

# **VOICE CLONING**



---

# **VOICE CLONING**

## **using Tacotron 2 and WaveNet**

---

**Dinda Majesty**  
Politeknik Pos Indonesia



**Kreatif Industri Nusantara**

***Penulis:***

Dinda Majesty

ISBN : xxx-xxx-xxxxx-x-x

***Editor:***

Rolly Maulana Awangga

***Penyunting:***

Syafrial Fachrie Pane

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

Dinda Majesty

***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, 2021

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

---

DINDA MAJESTY, Informatics Research Center., Politeknik Pos Indonesia, Bandung,  
Indonesia

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



# CONTENTS IN BRIEF

---

<b>1 Sistem Text to Speech (TTS)</b>	<b>1</b>
<b>2 Neural Network</b>	<b>7</b>
<b>3 Mengenal Python</b>	<b>11</b>
<b>4 Instalasi Anaconda</b>	<b>35</b>
<b>5 Instalasi Pip dan Perintah CLI</b>	<b>49</b>



# DAFTAR ISI

---

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Dinda Majesty</i>	
<b>1 Sistem Text to Speech (TTS)</b>	<b>1</b>
1.1 Text to Speech (TTS)	1
1.1.1 Artificial Intelligence	2
1.1.2 Machine Learning	3
1.1.3 Deep Learning	3

1.2 Cara Kerja TTS	4
<b>2 Neural Network</b>	<b>7</b>
2.1 Neural Network	7
2.1.1 Sejarah Neural Network	8
2.1.2 Konsep Neural Network	9
2.1.3 Fungsi Neural Network	9
2.1.4 Keuntungan dan Penerapan Neural Network	9
<b>3 Mengenal Python</b>	<b>11</b>
3.1 Sejarah Python	11
3.1.1 Perbedaan Python 2 dan 3	12
3.1.2 Implementasi dan Penggunaan Python di Perusahaan Dunia	12
3.1.3 Jenis-Jenis Variabel	13
3.1.4 Input dan Output	14
3.1.5 Operator Aritmatika dan Konversi Tipe Data	14
3.1.6 Perulangan	15
3.1.7 Looping For	16
3.1.8 While	16
3.1.9 Perulangan Bertingkat	17
3.1.10 Break	18
3.1.11 IF Statement	19
3.1.12 ELSE statement	20
3.1.13 Elif	21
3.2 Praktek Pertama Python	22
3.2.1 Modulus	22
3.2.2 Hello NPM	22
3.2.3 Hello NPM (3 Digit Belakang)	22
3.2.4 Hello NPM (Digit ke-3)	23
3.2.5 Variabel Alfabet	23
3.2.6 Penjumlahan NPM	23
3.2.7 Perkalian NPM	24
3.2.8 Print Vertical	24
3.2.9 Digit Genap NPM	25
3.2.10 Digit Ganjil NPM	25
3.2.11 Bilangan Prima NPM	25
3.3 Peringatan Error dan Cara Mengatasinya	26

3.3.1	Except	27
3.3.2	Run Script Hello World di Spyder	27
3.3.3	Pemakaian Variable Explorer	29
3.4	Indentasi	29
3.4.1	Penjelasan Indentasi	29
3.4.2	Jenis-Jenis Error Indentasi	30
3.4.3	Cara Membaca Error	30
3.4.4	Cara Menangani Error	30
3.4.5	Quiz: 1	32
<b>4</b>	<b>Instalasi Anaconda</b>	<b>35</b>
4.1	Pengenalan Anaconda	35
4.1.1	Instalasi Anaconda 3 Windows 10 x64	35
4.1.2	Update Anaconda dan Spyder	42
4.1.3	Instalasi Anaconda Ubuntu 19.04	42
4.1.4	Konfigurasi <i>Python</i>	46
4.1.5	Quiz 2	47
<b>5</b>	<b>Instalasi Pip dan Perintah CLI</b>	<b>49</b>
5.1	Pip	49
5.1.1	Instalasi Pip	49
5.1.2	Command Line Interface/Interpreter	51



# DAFTAR GAMBAR

---

1.1	Korelasi antara Deep Learning, Machine Learning, dan Artificial Intelligence	2
1.2	Perbedaan Machine Learning dan Deep Learning	4
1.3	Cara Kerja Text to Speech (TTS)	5
2.1	Sel Saraf (Neuron)	8
3.1	Perbedaan Python 2 dan Python 3	12
3.2	<i>Launch Spider</i>	28
3.3	<i>Print Hello World</i>	28
3.4	<i>Hello World</i>	28
3.5	<i>Variable Explorer</i>	29
3.6	<i>Indentasi</i>	30
3.7	<i>Error Indentasi</i>	31
3.8	<i>Syntax Error</i>	31

