

# **DASAR-DASAR PYTHON**



---

# DASAR-DASAR PYTHON

---

**Rolly M. Awangga**  
Informatics Research Center



**Kreatif Industri Nusantara**

***Penulis:***

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

***Editor:***

M. Yusril Helmi Setyawan

***Penyunting:***

Syafrial Fachrie Pane

Khaera Tunnisia

Diana Asri Wijayanti

***Desain sampul dan Tata letak:***

Deza Martha Akbar

***Penerbit:***

Kreatif Industri Nusantara

***Redaksi:***

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

***Distributor:***

Informatics Research Center

Jl. Sariasisih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara  
apapun tanpa ijin tertulis dari penerbit

*'Jika Kamu tidak dapat  
menahan lelahnya  
belajar, Maka kamu harus  
sanggup menahan  
perihnya Kebodohan.'*

*Imam Syafi'i*

## CONTRIBUTORS

---

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia



# CONTENTS IN BRIEF

---

<b>1 Java</b>	<b>1</b>
<b>2 <i>Java</i></b>	<b>3</b>
<b>3 Android</b>	<b>65</b>
<b>4 Fungsi dan Kelas</b>	<b>73</b>



# DAFTAR ISI

---

Daftar Gambar	xiii
Daftar Tabel	xv
Foreword	xix
Kata Pengantar	xxi
Acknowledgments	xxiiii
Acronyms	xxv
Glossary	xxvii
List of Symbols	xxix
Introduction	xxxii
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
<b>1 Java</b>	<b>1</b>
<b>2 Java</b>	<b>3</b>
2.1 Sejarah Java	3
2.2 Karakteristik Java	6

2.3	Komponen Utama <i>Platform Java</i>	7
2.4	<i>Variabel Java</i>	8
2.5	Versi-Versi Java	9
2.6	Fitur-Fitur Java	10
2.7	Pemanfaatan Teknologi Java	11
2.8	Kelebihan Java	15
2.9	Kekurangan java Java	16
2.10	Contoh Pemrograman Java Dasar	16
2.10.1	Menampilkan Teks pada Java	16
2.10.2	Menuliskan Komentar	17
2.10.3	Menggunakan Variabel	17
2.10.4	Menggunakan Tipe Data	17
2.10.5	Menggunakan Operator	18
2.10.6	Menggunakan String	18
2.10.7	Menggunakan Boolean	19
2.10.8	Menggunakan IF dan Else	19
2.10.9	Menggunakan While Loop	19
2.10.10	Menggunakan Break pada Java	20
2.10.11	Menggunakan Array	20
2.10.12	Menggunakan Method pada Java	21
2.10.13	Menggunakan Class / Object	21
2.10.14	Menggunakan Class Atribut	21
2.10.15	Menggunakan Class Method	22
2.10.16	Menggunakan Constructors	22
2.11	Memahami Konsep Dasar OOP pada Java	23
2.11.1	Kenapa Harus Belajar OOP?	23
2.12	Sejarah <i>Python</i>	23
2.13	Perbedaan <i>Python 2.x</i> dan <i>Python 3.x</i>	24
2.14	Installasi Python	34
2.14.1	Windows (Windows 10)	35
2.15	Instalasi Pip	51
2.15.1	Windows (Windows 10)	51
2.15.2	Linux (Ubuntu 19.04)	54
2.16	Setting Environment	55
2.16.1	Windows (Windows 10)	55
2.16.2	Linux (Ubuntu 19.04)	61
2.17	Command Line Interface/Interpreter	61
2.17.1	Windows (Windows 10)	61

2.17.2 Linux (Ubuntu 19.04)	63
<b>3 Android</b>	<b>65</b>
3.1 Sejarah Android	65
3.2 Pengertian Android	67
3.3 Varian Android	67
3.3.1 Android Murni atau Android Stock	67
3.3.2 While	68
3.4 Kondisi	68
3.5 Error	70
3.6 Try Except	71
<b>4 Fungsi dan Kelas</b>	<b>73</b>
4.1 Teori	73
4.1.1 Fungsi	73
4.2 Package	74
4.3 Class, Object, Atribute, and Method	74
4.4 Pemanggilan Class	75
4.5 Pemakaian Package Fungsi Apabila File Didalam Folder	75
4.6 Pemakaian Package Kelas Apabila File didalam Folder	75
Daftar Pustaka	77



## DAFTAR GAMBAR

---

2.1	Logo Java	4
2.2	Penemu Java James Gosling	5
2.3	Gambar Aplikasi Facebook	12
2.4	Gambar Aplikasi Opera Mini	13
2.5	Gambar Aplikasi Kamus Online	14
2.6	Gambar Aplikasi Game	15
2.7	Gambar hasil print	24
2.8	Gambar perintah print	25
2.9	Gambar hasil pembagian	26
2.10	Gambar perintah pembagian	26
2.11	Gambar hasil error	27
2.12	Gambar perintah error	28
2.13	Gambar hasil looping	29

2.14	Gambar perintah looping	30
2.15	Gambar hasil unicode (bytes)	31
2.16	Gambar perintah unicode (bytes)	32
2.17	Gambar hasil unicode	33
2.18	Gambar perintah unicode	34
2.19	Run Setup Anaconda	35
2.20	Setup Loading	36
2.21	Welcome to Anaconda Setup	37
2.22	<i>License Agreement</i>	38
2.23	<i>Just Me(recomended)</i>	39
2.24	<i>Pilih lokasi</i>	40
2.25	<i>Centang Anaconda to my PATH</i>	42
2.26	<i>Installation Complete</i>	44
2.27	<i>Installation Complete</i>	46
2.28	<i>Anaconda+JetBrains</i>	48
2.29	<i>Thanks for install Anaconda</i>	50
2.30	<i>Install pip</i>	52
2.31	<i>Install pip Selesai</i>	53
2.32	<i>Melihat Versi pip</i>	54
2.33	Gambar instal pip	55
2.34	<i>Properties</i>	56
2.35	<i>Advanced system settings</i>	57
2.36	<i>Environment Variables</i>	58
2.37	<i>Path</i>	59
2.38	<i>Edit Environment Variable</i>	60
2.39	Gambar setpath	61
2.40	<i>CLI in Command Prompt</i>	62
2.41	Gambar running script dengan CLI	63
3.1	Logo Android	66
3.2	Pendiri Android	66

## DAFTAR TABEL

---



## Listings

---



# **FOREWORD**

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini diciptakan bagi yang awam dengan java sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat*

*Februari, 2019*



## ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para mahasiswa dan dosen agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk dosen Pembimbinga Pak Syafrial Fahri Pane yang telah membimbing kami untuk membuat buku ini

R. M. A.



## ACRONYMS

---

ACGIH	American Conference of Governmental Industrial Hygienists
AEC	Atomic Energy Commission
OSHA	Occupational Health and Safety Commission
SAMA	Scientific Apparatus Makers Association



## GLOSSARY

---

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald



# SYMBOLS

---

$A$  Amplitude

$\&$  Propositional logic symbol

$a$  Filter Coefficient

$B$  Number of Beats



# INTRODUCTION

---

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[1].

$$ABC\mathcal{DEF}\alpha\beta\Gamma\Delta \sum_{def}^{abc} \quad (I.1)$$



# **BAB 1**

---

## **JAVA**

---



## BAB 2

---

# JAVA

---

### 2.1 Sejarah Java

Kopi Jawa *Java Coffee* terkenal bercita rasa tinggi dan salah satu jenis *Arabica* yang terbaik di dunia. Tetapi bagi James Gosling dan rekannya di *Sun Microsystems*, kopi yang diseduh di sebuah kafe *Peet* menjadi inspirasi untuk nama bahasa pemrograman komputer baru yang berhasil dikembangkan. *Java* menjadi pilihan menggantikan nama *Oak*, dari jenis pohon yang tumbuh di depan jendela ruang kerja Gosling. *Greentalk* adalah nama yang diperkenalkan Gosling pertama kali untuk bahasa pemrograman tersebut dengan file ekstensi ".gt" sebelum menjadi *Oak*.

Sayangnya nama *Oak* sudah dipakai perusahaan lain, yaitu *Oak Technology* sebagai merek dagang produknya. Usaha untuk mengganti nama ternyata tidak semudah yang dibayangkan. Atas usul pengacara dan ahli hukum perusahaan, perdebatan dengan berbagai pendapat dilakukan para insinyur, manajer pemasaran, penasehat hukum, dan direksi *Sun Microsystems*, untuk menemukan nama yang tepat selama berhari-hari.

Nama-nama yang kemudian menjadi kandidat adalah *Silk*, *DNA*, dan *Java*. Entah siapa yang pertama kali mengusulkan nama *Java* atau sejak kapan nama *Java* dipakai, tidak begitu diperhatikan karena alternatif pilihan nama tersebut dilakukan

secara kolektif. Kelak Kim Polese, manajer pemasaran saat itu yang sekarang adalah CEO *Marimba Inc.* Akhirnya memakai merek dagang *Java*.

Terciptanya *Java* berawal dari ambisi *Sun Microsystems* untuk menciptakan *platform universal* yang dapat mengintegrasikan berbagai mesin. Projek rahasia tersebut yang membawa misi besar itu diberi nama dengan *Green Project*. Projek tersebut melibatkan Patrick Naughton, Mike Sheridan, serta James Gosling dan kemudian dibantu oleh 13 orang staf. Mereka bekerja dengan tertutup dan mengasingkan diri pada sebuah gedung di *Sand Hill Road, Menlo Park, California, AS*. Projek yang dimulai pada Desember tahun 1990 akhirnya membuat hasil setelah bekerja keras kurang lebih selama 18 bulan dan menghabiskan dana jutaan *dolar AS*.



Gambar 2.1 Logo Java

Pada 3 September 1992 mereka mendemonstrasikan *Star7*, sebuah *PDA* dengan *input touchscreen* yang dapat menjalankan berbagai aplikasi interaktif. Termasuk menciptakan animasi Duke yang menjadi maskot *Java*. James Gosling serta kawan-kawannya telah mengantarkan bahasa pemrograman baru (*Java*) yang dapat berjalan pada semua platform peranti elektronika. Perbedaan *platform* diatasi dengan membuat mesin virtual pada arsitektur bahasa pemrograman yang baru. Mesin virtual tersebut akan menerjemahkan kode pemrograman menjadi bahasa yang dikenali mesin apa pun. *Java* juga dikenal sangat andal dan memiliki sistem keamanan sendiri.

*Java* hadir pada momentum yang tepat saat internet dan kebutuhan aplikasi multimedia mulai berkembang. James Gosling membuktikan kehebatan *Java* bersama John Gage, direktur *Sun Science Office* saat memberikan presentasi bertajuk "*Hollywood meets-Silicon-Valley*" di awal tahun 1995. Ia berhasil memperlihatkan gerakan molekul tiga dimensi di tengah-tengah layar komputer dengan menggerakkan *mouse*. Apalagi

sejak *HotJava* (sebelumnya disebut *WebRunner*) browser internet berbasis Java siap diluncurkan sebulan kemudian. Kerjasama antara *Sun Microsystems* dan *Netscape* untuk memasang *Java* pada browser *Netscape Communicator* saat dirilis kemudian ikut mempercepat ketenaran *Java*.

Sejak dirilis pada 23 Mei tahun 1995, *Java* segera melejit menjadi bahasa pemrograman yang favorit. *Java* menghasilkan gelombang baru dalam dunia komputasi. Apalagi *Sun Microsystem* memberikan *source code Java* dengan cuma-cuma melalui media internet. Dengan demikian *Java* segera tersebar dan orang-orang dapat mencoba dan memberikan umpan balik. Respon yang diberikan para pengguna *Java* ikut berkontribusi memperbaiki dari versi *alpha* (*1.0a2*) hingga versi 2 pada saat ini. Keberhasilan *Sun* ini menghadirkan *Java* sebagai yang terdepan dalam komunikasi internet tidak lepas dari peran James Gosling, arsitek bahasa pemrograman *Java*.

James Gosling lahir pada tanggal 19 Mei tahun 1956 dari t3 bersaudara di dekat *Calgary, Canada*. Sejak kecil dia memang sangat tertarik dengan elektronika. Saat dia berusia 12 tahun, orangtuanya mendapatinya berhasil membuat permainan *tic tac toe* dengan memanfaatkan komponen suku cadang dari telefon dan televisi. Melihat minat dan bakat tersebut, suatu ketika sahabat orangtuanya mengajak Gosling ke laboratorium komputer di *Universitas Calgary*. Saat itu usianya masih 14 tahun.



**Gambar 2.2** Penemu Java James Gosling

Sejak saat itulah ia lebih sering menghabiskan banyak waktu di laboratorium komputer daripada belajar di kelas. Lulus dari SMU, ia melanjutkan di Universitas *Calgary*. Saat menyelesaikan sarjana, ia mengembangkan editor teks *Emacs*, yang kelak menjadi editor teks yang paling banyak digunakan pada sistem operasi *Unix*. Kemudian ia mengambil pendidikan Master di Universitas *Alberta* Alberta sebelum melanjutkan program doktor di Universitas *Carnegie Mellon* di *Pittsburgh*. Ia memperoleh gelar Ph.D setelah berhasil mempertahankan tesisnya yang berjudul

"*The Algebraic Manipulation of Constraints*" pada tahun 1983. Ia segera bergabung dengan IBM selepas kuliah.

Sayang hasil pekerjaannya tidak pernah diproduksi. Setahun kemudian, ia bergabung dengan *Sun Microsystems* hingga menjadi bagian *Green Team* untuk menjalankan projek rahasia *Green Project*. Berkat kemampuannya, kariernya segera melejit sehingga menduduki posisi *Vice President (VP)* *Sun Microsystems* dan *Chief Technology Officer (CTO)* *Suns Developer Product*. Saat ini, ia masih berkontribusi pada *Real-Time Specification of Java* dan peneliti di laboratorium *Sun* untuk *software development tools*. Selain menjadi seorang arsitek dari *Java*, James juga membangun sistem akuisisi data satelit, *multiprosesor* untuk *Unix*, beberapa kompiler, *mail system* dan insinyur utama pembuat *windows manager NEWS (Network Extensible Windowing System)*.

