perbaris

```
nama = "Abdan Hafidz"
```

```
usia = 18
```

```
print("Halo :", nama)
```

```
Print("Usia :", usia)
```

*Output : Halo :Abdan Hafidz
Usia :18*

Keluaran

Formatting Output

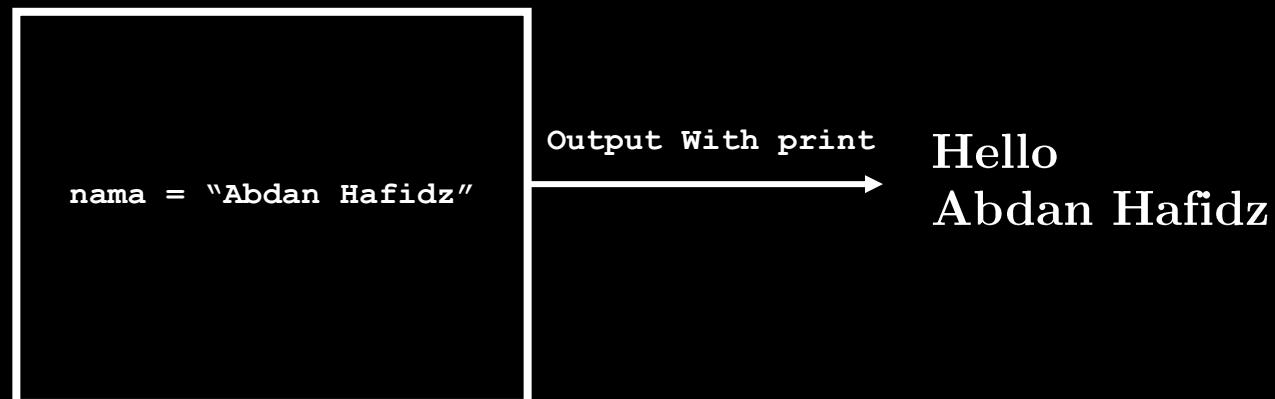
```
nama = "Abdan Hafidz"  
usia = 18  
print(f"Halo : {nama} \n Usia : {usia}")
```

*Output : Halo :Abdan Hafidz
Usia :18*

Masukan

Kita ingin program kita menjadi lebih interaktif. Orang dapat menggunakan program kita dengan fleksibel.

Jika sebelumnya kita membuat program mencetak nama:

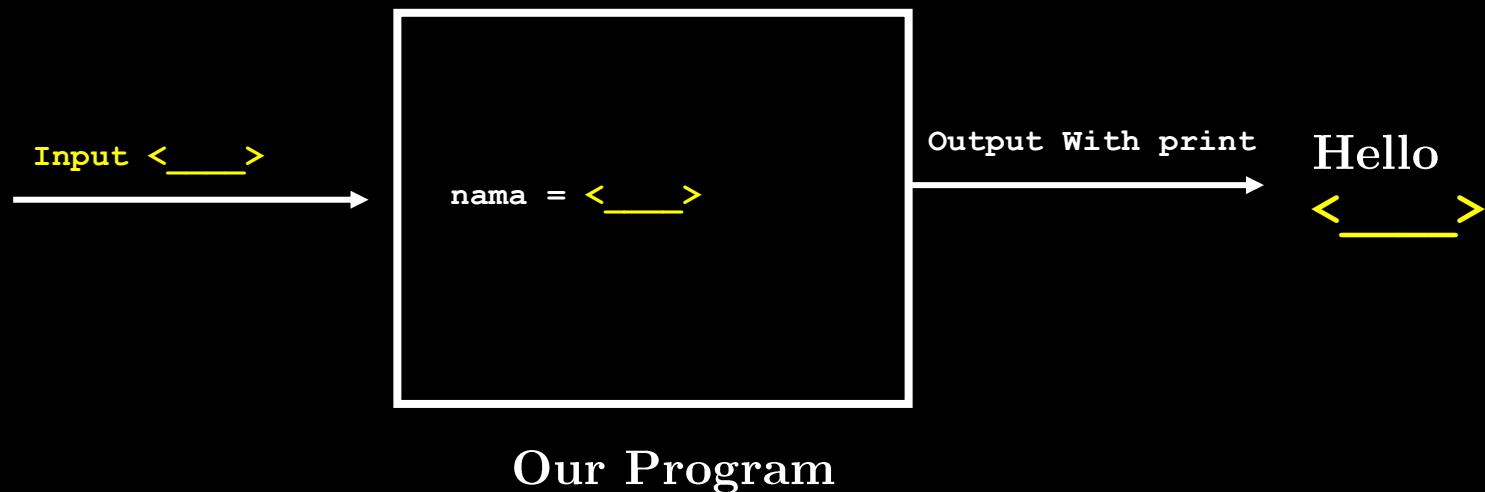


Our Program

Masukan

Kita ingin program kita menjadi lebih interaktif. Orang dapat menggunakan program kita dengan fleksibel.