3.9	<i>Syntax yang Telah Diperbaiki</i>	32
4.1	Run Setup Anaconda	36
4.2	Setup Loading	36
4.3	Welcome to Anaconda Setup	37
4.4	<i>License Agreement</i>	37
4.5	<i>Just Me(recomended)</i>	38
4.6	<i>Pilih lokasi</i>	39
4.7	<i>Centang Anaconda to my PATH</i>	39
4.8	<i>Waiting Installation Complete</i>	40
4.9	<i>Installation Complete</i>	40
4.10	<i>Anaconda+JetBrains</i>	41
4.11	<i>Thanks for install Anaconda</i>	42
4.12	Gambar halaman download	43
4.13	Gambar install anaconda	43
4.14	Gambar eksekusi anaconda	44
4.15	Gambar anaconda license agreement	44
4.16	Gambar perintah yes or no	45
4.17	Gambar path anaconda	45
4.18	Gambar perintah install spyder3	46
4.19	Gambar setpath	47
5.1	<i>Install pip</i>	50
5.2	<i>Install pip Selesai</i>	50
5.3	<i>Melihat Versi pip</i>	51
5.4	<i>CLI in Command Prompt</i>	51

## DAFTAR TABEL

---



# Listings

---

3.1	Input dan Output	14
3.2	While Loop	15
3.3	For Loop	15
3.4	Nested Loop	15
3.5	if Statement	19
3.6	Elif	19
3.7	Else	19
3.8	Nested If	20
3.9	Modulus	22
3.10	Hello NPM	22
3.11	3 Digit Belakang	23
3.12	Digit ke-3	23
3.13	Variabel Alfabet	23
3.14	Penjumlahan NPM	24
3.15	Perkalian NPM	24
3.16	Print Vertical	24
3.17	Digit Genap NPM	25
3.18	Digit Ganjil NPM	25

3.19 Bilangan Prima NPM	25
3.20 Try Except	27

# **FOREWORD**

---

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa



# KATA PENGANTAR

---

Buku ini berisi penjelasan teknologi serta tata cara yang harus dilakukan dalam pembuatan *voice cloning* berbahasa indonesia. Buku ini diharapkan dapat membantu orang-orang memahami teknologi terkini terkait *voice cloning* dan cara kerjanya, hal yang dibutuhkan dalam pembuatan *voice cloning* serta cara membuat voice cloning.

TIM PENULIS

*Bandung, Jawa Barat*

*Desember, 2021*



## ACKNOWLEDGMENTS

---

Terima kasih atas semua masukan dari para dosen D4 Teknik Informatika Politeknik Pos Indonesia agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk dosen pembimbing yang telah membimbing dan membantu saya menyelesaikan pembuatan buku ini sebagai syarat kelulusan Internship.

D. M.



## ACRONYMS

---

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



## GLOSSARY

---

chatbot	Merupakan sebuah teknologi layanan obrolan yang ditangani oleh sistem atau robot.
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

---

DINDA MAJESTY

Informatics Research Center  
Bandung, Jawa Barat, Indonesia

Perkembangan teknologi yang semakin maju dalam sistem *Text to Speech (TTS)* memungkinkan manusia membuat kloningan suara hanya dengan beberapa detik sampel suara. Hasil kloningan suara dapat dimodifikasi sesuai dengan inputan teks, sehingga kloningan suara tersebut akan mengucapkan kalimat yang sesuai dengan teks yang diinputkan. Suara yang dihasilkanpun terdengar alami mirip seperti suara manusia aslinya.



# BAB 1

---

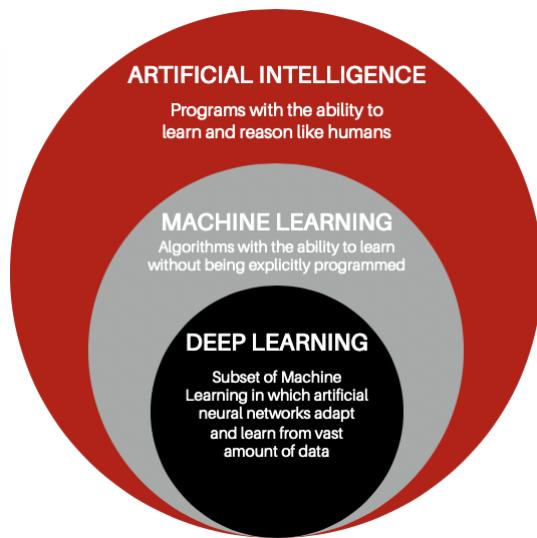
## SISTEM TEXT TO SPEECH (TTS)

---

Sebelum mengetahui teknologi apa saja yang digunakan serta tata cara pembuatan sistem voice cloning ini, akan lebih baik apabila kita memahami terlebih dahulu mengenai sistem *Text to Speech (TTS)*.