## 2.2 Karakteristik Java

Terdapat beberapa karakteristik dari *Java*, diantaranya yaitu:

### 1. Sederhana (*Simple*)

Bahasa pemrograman *Java* menggunakan sintaks mirip dengan *C++* namun sintaks pada *Java* telah banyak diperbaiki terutama menghilangkan penggunaan pointer yang rumit dan *multiple inheritance*. Serta *Java* juga menggunakan *automatic memory allocation* dan *memory garbage collection*.

### 2. Berorientasi objek (*Object Oriented*)

*Java* menggunakan pemrograman berorientasi objek yang membuat program dapat dibuat secara modular dan dapat dipergunakan kembali. Pemrograman berorientasi objek memodelkan dunia nyata kedalam objek dan melakukan interaksi antar objek-objek tersebut.

### 3. Terdistribusi (*Distributed*)

*Java* dibuat untuk membuat aplikasi terdistribusi secara mudah dengan adanya *libraries networking* yang terintegrasi pada *Java*

### 4. Interpreted

Program *Java* dijalankan menggunakan *interpreter* yaitu *Java Virtual Machine (JVM)*. Hal ini menyebabkan *source code Java* yang telah dikompilasi menjadi *Java bytecodes* dapat dijalankan pada *platform* yang berbeda-beda.

### 5. Robust

*Java* mempunyai reliabilitas yang tinggi. *Compiler* pada *Java* mempunyai kemampuan mendeteksi *error* secara lebih teliti dibandingkan bahasa pemrograman lain. *Java* mempunyai *runtime-Exception handling* untuk membantu mengatasi *error* pada pemrograman.

### 6. secure

sebagai bahasa pemrograman untuk aplikasi internet dan terdistribusi, *Java* memi-

liki beberapa mekanisme keamanan untuk menjaga aplikasi tidak digunakan untuk merusak sistem komputer yang menjalankan aplikasi tersebut.

#### 7. ***Architecture Neutral***

Program Java merupakan *platform independent*. Dimana program hanya cukup mempunyai satu buah versi yang dapat dijalankan pada *platform* berbeda dengan *Java Virtual Machine*.

#### 8. ***Portable***

*Source code* maupun program pada Java dapat dengan mudah dibawa ke *platform* yang berbeda-beda tanpa harus melakukan kompilasi ulang.

#### 9. ***Performance***

*Performance* pada Java sering dikatakan kurang tinggi. Namun *performance* pada bahasa pemrograman Java dapat ditingkatkan menggunakan kompilasi Java lain seperti buatan *Inprise*, *Microsoft* maupun *Symantec* yang menggunakan *Just In Time Compilers (JIT)*

#### 10. ***Multithreaded***

Java juga mempunyai kemampuan untuk membuat suatu program yang dapat melakukan beberapa pekerjaan secara sekaligus dan simultan.

#### 11. ***Dynamic***

Java juga didesain untuk dapat dijalankan pada lingkungan yang dinamis. Perubahan pada suatu kelas dengan menambahkan *properties* ataupun *method* dapat dilakukan tanpa mengganggu program yang menggunakan kelas tersebut.

### 2.3 Komponen Utama *Platform Java*

*Platform*, juga dapat diartikan sebagai lingkungan perangkat keras atau perangkat lunak dimana program dijalankan. Pada momnya *platform* berdasarkan nama sistem operasi yang digunakan, misalnya *Windows 2000*, *Linux*, *Solaris*, atau *MacOS*. Untuk Java sendiri terdiri dari 2 komponen utama, yaitu:

#### 1. ***Java Virtual Machine (JVM)***

Sebuah mesin imajiner (maya) yang bekerja dengan menyerupai aplikasi pada sebuah mesin nyata. JVM menyediakan spesifikasi *hardware* dan *platform* dimana kompilasi kode Java terjadi. Spesifikasi inilah yang membuat aplikasi berbasis Java menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh JVM. Aplikasi program Java diciptakan dengan *file* teks berekstensi *.java*. Program ini dikompilasi menghasilkan satu berkas *bytecode* berekstensi *.class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

## 2. Java Application Programming Interface (JAVA API)

merupakan komponen-komponen dan kelas *java* yang sudah jadi, yang memiliki berbagai kemampuan. Kemampuan untuk menangani objek, string, angka, dsb. Pada API terdapat fungsi atau perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh para *programmer*. Java API terdiri dari tiga bagian utama:

- **Java Standard Edition (SE)**, sebuah standar API untuk merancang aplikasi *desktop* dan *applets* dengan bahasa dasar yang mendukung grafis, keamanan, konektivitas basis data dan jaringan.
- **Java Enterprise Edition (EE)**, sebuah inisiatif API untuk merancang aplikasi *server* dengan mendukung untuk basis data.
- **Java Micro Edition (ME)**, sebuah API untuk merancang aplikasi yang jalan pada alat kecil seperti telepon genggam, komputer genggam dan pager.

## 2.4 Variabel Java

Bahasa pemrograman pada umumnya, mengenal adanya *variabel* yang digunakan untuk menyimpan nilai atau data. Sedangkan *java* sendiri dikenal sebagai bahasa pemrograman dengan sifat *strongly typed* yang artinya diharuskan mendeklarasikan tipe data dari semua *variabel*, dan apabila lupa atau salah mengikuti aturan pendeklarasian *variabel*, maka akan mendapatkan *error* pada saat proses kompilasi. *Variabel* merupakan *container* yang digunakan untuk menyimpan suatu nilai pada sebuah program dengan tipe tertentu. Untuk mendefinisikan *variabel*, kita dapat menggunakan *identifier* untuk menamai *variabel* tersebut :

### 1. Identifier

*Identifier* adalah kumpulan karakter yang dapat digunakan untuk menamai *variabel*, *method*, *class*, *interface* dan *package*. Sebagaimana bahasa pemrograman pada umumnya, *java* memiliki peraturan untuk *identifier* yang *valid* atau sah. *identifier* dapat dikatakan *valid* atau sah jika diawali dengan :

- Huruf / abjad
- Karakter mata uang
- *Underscore*

*Identifier* tidak boleh mengandung @, spasi ataupun diawali dengan angka. Selain itu, *identifier* tidak boleh menggunakan *keyword* atau kata-kata yang memiliki arti atau digunakan dalam pemrograman *java*.

### 2. Deklarasi Variabel

Sintaks dasar :

[**tipe data**]

[**nama variabel**]

Menuliskan tipe data dari variabel, contoh :

```
int bilangan;  
char karakter;  
float bildenimal;  
boolean status;
```

Setelah mendeklarasikan variabel dengan tipe data, selanjutnya memberikan nilai variabel tersebut dengan tanda =

```
bilangan = 20;  
karakter = k;  
bildenimal = 22.2f;  
status = true;
```

Dapat juga mendeklarasikan dan memberikan nilai dalam satu baris.

```
int bilangan = 20;  
char karakter = k;  
float bildenimal = 22.2f;  
boolean status = true;
```

Kita dapat membuat variabel menjadi konstanta yang tidak dapat diubah lainnya dengan menambahkan keyword sebelum tipe data dari variabel. Contoh :

```
final int konstantainteger = 10;  
final float pajak = 15.5;
```

Agar konstanta ini dapat diakses oleh class lain tanpa harus membuat objek terlebih dahulu, maka kita dapat menambahkan modifier public dan keyword static seperti berikut :

```
public static final konstantainteger = 10;
```

## 2.5 Versi-Versi Java

Setelah sebelumnya dijelaskan mengenai pengertian, sejarah, karakteristik, dan komponen utama *platform java* maka selanjutnya akan membahas mengenai versi-versi pada *java*. Dari masa ke masa perkembangan *java* sudah semakin canggih dalam setiap fitur-fiturnya. Berikut ini adalah versi *java*, mulai dari yang pertama kali dirilis hingga versi terbaru yang akan dimunculkan :

1. Versi pertama *java* ditahun 1996 sudah merupakan versi rilis sehingga dinamakan *Java Versi 1.0*. *Java* versi ini menyertakan banyak paket standar awal yang terus dikembangkan pada versi selanjutnya:

- *java.lang*: Peruntukan kelas elemen-elemen dasar
- *java.io*: Peruntukan kelas *input* dan *output*, termasuk penggunaan berkas.

- *java.util*: Peruntukan kelas pelengkap seperti kelas struktur data dan kelas kelas penanganan.
  - *java.net*: Peruntukan kelas TCP/IP, yang memungkinkan berkomunikasi dengan komputer lain menggunakan jaringan TCP/IP.
  - *java.awt*: Kelas dasar untuk aplikasi antarmuka dengan pengguna (GUI)
  - *java.applet*: Kelas dasar aplikasi antar muka untuk diterapkan pada penjelajah web.
2. Pada tahun 1998, *Sun Microsystem* memperkenalkan *java* versi 1.2 atau lebih dikenal dengan nama *Java 2* yang terdiri atas *JDK* dan *JRE* versi 1.2. pada *Java 2* ini, dibagi menjadi 3 kategori yaitu :
- ***Java 2 Standart Edition (J2SE)*** Dimana pada kategori ini digunakan untuk menjalankan serta mengembangkan aplikasi-aplikasi *java* pada level *PC Personal Computer*
  - ***Java 2 Enterprise Edition (J2EE)*** Dimana pada kategori ini digunakan untuk menjalankan serta mengembangkan aplikasi-aplikasi *java* pada lingkungan enterprise dengan menambahkan sebuah fungsionalitas-fungsionalitas *java* semacam *EJB (Enterprise Java Bean)*, *Java CORBA*, *Servlet* dan *JSP* serta *Java XML (Extensible Markup Language)*.
  - ***Java 2 Micro Edition (J2ME)*** Dimana pada kategori ini digunakan untuk menjalankan serta mengembangkan sebuah aplikasi-aplikasi *java* pada *handled devices* atau perangkat-perangkat semacam *handphone*, *Palm*, *PDA*, dan *Pocket PC*. Dimana J2ME dirancang untuk dapat menjalankan program *java* pada perangkat-perangkat semacam *handphone* dan *PDA*, yang memiliki karakteristik yang berbeda dengan sebuah komputer biasa, misalnya kecilnya jumlah memori pada *handphone* dan *PDA*. J2ME terdiri atas komponen-komponen sebagai berikut *Java Virtual Machine (JVM)* dan *Java API (Application Programming Interface)* serta *Tools* lain untuk pengembangan aplikasi *java* semacam emulator *Java Phone* dan emulator *Motorolla*. Dalam J2ME dibagi menjadi dua bagian yang dikenal dengan istilah *configuration* dan *profile*

## 2.6 Fitur-Fitur Java

*Java* memiliki bermacam-macam fitur yang terbilang cukup berguna bagi para *programmer*. Fitur-fitur yang ada pada *java* diantaranya yaitu :

1. ***Java Virtual Machine (JVM)*** adalah sebuah mesin imajiner yang cara kerjanya menyerupai aplikasi pada sebuah mesin nyata. *JVM* menyediakan spesifikasi *hardware* dan *platform* kompilasi kode *Java* terjadi. Spesifikasi inilah yang membuat aplikasi berbasis *Java* menjadi bebas dari *platform* manapun karena proses kompilasi diselesaikan oleh *JVM*. Aplikasi program *Java* diciptakan dengan file teks berekstensi *.java*. Program ini dikompilasi menghasilkan

satu berkas *bytecode* berekstensi *class* atau lebih. *Bytecode* adalah serangkaian instruksi serupa instruksi kode mesin. Perbedaannya adalah kode mesin harus dijalankan pada sistem komputer dimana kompilasi ditujukan, sementara *bytecode* berjalan pada *java interpreter* yang tersedia di semua *platform* sistem komputer dan sistem operasi.

2. ***Garbage Collection*** Banyak bahasa pemrograman lain yang mengijinkan seorang *programmer* mengalokasikan memori pada saat dijalankan. Namun, setelah menggunakan alokasi memori tersebut, harus terdapat cara untuk menempatkan kembali blok memori tersebut supaya program lain dapat menggunakan-nya. Dalam *C*, *C++* dan bahasa lainnya, adalah *programmer* mutlak bertanggung jawab akan hal ini. Hal ini dapat menyulitkan bilamana *programmer* tersebut alpa untuk mengembalikan blok memori sehingga menyebabkan situasi yang dikenal dengan nama *memory leaks*. Pada program *Java* melakukan *garbage collection* ang dimana program tidak perlu menghapus sendiri objek-objek yang tidak digunakan lagi. Dimana pada fasilitas ini mengurangi beban pengelolaan memori oleh *programmer* dan mengurangi atau mengeliminasi sumber kesalahan terbesar yang terdapat pada bahasa yang memungkinkan alokasi dinamis.
  - (a) ***Code Security*** akan otomatis terimplementasi pada *java* melalui penggunaan *Java Runtime Environment* (JRE). *Java* menggunakan model pengamanan 3 lapis untuk melindungi sistem dari *untrusted Java Code*.
    - Pertama, *class-loader* menangani pemuatan kelas *Java* ke *runtime interpreter*. Proses ini menyediakan pengamanan dengan memisahkan kelaskelas yang berasal dari *local disk* dengan kelaskelas yang diambil dari jaringan. Hal ini membatasi aplikasi Trojan karena kelaskelas yang berasal dari *local disk* yang dimuat terlebih dahulu.
    - Kedua, *bytecode verifier* membaca *bytecode* sebelum dijalankan dan menjamin *bytecode* memenuhi aturanaturan dasar bahasa *java*.
    - Ketiga, manajemen keamanan menangani keamanan tingkat aplikasi dengan mengendalikan apakah program berhak mengakses sumber daya seperti sistem file, port jaringan, proses eksternal dan sistem windowing.