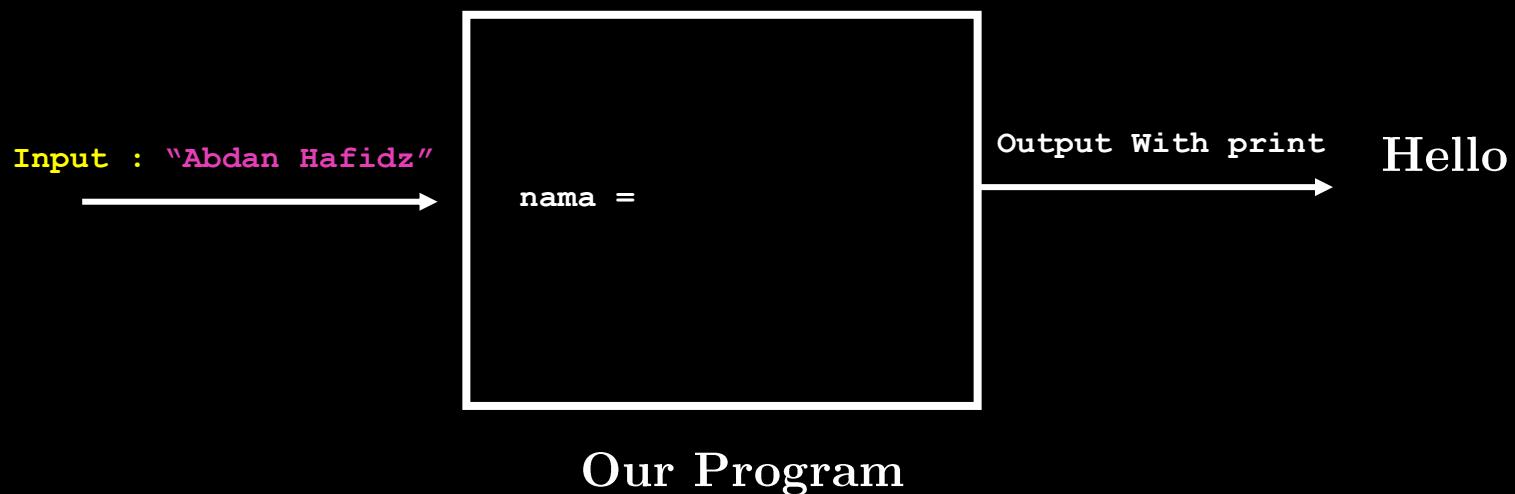
Jika sebelumnya kita membuat program mencetak nama:



Masukan

Kita ingin program kita menjadi lebih interaktif. Orang dapat menggunakan program kita dengan fleksibel.

Jika sebelumnya kita membuat program mencetak nama:



Masukan

Untuk menggunakan input, kita bisa memanfaatkan
input()

```
nama = input()  
print("Halo :", nama)
```

Masukan

Kita bisa melakukan casting nantinya input bertipe data apa, tapi secara default data akan terstore sebagai string dengan `input()`

```
usia = int(input())
x = float(input())
print(x + usia)
```

Input Map

Multiple data bisa dicast sebagai single line input dengan menggunakan map

```
namadepan, namabelakang = map(str, input().split())
usia, jumlahpacar = map(int, input().split())
```

Operator Python

Python Operator Precedence

Precedence	Operator Sign	Operator Name
Highest	**	Exponentiation
	+X, -X, ~X	Unary positive, unary negative, bitwise negation
	*, /, //, %	Multiplication, division, floor, division, modulus
	+, -	Addition, subtraction
	<<, >>	Left-shift, right-shift
	&	Bitwise AND
	^	Bitwise XOR
		Bitwise OR
	==, !=, <, <=, >, >=, is, is not	Comparison, identity
	not	Boolean NOT
	and	Boolean AND
Lowest	or	Boolean OR