### 1.1 Text to Speech (TTS)

Text to Speech merupakan proses pembuatan suara melalui inputan berupa text. mungkin teman-teman bingung, bagaimana bisa text berubah menjadi suara, semestinya text itu hanya dapat kita lihat menggunakan panca indra kita yaitu mata sedangkan suara merupakan sesuatu yang tidak bisa kita lihat menggunakan mata tetapi bisa kita dengar menggunakan telingga. Text to Speech ini ada untuk menjawab kebingungan teman-teman tersebut. Di zaman sekarang ini, dimana teknologi sudah semakin canggih, mengubah teks menjadi suara bukanlah hal yang sulit untuk dilakukan, apalagi setelah teman-teman mengenal deep learning, machine learning, dan artificial intelligence. Bagi teman-teman yang masih belum kenal, yuk kenalan dulu.



**Gambar 1.1** Korelasi antara Deep Learning, Machine Learning, dan Artificial Intelligence

### 1.1.1 Artificial Intelligence

Artificial Intelligence (AI) merupakan kecerdasan yang dibuat dan diberikan atau diterapkan pada program komputer yang diharapkan memiliki kecerdasan seperti manusia sehingga memiliki kemampuan memahami, merencanakan, mengambil keputusan, dan problem solving. Dalam pembuatannya, setiap algoritma memungkinkan mesin untuk meniru, mengembangkan, dan berprilaku seperti manusia. Sederhananya AI itu merupakan sebuah teknologi yang memungkinkan kita menciptakan sebuah mesin yang memiliki kecerdasan berdasarkan pada perintah atau algoritma yang kita berikan pada program tersebut. algoritma berupa langkah-langkah yang kita lakukan dalam menyelesaikan suatu permasalahan. Misalnya kita ingin menciptakan mesin yang dapat menjawab semua pertanyaan kita seperti chatbot, maka kita membutuhkan algoritma atau langkah-langkah apa yang dilakukan oleh chatbot untuk memahami dan mampu memberikan respon secara cepat dan tepat kepada kita.

Artificial Intelligence dibuat dengan 3 tujuan, diantaranya:

1. tujuan utama, Membuat mesin menjadi lebih pintar dan terus mengalami peningkatan seiring dengan perkembangan teknologi
2. tujuan ilmiah, Memahami apa itu kecerdasan buatan sehingga dapat dimanfaatkan untuk membuat mesin yang lebih cerdas dalam membantu dan memecahkan masalah secara efektif dan efisien.

3. tujuan entrepreneurial, Membuat mesin lebih bermanfaat sehingga memudahkan manusia dalam melakukan perkerjaan dan memecahkan masalah secara cepat, tepat, dan lebih teliti.