## 2.7 Pemanfaatan Teknologi Java

Bahasa pemrograman *Java* digunakan untuk membuat aplikasi, dekstop, web dan lainnya, sebagai mana dibuat dengan menggunakan bahasa pemrograman konvensional lainnya. Berikut ini adalah beberapa aplikasi yang merupakan hasil dari pemanfaatan teknologi *Java* :

### 1. Aplikasi Facebook

termasuk dari salah satu *social networking platform* yang paling populer. Jaringan seperti Facebook tidak hanya menghubungkan satu orang dengan orang

lainnya di dunia tanpa batas, akan tetapi menciptakan komunikasi atas profil, kegemaran, dan terutama pencitraan. Tidak mengherankan apabila jejaring sosial seperti Facebook sering dimanfaatkan untuk keperluan promosi, kampanye, bahkan propaganda. Tidak hanya untuk mengkomunikasikan citra individu, akan tetapi termasuk pula kelompok, organisasi, maupun produk. Di tahun 2008 lalu, Facebook bahkan mampu mengalahkan popularitas jejaring sosial ternama lainnya seperti *MySpace*, *Friendster*, dan *Multiply*. Siapapun bisa memanfaatkan ataupun mengoptimalkan Facebook Di sini disediakan dua buah *ebook* panduan menggunakan Facebook untuk pemula (*for Dummies*) dan tingkat lanjut (*advanced user*). Aplikasi online Facebook tidak hanya mampu mengkomunikasikan melalui pesan, akan tetapi mampu pula menciptakan komunikasi interaktif online. Di antara situs jejaring sosial, Facebook memiliki fitur yang paling lengkap. Pengguna yang memanfaatkan aplikasi jejaring online Facebook tidak hanya individu, akan tetapi kelompok, organisasi, dan bahkan suatu negara. Yang tentunya kita masih ingat ketika Israel pernah memanfaatkan Facebook (bersama dengan Twitter) untuk membela kepentingan mereka di Jalur Gaza.



**Gambar 2.3** Gambar Aplikasi Facebook

## 2. Aplikasi Opera Mini

adalah penjelajah web yang dirancang khusus untuk ponsel, dan juga untuk *PDA Phone* dan *Smartphone*. Penjelajah Web ini tersedia dalam *Java*, *Android*, *Windows Mobile*, *iOS*, *BlackBerry OS*, *UIQ3*, *Symbian*, dan *Zeebo*. Opera Mini adalah sebuah aplikasi yang gratis, didukung melalui kemitraan antara para pengembang, yaitu *Opera Software*, situs aplikasi gratis Gamejump, situs web Yahoo, dan Mesin Pencari Google. Opera Mini berasal dari penjelajah web Opera untuk komputer pribadi, yang telah tersedia untuk umum sejak 1996. Opera Mini pada awalnya dirancang untuk ponsel yang telah mampu terkoneksi atau tersambung ke jaringan internet. Opera Mini pertama kali diperkenalkan

pada tanggal 10 Agustus 2005 sebagai *pilot project* dalam kerjasama dengan stasiun televisi Norwegia, TV 2. Sehingga pada saat itu, Opera Mini hanya tersedia untuk pelanggan TV 2. Opera Mini terbaru mempunyai fitur yang tidak ada pada browser bawaan ponsel biasa yaitu mengupload file secara langsung. Opera Mini terkenal akan dengan kecepatan *browsingnya* yang cukup cepat, biasanya lebih cepat daripada browser bawaan ponsel. Keunggulan lainnya dari Aplikasi Opera Mini ini adalah tarifnya yang relatif lebih murah dari browser bawaan milik ponsel, dikarenakan Opera Mini dapat mengkompres suatu halaman. Keunggulan yang menarik ialah dapat menyimpan halaman website yang dianggap penting atau menarik oleh penggunanya, dan selanjutnya disimpan di folder yang telah dipilih pengguna pada memori telepon atau kartu memori. Opera Mini juga dapat menyinkronkan bookmark, dan tekan cepat ke akun milik anda di Komunitas Opera. Versi Opera Mini 6 menyertakan fitur Share yang memungkinkan pengguna membagikan halaman yang dibaca ke account jejaring sosial Facebook, Twitter, dan My Opera



**Gambar 2.4** Gambar Aplikasi Opera Mini

### 3. Kamus

sangat berguna dalam berbahasa, tahap pembelajaran kamus berperan penting dalam memahami kosa kata. Semangat belajar berbahasa tentu dilengkapi fasilitas yang lengkap diantaranya kamus tebal yang dibawa-bawa sepanjang waktu. Penggunanya tidak perlu repot lagi membawa kamus yang tebal, karena ada solusi jitu yang akan membantu untuk menyelesaikan masalah yang berhubungan dengan bahasa yang sulit untuk dimengerti. Karena dengan menggunakan aplikasi kamus bisa mengurangi kekipunan sebab tanpa kita sadari aplikasi kamus mengajak agar orang sedikit berfikir dan menganalisa tentang bahasa yang in-

gin di mengerti. Karena dengan berfikir dan menganalisa itu dapat mengurangi kepikunan dalam usia yang sudah lanjut usia. Sesuai dengan konsep belajar bahasa yang menyarankan agar kita praktik kapanpun dan dimanapun kita merasa senang untuk belajar.



**Gambar 2.5** Gambar Aplikasi Kamus Online

#### 4. Aplikasi Game

Bermain game juga ada manfaatnya. Manfaatnya yang selama ini banyak dirasakan adalah melatih kepekaan, melatih problem solving, berpikir logis dan kreatif, dan yang paling berpengaruh adalah secara tidak langsung para penggunanya belajar bahasa Inggris. Manfaat yang terakhir ini sungguh luar biasa, kita sebagai orang Indonesia sangat buruk dalam mempelajari bahasa Inggris tetapi dengan bermain game mau tidak mau kita harus belajar bahasa Inggris agar kita bisa memecahkan permasalahan yang ada didalam pemainan game dan bisa memenangkan game itu. Karena bahasa yang digunakan dalam game pada umumnya adalah bahasa Inggris, dan dialog, pengantar, tips, pilihan yang ada memakai bahasa Inggris, sehingga semakin sering kita bermain game, kita secara tidak langsung telah berlatih bahasa Inggris. Hasil penelitian menyebutkan bahwa para pemain game ini memiliki fokus yang lebih terhadap apa yang terjadi di sekelilingnya, jika dibandingkan dengan mereka-mereka yang jarang main game, apalagi yang tidak main sama sekali. Manfaat dari bermain game juga dapat dirasakan bagi orang yang sudah tua atau dengan kata lain lanjut usia karena dengan bermain game bisa mengurangi kepikunan sebab tanpa kita sadari game mengajak pemainnya agar sedikit berfikir dan menganalisa bagaimana cara menyelesaikan permain game. Karena dengan berfikir dan menganalisa itu dapat mengurangi kepikunan dalam usia yang sudah lansia. Ternyata maksud dari semua pembuat game dan kebanyakan programmer game adalah berusaha melatih kepekaan, problem solving, dan logika pemain ketika dihadap-

kan pada suatu keadaan yang kurang menguntungkan, dibumbui dengan kejutan-kejutan dan ketegangan yang memompa adrenalin dan emosi. Contohnya : *KONAMI*, *SILENT HILL*, dan juga *developer game-game* yang lain ingin men-gatakan bahwa dalam keadaan serba sulit, asal kita mau berusaha, pasti ada jalan keluarnya



**Gambar 2.6** Gambar Aplikasi Game

## 2.8 Kelebihan Java

Ada beberapa kelebihan yang dimiliki oleh *Java* dibanding dengan bahasa pemrograman yang lain, yaitu:

### 1. *Multiplatform*

Kelebihan utama dari *Java* ialah dapat dijalankan di beberapa *platform* atau sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program *Java* dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin atau *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa *platform* tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis *java* dikerjakan diatas *operating system Linux* tetapi dijalankan dengan baik di atas *Microsoft Windows*. *Platform* yang didukung sampai saat ini adalah *Microsoft Windows*, *Linux*, *Mac OS* dan *Sun Solaris*. Penyebanya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs *Java*) untuk meninterpretasikan *bytecode* tersebut.

### 2. *Object Oriented Programming*

OOP tersebut berarti bahwa semua aspek yang terdapat di *java* adalah Objek. *Java* merupakan salah satu bahasa pemrograman berbasis objek secara murni. Semua tipe data diturunkan dari kelas dasar yang disebut *Object*. Hal ini

sangat memudahkan pemrogram untuk mendesain, membuat, mengembangkan dan mengalokasi kesalahan sebuah program dengan basis *Java* sebagai salah satu bahasa pemograman termudah, bahkan untuk fungsi fungsi yang advance seperti komunikasi antar komputer sekalipun.

### 3. Perpustakaan Kelas yang Lengkap

*Java* terkenal dengan kelengkapan pada *library* atau perpustakaan (kumpulan program program yang disertakan dalam pemrograman *java*) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas *Java* yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi. Bergaya *C++*, memiliki sintaks seperti */C++/* sehingga menarik banyak pemrogram *C++* untuk pindah ke *Java*. Saat ini pengguna *Java* sangat banyak, sebagian besar adalah programmer dari *C++* yang pindah ke *java*. Universitas-universitas di Amerika juga mulai berpindah dengan mengajarkan *java* *Java* kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer. Dimana pada pengumpulan sampah otomatis, memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa *C++* yang dipakai secara luas).

## 2.9 Kekurangan *java Java*

Kekurangan dari *Java* adalah sebagai berikut :

1. *Java* memiliki kecepatan yang kurang dari *C++*
2. *Java* menggunakan atau memakan memori banyak
3. *Java* mudah didekompile yang dikarenakan kode jadi *Java* merupakan byte-code yang menyimpan banyak atribut bahasa tingkat tinggi, seperti *class*, *method*, dan *tipe data*.

## 2.10 Contoh Pemrograman *Java Dasar*

contoh-contoh dasar bahasa pemrograman *Java* yang bisa di praktikan menggunakan IDE Netbeans atau Eclipse

### 2.10.1 Menampilkan Teks pada *Java*

Kita bisa menampilkan teks pada *Java* menggunakan Kode System.out.println(). Contohnya seperti di bawah ini

```

1 public class MyClass {
2     public static void main(String[] args) {
3         System.out.println("Halo Polpos");
4     }
5 }
```

Hasilnya adalah :

```
1 Halo Polpos
```

## 2.10.2 Menuliskan Komentar

Untuk menambahkan komentar kita hanya perlu menambahkan tanda “//”

```

1 public class MyClass {
2     public static void main(String[] args) {
3         System.out.println("Halo Polpos"); // ini komentarnya
4     }
5 }
```

## 2.10.3 Menggunakan Variabel

Berikut ini merupakan contoh penggunaan variabel yaitu penggabungan nama depan dan belakang menggunakan bahasa pemrograman Java.

```

1 public class MyClass {
2     public static void main(String[] args) {
3         String firstName = "Ichsan";
4         String lastName = "Hizman";
5         String fullName = firstName + lastName;
6         System.out.println(fullName);
7     }
8 }
```

Maka Hasilnya adalah :

```
1 Ichsan Hizman
```

## 2.10.4 Menggunakan Tipe Data

Tipe data pada setiap bahasa pemrograman apapun sangat penting untuk dipelajari. Untuk contoh penggunaan Tipe Data pada Java yaitu :

```

1 public class MyClass {
2     public static void main(String[] args) {
3         int myNum = 16;           // integer (whole number)
4         float myFloatNum = 5.99f; // floating point number
5         char myLetter = 'E';     // character
6         boolean myBool = true;   // boolean
7         String myText = "Epi";   // String
8         System.out.println(myNum);
9         System.out.println(myFloatNum);
10        System.out.println(myLetter);
```

```

11     System.out.println(myBool);
12     System.out.println(myText);
13 }
14 }
```

Maka hasil yang akan ditampilkan seperti ini :

```

1 16
2 5.99
3 E
4 true
5 Epi
```

## 2.10.5 Menggunakan Operator

Contoh kode-kode di bawah yaitu menerapkan operator menggunakan bahasa pemrograman java.

```

1 public class MyClass {
2     public static void main(String[] args) {
3         int sum1 = 100 + 50;
4         int sum2 = sum1 + 250;
5         int sum3 = sum2 + sum2;
6         System.out.println(sum1);
7         System.out.println(sum2);
8         System.out.println(sum3);
9     }
10 }
```

Maka hasilnya adalah :

```

1 150
2 400
3 800
```

## 2.10.6 Menggunakan String

String yaitu tipe data yang digunakan untuk menyimpan barisan karakter. Untuk penggunaan string sendiri contohnya sebagai berikut.

```

1 public class MyClass {
2     public static void main(String[] args) {
3         String txt = "Ichsan Hizman";
4         System.out.println(txt.toUpperCase());
5         System.out.println(txt.toLowerCase());
6     }
7 }
```

Maka hasil yang akan ditampilkan adalah :

```

1 ICHSAN HIZMAN
2 ichsan hizman
```

## 2.10.7 Menggunakan Boolean

Boolean adalah struktur aljabar yang bisa kita gunakan untuk operasi logika seperti OR, AND, dan NOR. Contohnya yaitu :

```

1 public class MyClass {
2 public static void main(String [] args) {
3 int x = 15;
4 int y = 9;
5 System.out.println(x > y); // returns true , because 10 is higher than
6         9
7 }
```

Maka hasil yang ditampilkan adalah :

```
1 true
```

## 2.10.8 Menggunakan IF dan Else

Penggunaan IF dan Else pada pemrograman Java yaitu seperti dibawah ini:

```

1 public class MyClass {
2 public static void main(String [] args) {
3 if (25 > 18) {
4 System.out.println("25 is greater than 18"); // obviously
5 }
6 }
7 }
```

Hasilnya adalah :

```
1 25 is greater than 18
```

satu lagi contoh IF dan Else

```

1 public class MyClass {
2 public static void main(String [] args) {
3 int time = 20;
4 if (time < 18) {
5 System.out.println("Good day.");
6 } else {
7 System.out.println("Good evening.");
8 }
9 }
10 }
```

Hasilnya adalah :

```
1 Good evening
```

## 2.10.9 Menggunakan While Loop

Dibawah ini adalah contoh penggunaan While Loop pada pemrograman Java :

```

1 public class MyClass {
2 public static void main(String[] args) {
3 int i = 0;
4 while (i < 11) {
5 System.out.println(i);
6 i++;
7 }
8 }
9 }

```

Maka hasilnya adalah sebagai berikut :

```

1 0
2 1
3 2
4 3
5 4
6 5
7 6
8 7
9 8
10 9
11 10