Increament - Decreament

Kita bisa menambahkan nilai dari variable yang sudah terassign :

x = 5

x += 2 *x sekarang adalah 7 karena $x += 2$ itu artinya $x' = x + 2$*

x *= 2 *x sekarang adalah 14 karena $x *= 2$ itu artinya $x' = x * 2$*

x /= 7 *x sekarang adalah 2 karena $x /$ itu artinya $x' = x / 7$*

x ** 2 *x sekarang adalah 4 karena $x ** 2$ itu artinya $x' = x$ pangkat 2*

Increament - Decreament

Kita bisa menambahkan nilai dari variable yang sudah terassign :

```
nama = "Abdan"
```

```
nama += " Hafidz"      nama sekarang adalah "Abdan Hafidz"  
                        karena nama += " Hafidz" itu artinya  
                        nama' = nama + " Hafidz"
```

Latihan Dulu Brow

Deskripsi Soal

Sebuah Apel dihargai sebesar N dan sebuah Jeruk dihargai sebesar M . Jika Andi membeli X buah Apel dan Y buah Jeruk. Tentukanlah berapa harga yang harus dibayar Andi!

Format Input

Satu baris berisi bilangan $N \ M \ X \ Y$

Format Output

Sebuah baris berisi jawaban harga yang harus dibayar Andi.

2.4

Percabangan

if-else

Percabangan

Percabangan merupakan ekspresi program yang akan memvalidasi suatu kondisi. Jika suatu kondisi terjadi (bernilai benar) maka perintah terkait akan dilaksanakan, dan jika kondisi itu terjadi (bernilai salah) memungkinkan program melakukan perintah lainnya.

Perfect if else statement



if-else

Jika maka. ..., jika tidak maka....

```
if (<kondisi>):
    <perintah dijalankan jika
    <kondisi> benar >
else:
    <perintah dijalankan jika
    <kondisi> salah >
```

<kondisi>

Bertipe datakan
Boolean

if-else

Jika maka. ..., jika tidak maka....

```
x = 9
if(x == 9):
    print("X sama dengan Sembilan")
else:
    print("X tidak sama dengan Sembilan")
```

if-elif

```
if(<kondisi1> :  
    <perintah dijalankan jika  
    <kondisi1> benar >  
elif(<kondisi2> :  
    <perintah dijalankan jika  
    <kondisi1> salah dan  
    <kondisi2> benar >
```

if-elseif

Jika maka. ..., jika tidak maka....

```
int x = 9;  
if(x >= 9):  
    print("X lebih dari sama dengan 9")  
elif(x>6):  
    print("x memenuhi 6<x<9")  
else:  
    print("x ≤ 6")
```

Nested if

```
if(<kondisi1>):
    if(<kondisi2>):
        <perintah dijalankan jika kondisi 1 dan 2 terpenuhi>
    else:
        <perintah dijalankan jika kondisi 1 terpenuhi tapi 2 tidak>

elif(<kondisi3>):
    if(<kondisi4>):
        ...
    else:
        ...
```

Clean Logical Nested If Logic Operator

```
if (<kondisi1> && <kondisi2>):  
    <perintah dijalankan jika kondisi  
    1 dan 2 terpenuhi>  
else:  
    <perintah dijalankan jika kondisi  
    1 atau 2 tidak terpenuhi>
```

Logic Operator

Logical Operators

Operator	Meaning	Example	Result
<code>&&</code>	Logical and	<code>(5<2)&&(5>3)</code>	False
<code> </code>	Logical or	<code>(5<2) !(5>3)</code>	True
<code>!</code>	Logical not	<code>!(5<2)</code>	True

Latihan Dulu Brow

Deskripsi Soal

Buatlah sebuah formulir biodata yang berisikan data :

Nama Depan, Nama Belakang, Nama Lengkap, Usia, Gender (L/P), Sudah Menikah / Belum (1 = Sudah, 0 = Belum).

Contoh Input

Nama Depan:Abdan

Nama Belakang:Hafidz

Usia:18

Gender:L

Sudah:0

Contoh Output

Seseorang Bernama Abdan Hafidz berusia 18 tahun adalah seorang laki – laki , dan belum menikah