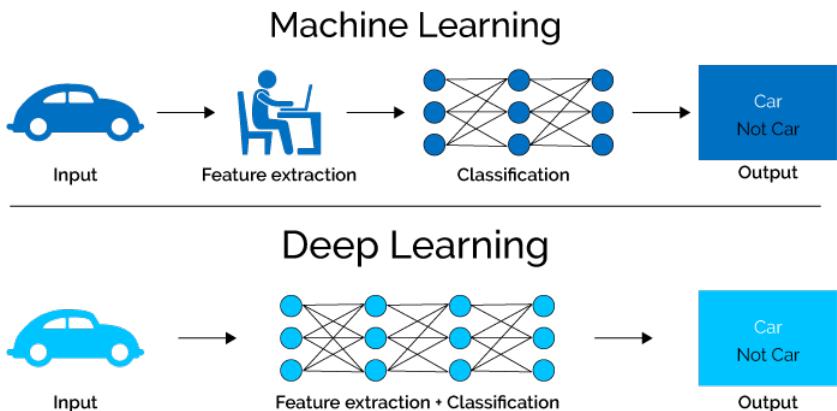
### **1.1.2 Machine Learning**

Machine Learning (ML) merupakan bagian dari Artificial Intelligence yang ditambahkan dengan metode-metode statistika dengan tujuan agar mesin dapat meningkatkan pengalamannya berdasarkan kepada data-data yang ada dan dilatih pada mesin tersebut. Dengan begitu mesin dapat melakukan tugas tertentu berdasarkan data dan algoritma yang diterapkan padanya. Data dan Algoritma yang dirancang membuat mesin terus mengalami peningkatan dari waktu ke waktu. Sebagai contoh, pada pembuatan sistem untuk memprediksi tweet yang bersifat positif, negatif, atau netral kita akan membutuhkan data tweet dari para pengguna twitter, kita bisa dapatkan melalui API yang telah disediakan oleh twitter. pada tahap pertama, kita mengambil 100 data tweet yang terdiri dari tweet positif, negatif, dan netral. Lalu kita melatih mesin menggunakan 100 data tersebut dan menggunakan algoritma naive bayes untuk mengetahui apakah data hasil prediksi kita sama dengan data tweet aslinya dan menghitung akurasi dari algoritma yang telah kita buat. Pada tahap pertama akurasi menunjukkan angka 90% dan ketika kita melakukan training kembali kepada algoritma atau model yang telah kita buat dengan menggunakan data yang lebih banyak lagi maka mesin akan menjadi lebih terlatih dan kemungkinan kesalahan prediksi menjadi sangat kecil bahkan akurasi yang didapatkan dari model bisa mencapai 99% atau bahkan kita bisa mengganti algoritma yang kita gunakan menggunakan algoritma lainnya yang menghasilkan akurasi yang lebih baik dari model atau algoritma kita sebelumnya. hal inilah yang dimaksud dengan machine learning, ada data dan algoritma serta perhitungan-perhitungan yang sering kita jumpai dalam statistika seperti perhitungan akurasi, rata-rata (mean), median, recall, precision, dll.

### **1.1.3 Deep Learning**

Setelah mengenal AI dan ML saatnya kita mengenal Deep Learning (DL) yang merupakan bagian dari Machine Learning yang berkaitan dengan algoritma yang berdasarkan pada struktur dan fungsi otak atau sering dikenal sebagai jaringan saraf tiruan. Pada dasarnya Deep Learning memiliki konsep seperti Machine Learning hanya saja dalam konteks yang lebih dalam. Contohnya, pada deep learning kita ingin membuat mesin dapat mengetahui apakah hewan yang ada digambar merupakan seekor anjing atau kucing, maka kita akan membuat algoritma untuk melakukan pengecekan seperti melakukan pengecekan pada bentuk ekor, bentuk telinga, apakah memiliki kumis atau tidak, dan ciri-ciri lainnya yang membuat kucing dan anjing terlihat berbeda tanpa kita perlu memberikan fitur pembedanya atau fitur mana yang lebih penting untuk dapat mengidentifikasi hewan tersebut secara manual. Deep Learning dapat mengetahui fitur tersebut tanpa kita beritahu secara manual seperti pada Machine Learning seperti terlihat pada gambar 1.2. Oleh karena itu Deep

Learning dianggap sebagai otak utama yang menciptakan kecerdasan buatan yang lebih manusiawi.



**Gambar 1.2** Perbedaan Machine Learning dan Deep Learning

Text to Speech termasuk bagian dari Deep Learning lebih tepatnya Neural Network. Neural Network merupakan cabang ilmu yang mengadopsi cara berpikir dan cara bekerja otak manusia yang memberikan rangsangan/stimulus (input), melakukan proses, dan menghasilkan sesuatu (output). Output dihasilkan dari inputan yang diberikan dan proses yang dilakukan oleh otak manusia. Kemampuan otak untuk memproses inputan atau informasi ini yang akan membuat otak manusia selalu mempelajari hal-hal baru dan terus berkembang, begitupun dengan mesin yang diberikan kecerdasan buatan atau AI. Contohnya saja ketika kita berusia 5 tahun dan otak kita belum memahami informasi-informasi yang sulit seperti pelajaran matematika dasar, menambah dan mengurangi ataupun membagi dan mengalikan angka. Setelah kita mempelajarinya selama 6 tahun di Sekolah Dasar maka otak kita sekarang bisa memahaminya bahkan hal tersebut merupakan hal yang sangat gampang. Pada dasarnya konsep AI sama dengan otak manusia yang selalu berkembang dan mempelajari hal-hal yang baru berdasarkan inputan atau informasi apa yang diberikan kepada otak dan otak akan memproses inputan tersebut hingga dapat memahami dan mengeluarkan hasil dari proses tersebut (output). Dalam pembuatan voice cloning ini kita akan mengajarkan kepada mesin cara membaca text dan mengucapkannya dengan menggunakan suara seperti yang kita contohkan, Oleh karena itu kita membutuhkan sistem text to speech.