```

## 2.10.10 Menggunakan Break pada Java

Break juga bisa kita gunakan pada Java. Contohnya yaitu :

```

1 public class MyClass {
2 public static void main(String[] args) {
3 for (int i = 0; i < 10; i++) {
4 if (i == 4) {
5 break;
6 }
7 System.out.println(i);
8 }
9 }
10 }

```

Maka hasil yang akan ditampilkan adalah :

```

1 0
2 1
3 2
4 3

```

## 2.10.11 Menggunakan Array

Array yaitu variabel yang bisa kita gunakan untuk menyimpan lebih dari satu data dengan tipe data yang sama. Contohnya seperti di bawah ini :

```

1 public class MyClass {
2 public static void main(String[] args) {
3 String[] cars = {"Mini Cooper", "BMW", "Suzuki", "Mazda"};

```

```
4 cars[0] = "Honda Jazz";
5 System.out.println(cars[0]);
6 }
7 }
```

Maka hasil yang akan ditampilkan adalah

```
1 Honda Jazz
```

### 2.10.12 Menggunakan Method pada Java

```
1 public class MyClass {
2 static void myMethod() {
3 System.out.println("Please Help me!");
4 }
5
6
7 public static void main(String[] args) {
8 myMethod();
9 myMethod();
10 myMethod();
11 }
12 }
```

Hasilnya adalah:

```
1 Please Help me!
```

### 2.10.13 Menggunakan Class / Object

Class / Object juga terdapat pada bahasa pemrograman Java. Contohnya adalah :

```
1 public class MyClass {
2 int x = 16;
3 public static void main(String[] args) {
4
5 MyClass myObj = new MyClass();
6 System.out.println(myObj.x);
7 }
8 }
```

Maka hasilnya adalah :

```
1 16
```

### 2.10.14 Menggunakan Class Atribut

Penggunaan class dan atribut pada Java contohnya yaitu :

```
1 public class MyClass {
2 int x;
3
4
5 public static void main(String[] args) {
6 MyClass myObj = new MyClass();
```

```

7 myObj.x = 100;
8 System.out.println(myObj.x);
9 }
10 }
```

Maka hasilnya adalah :

```
1 100
```

### 2.10.15 Menggunakan Class Method

Class method perlu kita pelajari agar kita bisa memahami pemrograman berorientasi objek. contohnya adalah dibawah ini :

```

1 public class MyClass {
2     static void myMethod() {
3         System.out.println("Hello Hizman");
4     }
5
6
7     public static void main(String[] args) {
8         myMethod();
9     }
10 }
```

Maka hasilnya adalah sebagai berikut:

```
1 Hello Hizman
```

### 2.10.16 Menggunakan Constructors

Contstructor merupakan method khusus pada pemrograman Java yang kemudian akan dieksekusi saat pembuatan objek. berikut adalah contohnya:

```

1 // Create a MyClass class
2 public class MyClass {
3     int x;
4
5     // Create a class constructor for the MyClass class
6     public MyClass() {
7         x = 16;
8     }
9     public static void main(String[] args) {
10        MyClass myObj = new MyClass();
11        System.out.println(myObj.x);
12    }
13 }
```

Maka hasilnya adalah :

```
1 16
```

## 2.11 Memahami Konsep Dasar OOP pada Java

OOP merupakan hal yang harus dipelajari untuk memahami Java lebih dalam. Karena pada dasarnya Java sendiri merupakan bahasa yang didesain untuk OOP.

Buktinya adalah :

```
1 class Hello {  
2     public static void main(String [] args){  
3         System.out.println("Hello World!");  
4     }  
5 }
```

OOP dikenal sebagai teknik pemrograman modern yang lebih efisien dan banyak digunakan pada Framework. Jika ingin belajar tentang Framework, maka pelajari dulu konsep dari OOP.

Apa saja yang akan kita pelajari pada OOP ini?

- Mengapa Harus Belajar OOP?
- Apa itu OOP?
- Objek dan Class
- Class Diagram

### 2.11.1 Kenapa Harus Belajar OOP?

Kenapa OOP itu penting?

Bukankah kita bisa bikin program dengan prosedural saja?

Yap tentu bisa

Tapi permasalahannya jika nanti programnya sudah semakin besar dan kompleks. Kode program akan sulit di-maintenance. Pada pemrograman prosedural, kita biasanya hanya memecah program menjadi beberapa prosedur. Lalu membuat variabel global dan lokal untuk menyimpan data.

## 2.12 Sejarah Python

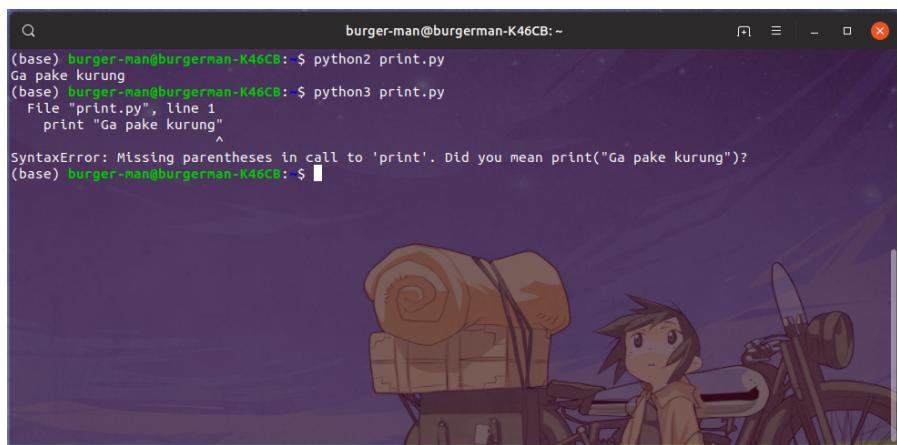
Python merupakan bahasa pemrograman tingkat tinggi yang dapat digunakan banyak hal, *Python* awalnya dirancang oleh **Guido van Rossum** pada tahun 1980 yang mana nama *Python* sebelum sebesar sekarang yaitu *ABC Programming Language* yang dijalankan di sistem operasi bernama *Amoeba Operating System*. Guido merasakan kehebatan dan kemampuan serta fitur yang berada pada bahasa pemrograman ABC ini sehingga Guido mengambil siktaks-sintaks yang berada pada bahasa pemrograman ABC ini, tentu saja banyak komplain yang berdatangan sehingga Guido terus melakukan perbaikan pada bahasa pemrograman yang sedang ia buat kala itu. Lalu, disinilah nama *Python* muncul dimuka bumi sebagai bahasa pemrograman, disaat Guido sedang menonton televisi dan menemukan kata '*Monty Python's Flying Circus*'.

Bahasa *Python* secara resmi dirilis pada tahun 1991, saat rilis pertama kali semua orang terkejut dengan sintaks yang dimiliki oleh *Python* ketika dibandingkan dengan bahasa lain seperti *Java*, *C*, *C++*, dan lain-lain pengekspresian bahasa ini cukup sederhana. Tujuan dari dibuatnya bahasa pemrograman ini adalah untuk mempermudah dalam membaca sebuah kode dari penulisan sintaks dan produktivitas dalam hal pengembangan tingkat lanjut.

## 2.13 Perbedaan *Python 2.x* dan *Python 3.x*

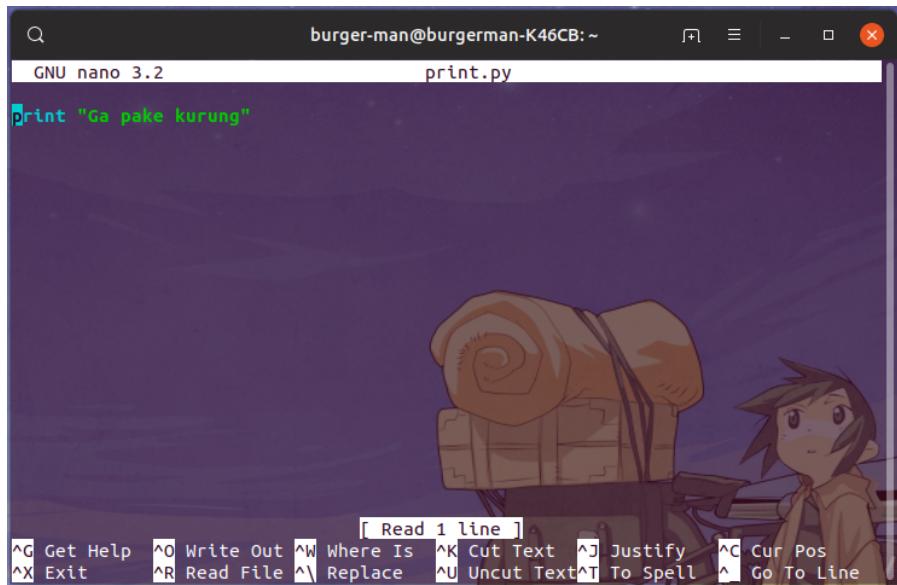
Banyak perbedaan yang akan kita temui jika kita dahulu pernah menggunakan *python* versi 2.x cukup lama sehingga berpindah ke versi 3.x, berikut contoh perbedaan pada *python* versi 2.x dan 3.x yang sangat penting untuk diketahui:

1. Perintah **print** Perbedaan perintah *print* pada dua versi ini adalah python 2.x tidak memakai kurung dan 3.x memakai kurung untuk perintah *print* bisa dilihat pada gambar 2.7 dan 2.8



```
(base) burger-man@burgerman-K46CB:~$ python2 print.py
Ga pake kurung
(base) burger-man@burgerman-K46CB:~$ python3 print.py
File "print.py", line 1
    print "Ga pake kurung"
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print("Ga pake kurung")?
(base) burger-man@burgerman-K46CB:~$
```

**Gambar 2.7** Gambar hasil print



The screenshot shows a terminal window titled 'burger-man@burgerman-K46CB:~'. The file 'print.py' is open in the nano editor. The code in the file is:

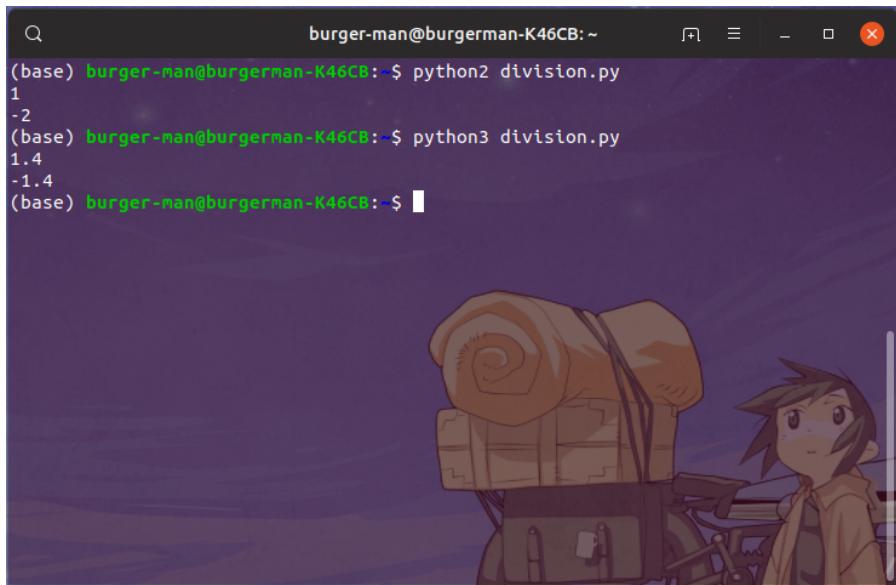
```
print "Ga pake kurung"
```

The terminal window has a dark background with a cartoon character of a boy with brown hair and a backpack. A menu bar at the bottom of the terminal window lists various nano editor commands:

- [ Read 1 line ]
- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut Text
- ^J Justify
- ^C Cur Pos
- ^X Exit
- ^R Read File
- ^| Replace
- ^U Uncut Text
- ^T To Spell
- ^ Go To Line

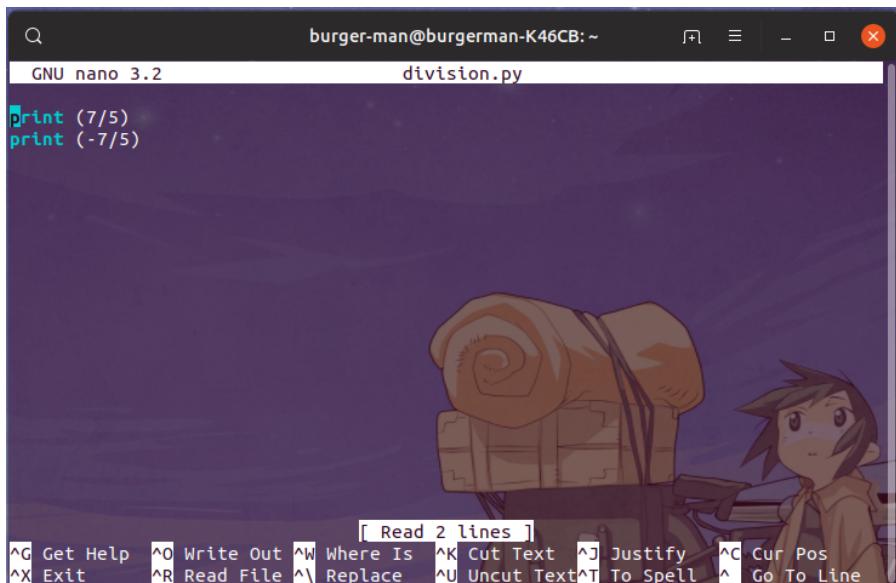
**Gambar 2.8** Gambar perintah print

2. Perintah pembagian **integer** Hasil dari perintah pembagian cukup jelas berbeda yang mana versi 2.x tidak secara mendetail untuk hasilnya sehingga angka yang dihasilkan bilangan **integer** sedangkan versi 3.x bertipe **float** perbedaannya bisa dilihat pada gambar 2.9 dan 2.10.



```
Q burger-man@burgerman-K46CB:~
(base) burger-man@burgerman-K46CB:~$ python2 division.py
1
-2
(base) burger-man@burgerman-K46CB:~$ python3 division.py
1.4
-1.4
(base) burger-man@burgerman-K46CB:~$
```

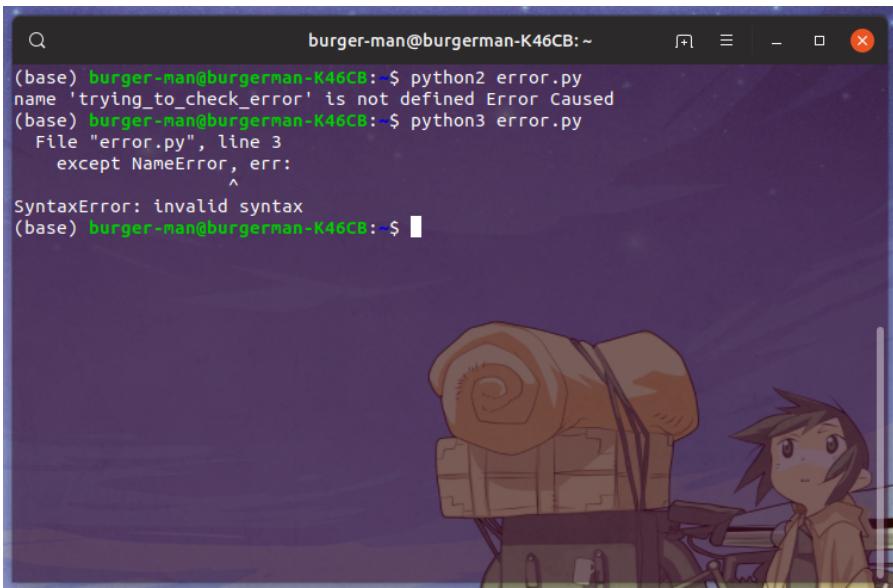
Gambar 2.9 Gambar hasil pembagian



```
Q burger-man@burgerman-K46CB:~ division.py
GNU nano 3.2
print (7/5)
print (-7/5)
```

Gambar 2.10 Gambar perintah pembagian

3. **Try and Except** Perbedaan pada *try and except* hanya berbeda di penggunaan , untuk versi 2.x dan as untuk versi 3.x.



```
burger-man@burgerman-K46CB:~
(base) burger-man@burgerman-K46CB:~$ python2 error.py
name 'trying_to_check_error' is not defined Error Caused
(base) burger-man@burgerman-K46CB:~$ python3 error.py
File "error.py", line 3
    except NameError, err:
          ^
SyntaxError: invalid syntax
(base) burger-man@burgerman-K46CB:~$
```

**Gambar 2.11** Gambar hasil error

```
try:    trying_to_check_errorexcept NameError, err:    print err, 'Error Caused'
```

^G Get Help ^O Write Out ^W Where Is ^K Cut  
^X Exit ^R Read File ^\ Replace ^U Unc

Gambar 2.12 Gambar perintah error

4. **Looping** Perbedaan pada looping hanya saja versi 3.x tidak bisa menggunakan sintaks **xrange** lagi.

Q

burger-man@burgerman

```
(base) burger-man@burgerman-K46CB:~$ python2 l
1
2
3
4
1
2
3
4
(base) burger-man@burgerman-K46CB:~$ python3 l
Traceback (most recent call last):
  File "loop.py", line 1, in <module>
    for x in xrange(1, 5):
NameError: name 'xrange' is not defined
(base) burger-man@burgerman-K46CB:~$ █
```

Gambar 2.13 Gambar hasil looping



The screenshot shows a terminal window titled "GNU nano 3.2" with the file name "loop.py" in the title bar. The terminal contains two lines of Python code:

```
for x in xrange(1, 5):
    print(x)

for x in range(1, 5):
    print(x)
```

At the bottom of the terminal, there is a menu bar with the following options:

- ^G Get Help
- ^O Write Out
- ^W Where Is
- ^K Cut
- ^X Exit
- ^R Read File
- ^\\ Replace
- ^U Uncu [ Cancelled ]

Gambar 2.14 Gambar perintah looping

5. **Unicode** Unicode ini cukup penting karena kita mengetahui bagaimana setiap versi merespons setiap unicode yang diberikan.



burger-man@burgerma

```
(base) burger-man@burgerman-K46CB:~$ python2 u
<type 'str'>
<type 'str'>
(base) burger-man@burgerman-K46CB:~$ python3 u
<class 'str'>
<class 'bytes'>
(base) burger-man@burgerman-K46CB:~$ █
```

**Gambar 2.15** Gambar hasil unicode (bytes)



burger-man@burgerma ~

GNU nano 3.2

unicode.p

```
print(type('default string '))
print(type(b'string with b '))
```

[ Read 2 lines

^G Get Help	^O Write Out	^W Where Is	^K Cut
^X Exit	^R Read File	^\\ Replace	^U Uncu

Gambar 2.16 Gambar perintah unicode (bytes)

Pada gambar 2.15 terlihat jelas bahwa perintah **bytes** hanya direspon pada versi 3.x sedangkan versi 2.x merespon **string**



```
Q burger-man@burgerman-K46CB:~$ python2
<type 'str'>
<type 'unicode'>
(base) burger-man@burgerman-K46CB:~$ python3
<class 'str'>
<class 'str'>
(base) burger-man@burgerman-K46CB:~$ █
```

**Gambar 2.17** Gambar hasil unicode

```
GNU nano 3.2          unicode.py

print(type('default string '))
print(type(u'string with u '))

[ Read 2 lines]
```

^G Get Help ^O Write Out ^W Where Is ^K Cut  
^X Exit ^R Read File ^\ Replace ^U Uncu

Gambar 2.18 Gambar perintah unicode

## 2.14 Installasi Python

Untuk installasi kali ini akan bagi menjadi dua sistem operasi yaitu Windows (Windows 10), dan Linux (Ubuntu 19.04). Installasi menggunakan *environment* Anaconda sebagai installasi *python*. Anaconda merupakan *environment open-source* un-

tuk bahasa pemrograman *Python*, dan *R* berfungsi untuk memanajemen penggunaan *package* pada *python* dan *R*.

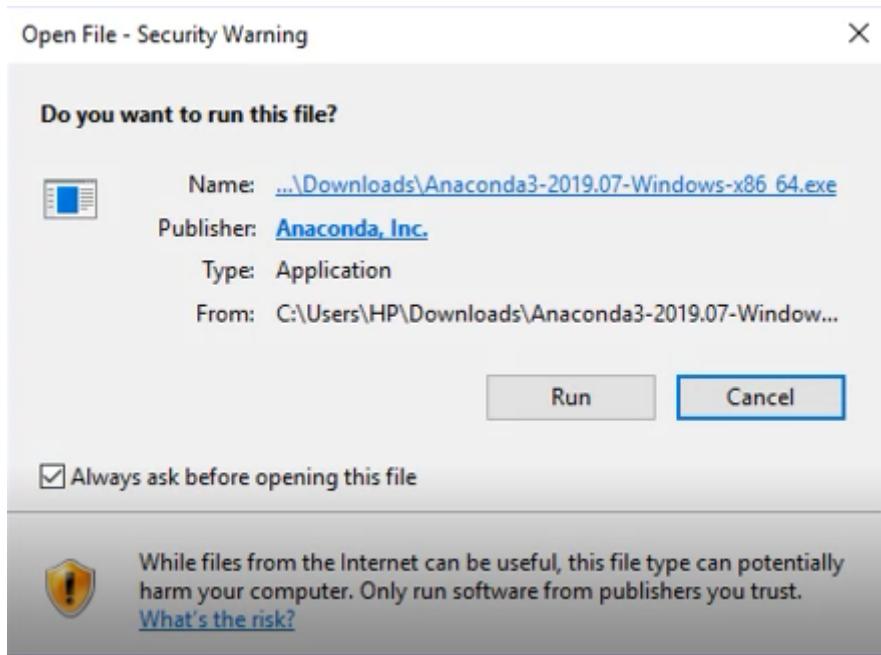
### 2.14.1 Windows (Windows 10)

Hal yang harus diperhatikan sebelum melakukan instalasi *Anaconda Python*

1. Perhatikan versi dari sistem operasi yang digunakan (versi 32bit atau 64bit)
2. Download file anaconda yang sesuai dengan versi sistem operasi (32bit atau 64bit)
3. *Download Anaconda Python* <https://www.anaconda.com/distribution/>

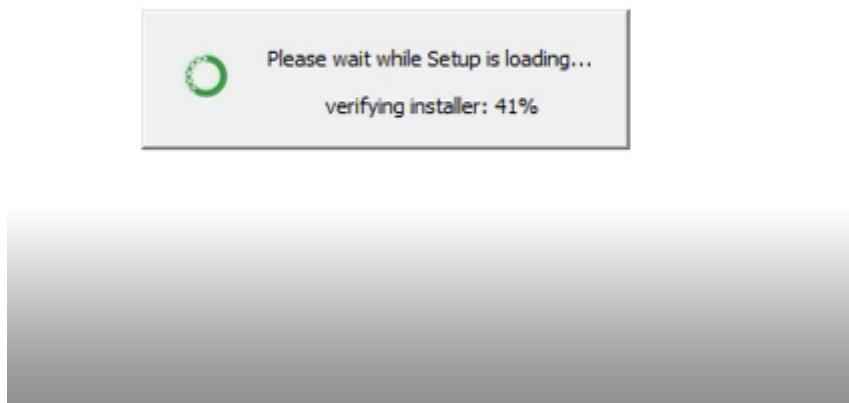
Berikut langkah-langkah instalasi anaconda.

1. Buka aplikasi *installer Anaconda* tersebut lalu akan muncul gambar *installer anaconda*.



Gambar 2.19 Run Setup Anaconda

2. Tunggu hingga *setup loading* selesai



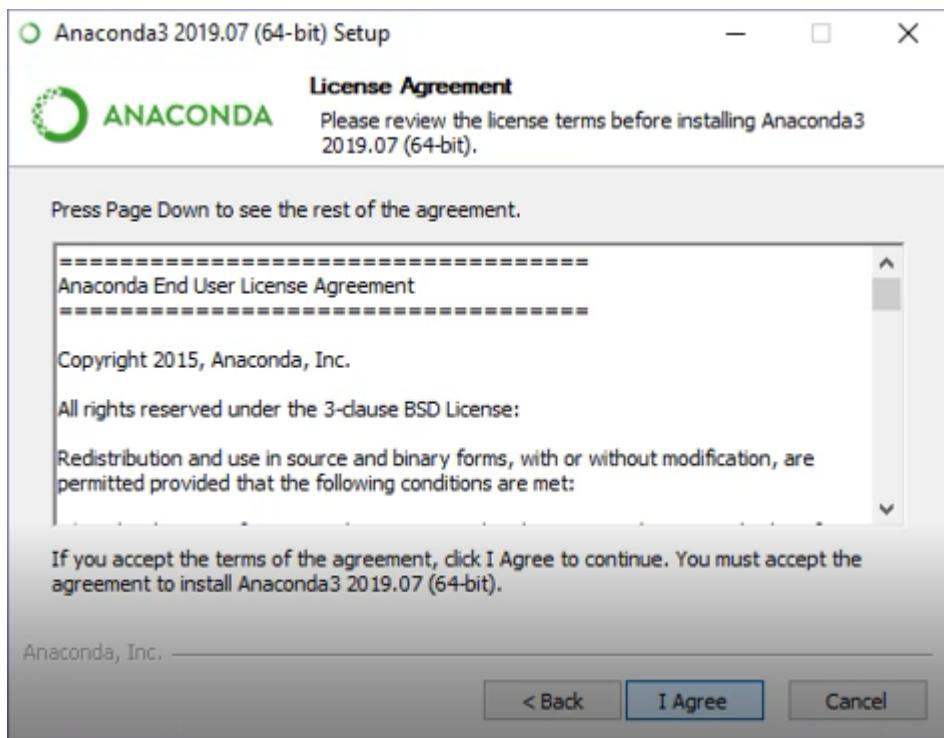
**Gambar 2.20** Setup Loading

3. Jika *setup loading* telah selesai, maka klik *next*



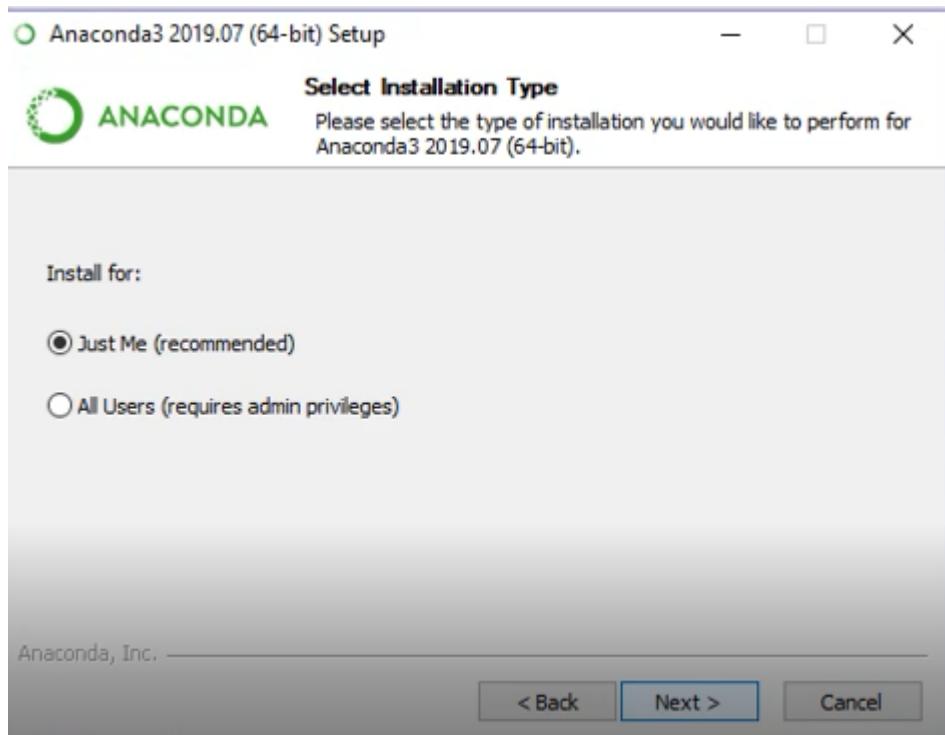
**Gambar 2.21** Welcome to Anaconda Setup

4. Pada *License Agreement* klik *I Agree* gambar *License Agreement*.



**Gambar 2.22 License Agreement**

5. Kemudian pilih *Just Me(Recomended)* agar sesuai dengan komputer yang digunakan, kemudian klik *next* gambar *Just Me(recomended)*.



**Gambar 2.23** Just Me(recomended)

6. Kemudian pilih lokasi tempat *menginstall anaconda* gambar *Pilih lokasi*.

## ○ Anaconda3 2019.07 (64-bit) Setup



ANACONDA

**Advanced Installation**

Customize how Anaconda

Setup will install Anaconda3 2019.07 (64-bit) in the default folder, click Browse and select another folder. Click Next to continue.

### Destination Folder

C:\Users\HP\Anaconda3

Space required: 2.9GB

Space available: 65.0GB

Anaconda, Inc.

7. Kemudian centang *Add Anaconda to my Path environment variable*, agar saat menginstall selenium langsung ke path anaconda tidak ke aplikasi yang lain. Klik *install* gambar *Centang Anaconda to my PATH*.

## ○ Anaconda3 2019.07 (64-bit) Setup



**ANACONDA**

**Advanced Installation**

Customize how Anaconda

### Advanced Options

Add Anaconda to my PATH environment

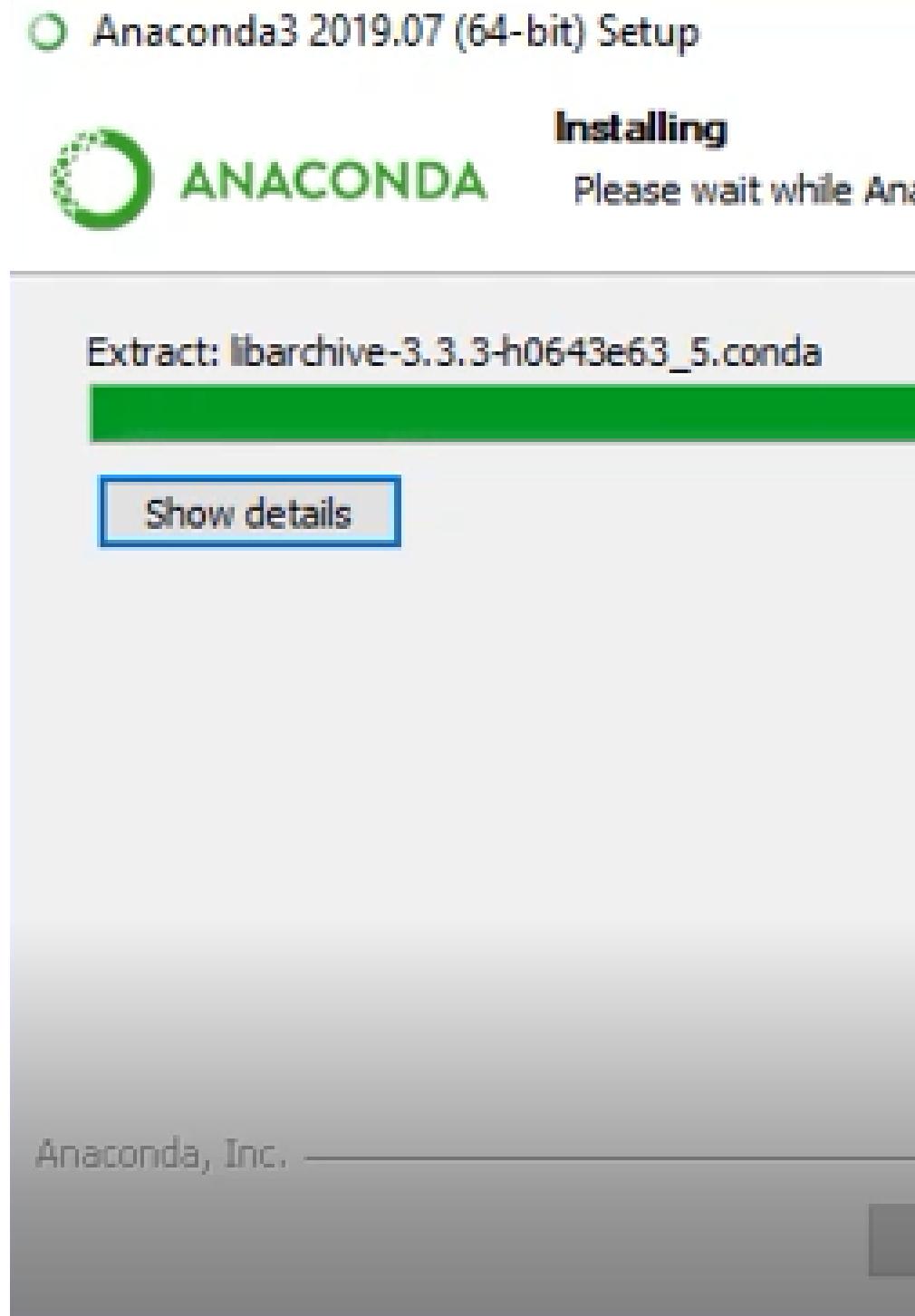
Not recommended. Instead, open Anaconda menu and select "Anaconda (64-bit)". This "Anaconda get found before previously installed cause problems requiring you to uninstall an

Register Anaconda as my default Python

This will allow other programs, such as PyCharm, Wing IDE, PyDev, and MSI binary detect Anaconda as the primary Python 3.7

Anaconda, Inc. —

8. Tunggu sampai proses *installasi* selesai gambar *Installation Complete*.



Gambar 2.26 Installation Complete

9. Apabila instalasi telah selesai klik *next*

## ○ Anaconda3 2019.07 (64-bit) Setup



ANACONDA

Installation Complete

Setup was completed

Completed

Processed C:\Users\HP\Anaconda3\Menu\notebo

Processed C:\Users\HP\Anaconda3\Menu\power

Processed C:\Users\HP\Anaconda3\Menu\spyder

Execute: "C:\Users\HP\Anaconda3\pythonw.exe

Running post install...

Execute: "C:\Users\HP\Anaconda3\pythonw.exe

Execute: "C:\Users\HP\Anaconda3\pythonw.exe

Execute: "C:\Users\HP\Anaconda3\pythonw.exe

Created uninstaller: C:\Users\HP\Anaconda3\Un

Completed

Anaconda, Inc. -

10. klik *next*

## ○ Anaconda3 2019.07 (64-bit) Setup



ANACONDA

Anaconda3 2019.07

Anaconda + JetBrains

Anaconda and JetBrains are working together to bring environments tightly integrated in the PyCharm IDE.

PyCharm for Anaconda is available at:

<https://www.anaconda.com/pycharm>



ANACONDA

Anaconda, Inc.

11. Jika sudah klik *finish* gambar *Thanks fo install Anaconda*.

## ○ Anaconda3 2019.07 (64-bit) Setup



Thanks for choosing  
Anaconda!

Anaconda is the most popular Python distribution.

Share your notebooks and data science projects  
on Anaconda Cloud.

[Learn more about Anaconda](#)

[Learn how to get started](#)

## 2.15 Instalasi Pip

### 2.15.1 Windows (Windows 10)

1. buka anaconda promt

2. ketikkan conda install -c anaconda pip

## Anaconda Prompt (Anaconda3)

```
(base) C:\Users\trian>conda install  
Collecting package metadata (current environment)  
Solving environment: done
```

```
--> WARNING: A newer version of conda is available:  
    current version: 4.7.10  
    latest version: 4.7.12
```

Please update conda by running

```
$ conda update -n base -c defaults
```

```
## Package Plan ##
```

```
environment location: C:\Users\trian>
```

```
added / updated specs:  
- pip
```

3. ketik y, lalu enter. Tunggu hingga proses instalasi selesai.

■ Anaconda Prompt (Anaconda3)

```
Total: 10.9 MB
```

```
The following packages will be UPDATED:
```

```
  conda           pkgs/main::conda-4.7.10-py37_0 --> anacon
```

```
The following packages will be SUPERSEDED by a higher-priority channel
```

```
  ca-certificates      pkgs/main --> anacon
  certifi              pkgs/main --> anacon
  openssl              pkgs/main --> anacon
  pip                  pkgs/main --> anacon
```

```
Proceed ([y]/n)? y
```

```
Downloading and Extracting Packages
```

```
openssl-1.1.1c      | 5.7 MB    | #####
certifi-2019.6.16   | 155 KB   | #####
ca-certificates-2019 | 166 KB   | #####
pip-19.1.1          | 1.8 MB    | #####
conda-4.7.12        | 3.0 MB    | #####
```

```
Preparing transaction: done
```

```
Verifying transaction: done
```

```
Executing transaction: done
```

```
(base) C:\Users\trian>
```

**Gambar 2.31** *Install pip Selesai*

4. jika telah selesai, lakukan pengecekan versi pip dengan mengetikkan pip -V

```

Anaconda Prompt (Anaconda3)
help                                         Show help for commands.

General Options:
-h, --help                                     Show help.
--isolated                                    Run pip in an isolated mode, ignoring environment variables.
-v, --verbose                                   Give more output. Option is additive, and can be combined with -q.
-V, --version                                   Show version and exit.
-q, --quiet                                     Give less output. Option is additive, and can be combined with -v.
--log <path>                                   Path to a verbose appending log.
--proxy <proxy>                                 Specify a proxy in the form [user:password@]host[:port].
--retries <retries>                             Maximum number of retries each connection attempt.
--timeout <sec>                                Set the socket timeout (default 15 seconds).
--exists-action <action>                         Default action when a path already exists: (a)bort,
                                                (d)elete, or (r)ename.
--trusted-host <hostname>                      Mark this host as trusted, even though it's not in the system's trust store.
--cert <path>                                    Path to alternate CA bundle.
--client-cert <path>                            Path to SSL client certificate, a single certificate in PEM format.
--cache-dir <dir>                               Store the cache data in <dir>.
--no-cache-dir                                 Disable the cache.
--disable-pip-version-check                     Don't periodically check PyPI to determine if there is a newer version available for download. Implied with --no-index.
--no-color                                      Suppress colored output

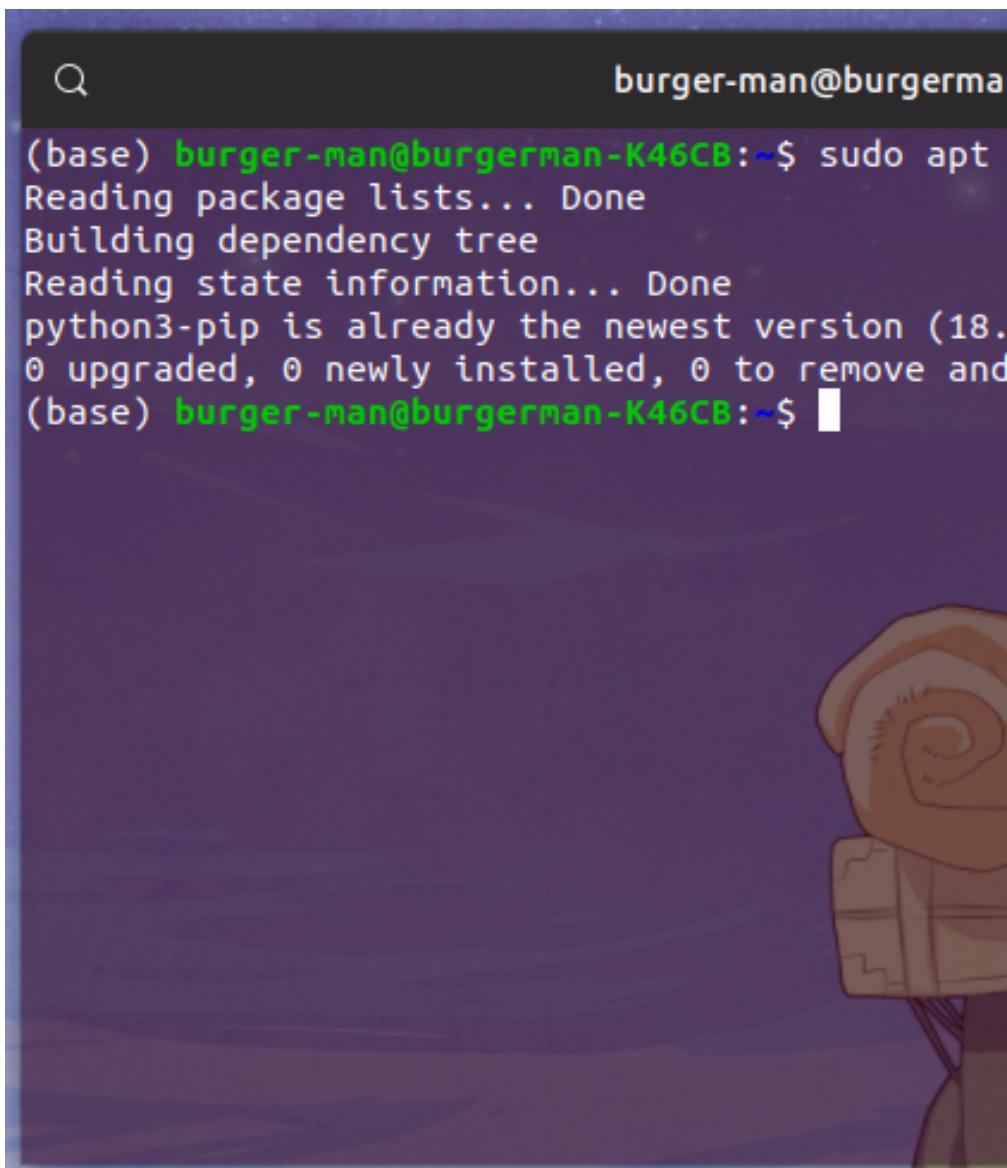
(base) C:\Users\trian>pip -V
pip 19.1.1 from C:\Users\trian\Anaconda3\lib\site-packages\pip (python 3.7)
(base) C:\Users\trian>

```

**Gambar 2.32 Melihat Versi pip**

### 2.15.2 Linux (Ubuntu 19.04)

1. pertama kita buka terminal kita lalu ketikkan perintah **sudo apt install python3-pip -y** untuk pip3 dan **sudo apt install python-pip -y** untuk pip contoh seperti gambar 2.33, lalu enter



```
(base) burger-man@burgerman-K46CB:~$ sudo apt update  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
python3-pip is already the newest version (18.0  
0 upgraded, 0 newly installed, 0 to remove and  
(base) burger-man@burgerman-K46CB:~$ █
```

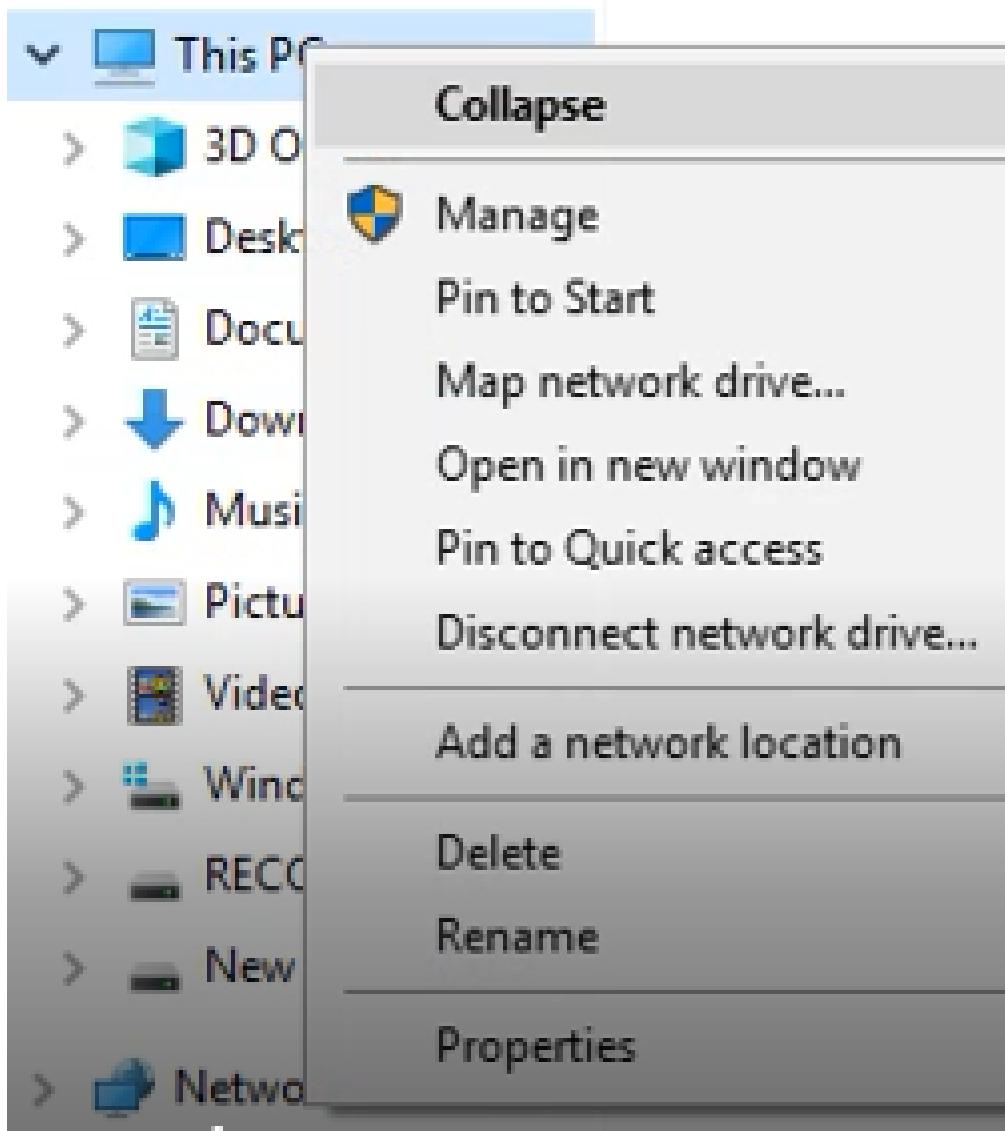
**Gambar 2.33** Gambar instal pip

## 2.16 Setting Environment

### 2.16.1 Windows (Windows 10)

1. Buka file explorer

2. Klik kanan pada This pc, lalu pilih properties



Gambar 2.34 Properties

3. Pilih menu Advanced system settings

-  Device Manager
-  Remote settings
-  System protection
-  Advanced system settings

**Gambar 2.35** *Advanced system settings*

## System Properties

Computer Name

Hardware

Advanced

System

You must be logged on as an Administrator to make changes.

### Performance

Visual effects, processor scheduling, memory usage

### User Profiles

Desktop settings related to your sign-in

### Startup and Recovery

System startup, system failure, and debugging info

## 5. Pilih Path

**Environment Variables****User variables for HP**

Variable	Value
OneDrive	C:\Users\HP
Path	C:\Users\HP
QT_DEVICE_PIXEL_RATIO	auto
TEMP	C:\Users\HP
TMP	C:\Users\HP

**Gambar 2.37 Path**

6. lalu pilih environment variable yang ingin ditambahkan, klik OK

## Edit environment variable

C:\Users\HP\Anaconda3

C:\Users\HP\Anaconda3\Library\mingw-w64\bin

C:\Users\HP\Anaconda3\Library\usr\bin

C:\Users\HP\Anaconda3\Library\bin

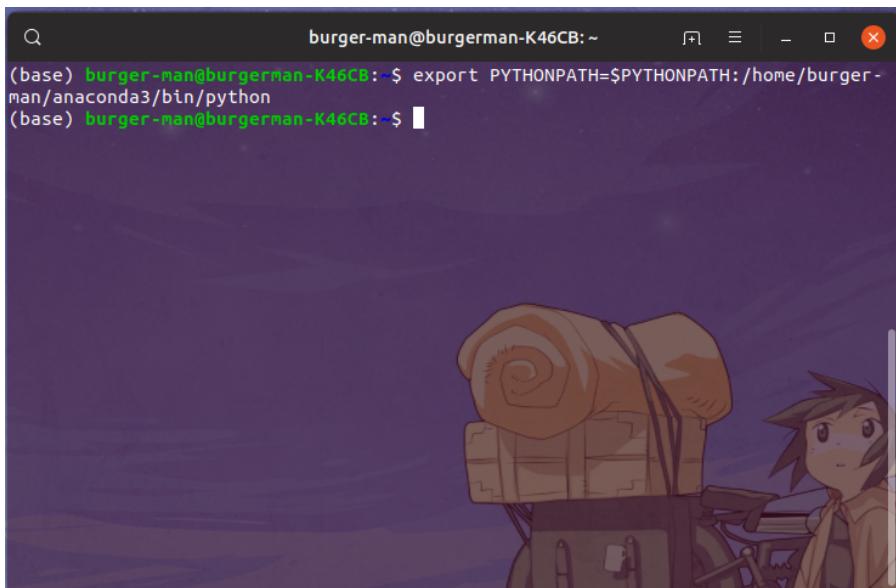
C:\Users\HP\Anaconda3\Scripts

%USERPROFILE%\AppData\Local\Microsoft\Windows\

C:\Users\HP\AppData\Local\Programs\MiK

## 2.16.2 Linux (Ubuntu 19.04)

1. pertama kita buka terminal kita lalu ketikkan perintah export PYTHONPATH=\$PYTHONPATH  
contoh seperti gambar 2.39, lalu enter



Gambar 2.39 Gambar setpath

## 2.17 Command Line Interface/Interpreter

### 2.17.1 Windows (Windows 10)

1. Buka command prompt lalu ketikkan python
2. Buatlah perintah print, input, perkalian, dan pembagian
3. Bisa juga menjalankan file .py yang telah dibuat di IDE dengan cara python namafile.py, lalu klik enter

## ④ Command Prompt

```
Microsoft Windows [Version 10.0.17763.112]
(c) 2018 Microsoft Corporation. All rights reserved.
```

```
C:\Users\trian>python
```

```
Python 3.7.3 (default, Apr 24 2019, 15:37:01)
```

```
Warning:
```

```
This Python interpreter is in a conda environment which  
has been activated. Libraries may not have been  
fully loaded. To activate a different environment  
please see https://conda.io/activate-environment.
```

```
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> print("Salah Apa Aku")
```

```
Salah Apa Aku
```

```
>>> input("Masukkan Nama Anda: ")
```

```
Masukkan Nama Anda: Dinda
```

```
'Dinda'
```

```
>>> 3*2
```

```
6
```

```
>>> 3/2
```

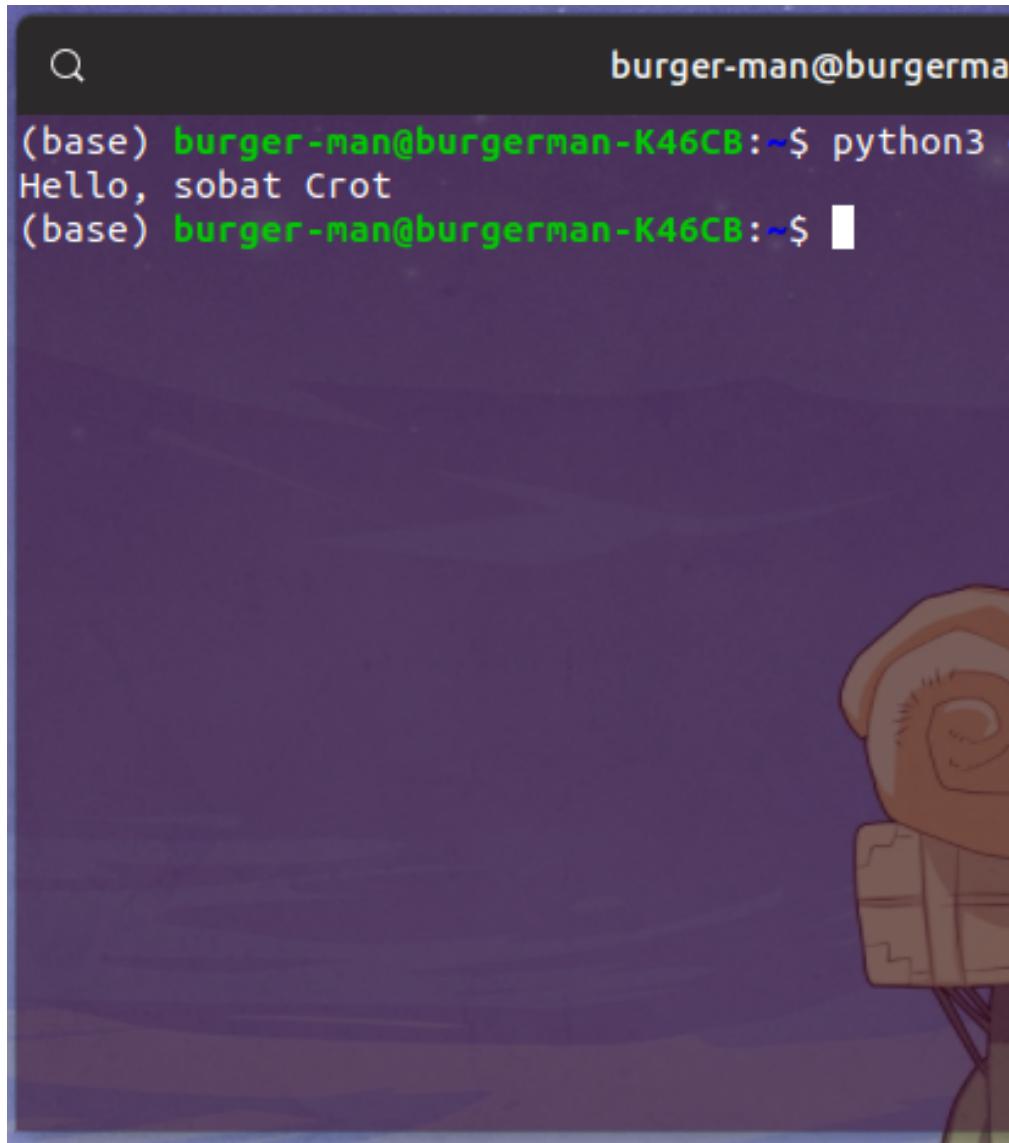
```
1.5
```

```
>>> exit()
```

### 2.17.2 Linux (Ubuntu 19.04)

Untuk menjalankan perintah CLI cukup mudah yaitu sebagai berikut

1. Buka terminal lalu ketikkan **python namafile.py** seperti gambar 2.41, lalu enter



The screenshot shows a terminal window with a dark purple background. In the top right corner, there is a small cartoon illustration of a character's head with a large ear and a brown cap. The terminal window has a dark grey header bar with a magnifying glass icon on the left and the user information "burger-man@burgerma" on the right. The main body of the terminal is white and contains the following text:

```
(base) burger-man@burgerman-K46CB:~$ python3  
Hello, sobat Crot  
(base) burger-man@burgerman-K46CB:~$
```

**Gambar 2.41** Gambar running script dengan CLI



## BAB 3

---

# ANDROID

---

### 3.1 Sejarah Android

Andy Rubin, Chris White, dan Nick Sears merupakan pencipta android, yang awalnya mereka ingin menciptakan perangkat kamera digital. Kemudian dialihkan penggunaannya menjadi smartphone. Tahun 2007 Android dirilis secara resmi oleh google yang telah membelinya pada tahun 2005 dengan suntikan dana 50 juta dolar. Setelah mendapat dana dari google, para pendiri android kemudian bergabung dengan google dan membuat sebuah protipe. Pada 2006, prototipe pertama yang dibuat yaitu Blackberry, yaitu smartphone yang memiliki keyboard.

Setahun kemudian, perusahaan apple merilis smartphone dengan keyboard layar sentuh pertama, yaitu iPhone. Dengan demikian, google mulai mengembangkan smartphone yang dia miliki dengan menggunakan keyboard layar sentuh agar dapat menyaingi iPhone. Setelah lama bersaing, pada 2010 merupakan tahun dimana android pertama kalinya mengambil alih pangsa pasar yang sebelumnya diambil alih oleh apple.



**Gambar 3.1** Logo Android

Mengapa apple bisa kalah dari android yaitu karena ketersediaan dan harga android yang murah membuat apple kalah dalam pasar smartphone. Selain itu android juga memiliki sistem operasi yang open source. Dengan itu banyak perusahaan smartphone yang bekerja sama dengan android. Android merupakan sistem yang berbasis linux yang dirancang untuk perangkat layar sentuh seperti smartphone dan komputer tablet.

Android merupakan sistem operasi dengan sumber terbuka, dan Google juga merilis kodennya di bawah Lisensi Apache. Kode dengan sumber terbuka dan juga lisensi perizinan pada Android, memungkinkan perangkat lunak dapat dimodifikasi secara bebas dan dapat didistribusikan dengan mudah. Android juga memiliki komunitas pengembang aplikasi yang memperluas fungsionalitas perangkat, umumnya ditulis dalam bahasa pemrograman Java.



**Gambar 3.2** Pendiri Android

Pada Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk sistem operasi Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android merupakan platform smartphone paling populer bagi para pengembang, dan digunakan sekitar 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan ada lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013.

Pada November 2013 juga, Android padat menguasai pangsa pasar telepon pintar global, produk Samsung yang merupakan pemimpin tingkat penjualan paling tinggi, dengan persentase 64% pada bulan Maret 2013. Pada Juli 2013, terdapat 11.868 perangkat sistem operasi Android yang berbeda versi. Keberhasilan Android ini juga menjadikannya sebagai target pada "perang telepon pintar" antar perusahaan-perusahaan teknologi. Hingga Mei 2013, sudah lebih dari 900 juta perangkat Android telah diaktifkan di seluruh dunia, dan 48 miliar aplikasi telah dipasang dari Google Play.

## 3.2 Pengertian Android

Android adalah sistem operasi yang dirancang untuk smartphone dan tablet. Sistem Android ini juga memiliki Linux yang menjadi pondasi dasar dari sistem operasi Android. Linux sendiri adalah sistem operasi yang khusus dirancang untuk smartphone dan komputer. Android dirancang untuk dipasang pada perangkat-perangkat mobile touchscreen seperti smartphone dan tablet. Sehingga sistem operasi yang berada di dalam smartphone saat ini dapat menyesuaikan spesifikasi kelas low-end hingga high-end.

Dan kemudian perkembangan sistem android dapat meningkat tajam. Android adalah sistem operasi yang terbuka atau open source, berarti pihak Google memperbolehkan dan membebaskan bagi pihak manapun dapat mengembangkan sistem operasi tersebut. Sistem operasi Android memiliki gudang aplikasi yaitu Google Playstore, Google Playstore ini dapat mendownload aplikasi sepuasnya dengan menggunakan perangkat seluler dengan sistem Android.

## 3.3 Varian Android

Android mempunyai tiga macam varian android, diantaranya yaitu :

### 3.3.1 Android Murni atau Android Stock

Android murni adalah sistem operasi Android yang tidak dirubah sama sekali. Sistem operasi yang digunakan benar-benar murni atau tidak diubah sama sekali dari awal. sistem operasi Android yang digunakan di hp Android murni merupakan versi original.

### 3.3.2 While

While merupakan perulangan yang akan terjadi apabila kondisinya True, perulangan akan terus berjalan hingga diperoleh kondisi False. Berikut ini merupakan contoh penggunaan sintaks perulangan while.

```

1 #perulangan while
2 hitung = 0
3 while (hitung < 9):
4     print ("hitungan ke : ", hitung)
5     hitung = hitung + 1
6
7 print ("Good bye!")

```

### 3.4 Kondisi

Pengambilan keputusan kadang diperlukan dalam sebuah program untuk menentukan tindakan apa yang akan dilakukan sesuai dengan kondisi yang terjadi, contoh kasus misalkan ada seorang anak bernama idam, seorang manusia yang membutuhkan makan, jika idam lapar maka idam akan makan. Maka dapat dijabarkan seperti dibawah ini :

Kondisi, jika :

Idam lapar

Maka :

Idam akan makan

Namun terkadang kondisi juga diberikan tambahan opsional sebuah kondisi tambahan, misalkan jika idam makan maka idam kenyang, namun jika tidak maka idam akan kelaparan. Penjabarannya dapat dilihat sebagai berikut :

Kondisi, jika :

Idam makan

Maka :

Idam akan kenyang

Jika tidak :

Idam akan kelaparan

Contoh diatas dapat ditulis dalam sintax python dengan menggunakan kondisi, pengkondisian dalam python dibagi menjadi 4, yaitu : IF, IF ELSE, ELIF, nested IF. Berikut merupakan pembahasannya.

**3.4.0.1 IF** IF adalah suatu struktur yang memiliki suatu perlakuan jika terjadi suatu kondisi. Akan tetapi, tidak terjadi sesuatu yang lain atau terjadi apa-apa ketika berada di dalam luar kondisi tersebut. IF hanya menjalankan satu kondisi dan menampilkan satu output. Contoh: kondisi dimana variabel a lebih besar dari variabel b, maka tampilkan hasil bahwa a lebih besar dari b.

```

1 #if statement

```

```
1 a = 330
2 b = 200
3 if a > a:
4     print("a lebih besar dari b")
```

**3.4.0.2 IF ELSE** IF ELSE digunakan apabila kondisi yang terjadi bernilai salah, maka lakukan else. Contoh: kondisi dimana variabel a lebih besar dari variabel b, maka jika b lebih besar dari a, tampilkan hasil b lebih besar dari a, jika salah maka tampilkan a lebih besar dari pada b

```
1 #else
2 a = 200
3 b = 33
4 if b > a:
5     print("b is greater than a")
6 else:
7     print("a is greater than b")
```

**3.4.0.3 ELIF** Kondisi ELIF merupakan suatu struktur logika majemuk yang memiliki banyak pilihan aksi terhadap berbagai kemungkinan kejadian yang terjadi. ELIF digunakan apabila kondisi pertama tidak benar maka lakukan kondisi lain (alternatif). Contoh: kondisi dimana variabel a sama dengan variabel b, maka jika b lebih besar dari a, tampilkan hasil b lebih besar dari a, namun jika a dan b bernilai sama, maka tampilkan a sama dengan b

```
1 #elif
2 a = 33
3 b = 33
4 if b > a:
5     print("b lebih besar dari a")
6 elif a == b:
7     print("a sama dengan b")
```

**3.4.0.4 Nested IF** Nested if merupakan if didalam if (if bersarang), terdapat dua if didalam satu kondisi. Contoh: variabel x sama dengan 41, kondisi pertama yaitu jika x besar dari 10 maka tampilkan lebih besar dari 10, kondisi kedua yaitu jika x besar dari 20, maka tampilkan lebih besar dari 20, jika salah maka tampilkan tidak melebihi 20.

```
1 #nested if
2 x = 41
3
4 if x > 10:
5     print("lebih besar dari 10,")
6     if x > 20:
7         print("lebih besar dari 20!")
8     else:
9         print("tidak melebihi 20.")
```

### 3.5 Error

1. NameError, terjadi apabila kode mengeksekusi nama yang tidak terdefenisikan. Contoh:

```
1 nama = "Dinda Majesty"
2 print(Nama)
```

Maka akan menghasilkan output NameError: name Nama is not defined. error ini dapat diatasi dengan mengubah variabel yang di print sesuai dengan variabel yang didefinisikan, karena penulisan pada python bersifat case-sensitive

2. SyntaxError, terjadi apabila kode python mengalami kesalahan saat penulisan. Contoh: menuliskan variabel yang didahului angka (1nama = Dinda Majesty) maka akan muncul error SyntaxError: invalid syntax. error ini dapat diatasi dengan memperhatikan tata cara penulisan kode pada bahasa pemrograman python.
3. Logic error merupakan kesalahan yang terjadi karena kesalahan pembacaan data pada command perintah seperti data tidak terbaca atau tidak ada, dan tidak sesuai dengan aturannya. Contoh kesalahan tipe data yaitu

```
1 a= 4
2 b=6
3
4 print(a+b)
```

4. TypeError, terjadi apabila kode melakukan operasi atau fungsi terhadap tipe data yang tidak sesuai. Contoh: melakukan penjumlahan terhadap tipe data string dan integer. error ini dapat diatasi dengan mengubah tipe data string menjadi integer.

```
1 a = "10"
2 b = 5
3
4 print(a + b)
```

Maka akan menghasilkan output error TypeError: can only concatenate str (not int) to str

5. IndentationError, terjadi apabila kode perulangan atau pengkondisian tidak menjorok kedalam (tidak menggunakan identasi), error ini dapat diatasi dengan menambahkan tab atau spasi. Contoh

```
1 a = 200
2 b = 330
3
4 if b > a:
5 print("b lebih besar dari a")
```

Maka akan menghasilkan output error IndentationError: expected an indented block

### 3.6 Try Except

Try Except merupakan salah satu bentuk penangan error di dalam bahasa pemrograman python, perintah try except ini memiliki fungsi untuk menangkap sebuah error dan tetap menjalankan program kita, sehingga program yang sedang dijalankan akan mengeksekusi program hingga akhir. Contohnya terdapat pada listing berikut

```
1 a="1"
2 b=2
3
4 try :
5     a+b
6 except:
7     print("Error , kedua tipe data berbeda")
```



## BAB 4

---

# FUNGSI DAN KELAS

---

### 4.1 Teori

#### 4.1.1 Fungsi

Fungsi adalah sebuah blok kode yang memiliki nama fungsi dan kode program di-dalamnya jika dijalankan maka fungsi itu akan mengembalikan nilai. Fungsi dapat dipanggil berkali-kali sesuai dengan nama fungsi yang telah didefinisikan. Fungsi memiliki nilai kembalian (return). Contoh fungsi

```
1 def nambahinAngka(angka1 , angka2):  
2     hasil = angka1 + angka2  
3     return hasil
```

Apabila kita dapat memberikan nilai ke angka1 dan angka2, dan apa bila sudah diberi nilai dan program sudah dijalankan, maka program pun akan mengembalikan nilai berupa hasil dari penjumlahan angka 1 dan angka 2.

## 4.2 Package

Package merupakan sekumpulan modul yang dikemas oleh programmer dengan tujuan agar mempermudah dalam pembuatan kode program. Kita dapat membuat sebuah kode program atau fungsi didalamnya dan dapat secara mudah menggunakan kode program itu dengan cara memanggilnya pada kode program lainnya atau import package. Contohnya adalah sebagai berikut

```

1 def my biodata(nama, umur):
2     bio = "nama saya " + nama + " umur saya " + umur
3     return bio
4
5 def my study(kampus, prodi):
6     study = "saya berkuliah di " + kampus + " program studi " + prodi
7     return study

```

Kode diatas merupakan isi dari le fungsi.py, sedangkan saya ingin menjalankan program fungsi.py pada main.py sehingga kode program pada le main.py akan dituliskan seperti berikut:

```

1 import fungsi
2 nama = "Dinda Majesty"
3 umur = "19 Tahun"
4 biodata = my biodata(nama, umur)
5 print(biodata)
6
7 kampus = "Politeknik Pos Indonesia"
8 prodi = "D4-Teknik Informatika"
9 kuliah = my study(kampus, prodi)
10 print(kuliah)

```

Kode program pada le main.py akan mengimport kode program yang ada pada le fungsi.py, sehingga dengan adanya fungsi dan package kita dapat dengan mudah melakukan pemanggilan fungsi yang telah kita deskripsikan sebelumnya, walaupun berada pada le python yang berbeda.

## 4.3 Class, Object, Atribute, and Method

Class atau Kelas merupakan sebuah blueprint/kerangka dari objek yang berisi fungsi dan dibuat untuk mendefenisikan objek dengan atribut yang sesuai dengan kelas yang telah dibuat yang nantinya akan diinisiasi. Objek adalah sebuah wujud yang dapat kita lakukan perintah sesuai dengan methodnya. Sebuah kelas harus memiliki objek yang nantinya akan di kodekan sesuai dengan fungsi yang telah dibuat pada kelas, tanpa adanya objek sebuah kelas tidak akan bisa menjalankan fungsi-fungsi didalamnya. Atribut berisi variabel yang memiliki tipe data dan dapat kita berikan pada objek, atribut ada 2 yaitu kelas atribut dan instansi atribut, perbedaannya hanya di letak, kalau kelas atribut ada di bawah kelas, dan instansi atribut ada didalam fungsi, atribut itu sebuah variabel yang dimiliki oleh parentnya seperti fungsi atau class. .Method merupakan kode program yang berisi tindakan atau perintah untuk menjalankan objek.

```
1 class Fungsi(object):  
2  
3     def Nama(self , namakamu):  
4         self.kamu = namakamu
```

## 4.4 Pemanggilan Class

Pemanggilan library kelas dapat dilakukan dengan cara import dan membuat objek dari kelas tersebut. Contohnya, kita memiliki le python yang diberi nama ngitung dan didalamnya terdapat class Ngitung yang memiliki banyak fungsi didalamnya. Untuk melakukan pemanggilan class maka kita bisa mengetikkan kode seperti berikut.

```
1 import Fungsi
```

## 4.5 Pemakaian Package Fungsi Apabila File Didalam Folder

Pemakaian Package fungsi apabila le terdapat didalam sebuah folder maka kita bisa menggunakan from folder import le dan from le import fungsi. Contohnya, kita memiliki folder src yang didalamnya terdapat le fungsi.py dan didalam fungsi.py terdapat fungsi Berhitung, untuk mengimportkan fungsi maka kita dapat mengetikkan kode seperti berikut.

```
1 from src import fungsi  
2 from fungsi import Berhitung
```

## 4.6 Pemakaian Package Kelas Apabila File didalam Folder

Pemakaian package kelas apabila le terdapat didalam sebuah folder maka kita bisa menggunakan from folder import le dan from le import kelas. Contohnya, kita memiliki folder src yang didalamnya terdapat le fungsi.py dan didalam fungsi.py terdapat kelas Ngitung, maka untuk melakukan import kelas kita dapat mengetikkan kode sebagai berikut.

```
1 from src import fungsi  
2 Kelas = fungsi.Nama(namakamu)
```



## DAFTAR PUSTAKA

---

1. R. Awangga, “Sampeu: Servicing web map tile service over web map service to increase computation performance,” in *IOP Conference Series: Earth and Environmental Science*, vol. 145, no. 1. IOP Publishing, 2018, p. 012057.