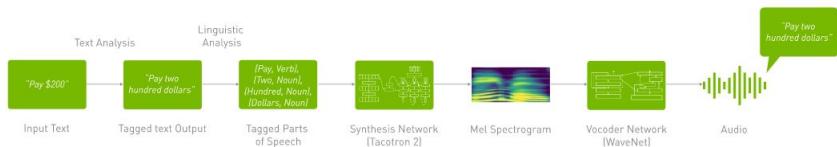
## 1.2 Cara Kerja TTS

Cara kerja TTS terdiri dari 2 proses, dapat dilihat pada gambar 1.3:

1. Model Teks ke Spektrogram, model ini mengubah teks menjadi fitur yang selaras dengan waktu seperti spektogram, mel-spektogram, atau frekuensi F0 dan fitur

lingistik lainnya. ada beberapa model yang bisa digunakan untuk mengubah teks menjadi spektrogram, diantaranya yaitu Tacotron, Tacotron2, Deep Voice 3, dll. Model ini juga disebut sebagai model encoder-decoder.

- Model spektrogram ke audio, model ini akan mengonversi spektrogram yang dihasilkan menjadi audio. Model ini disebut sebagai vocoder. beberapa model yang banyak digunakan untuk membuat vocoder yaitu WaveGlow, Griffin-Lim Algorithm, WaveNet, dll. Model Tacotron-2 sebagai model encoder-decoder dan model WaveNet yang dimodifikasi sebagai vocoder akan dibahas secara mendalam pada buku ini.



**Gambar 1.3** Cara Kerja Text to Speech (TTS)



## BAB 2

---

# NEURAL NETWORK

---

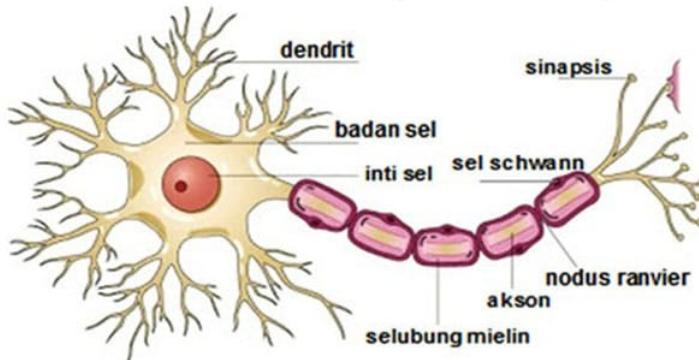
### 2.1 Neural Network

Neural Network terinspirasi dari cara kerja neuron yang ada pada otak manusia, neuron bertugas sebagai penerima stimulus/rangsangan dan pengirim informasi yang akan diolah atau diproses oleh otak menjadi suatu output. Neuron merupakan sistem saraf pusat dan terdapat sekitar 100 miliar neuron yang ada pada tubuh manusia. Perhatikan gambar 2.1

Neuron terdiri dari 3 bagian yaitu akson (akar), soma (batang), dan dendrite (cabang). Neuron juga dibedakan menjadi 3 yaitu neuron sensorik atau sel saraf indera karena fungsinya yang berhubungan dengan penerima (indra) dan saraf pusat (otak dan sumsum tulang belakang). Neuron motorik atau sel saraf penggerak yang berfungsi membawa rangsangan dari saraf pusat (otak dan sumsum tulang belakang) ke otot. dan yang terakhir yaitu neuron asosiasi atau sel saraf penghubung, sel ini menghubungkan atau meneruskan rangsangan dari sel saraf sensorik ke sel saraf motorik.

Tiap neuron yang ada dalam otak kita saling terhubung dan berkirim informasi atau rangsangan berupa neurotransmitter, neuron yang saling berinteraksi dengan mengirimkan rangsangan ini akan menghasilkan kemampuan tertentu pada kerja

# Sel Saraf (Neuron)



Gambar 2.1 Sel Saraf (Neuron)

otak kita. Contohnya, ketika kita bertemu dengan seorang kenalan yang menyapa kita maka otak akan berkerja sehingga dapat mengenali orang tersebut dan akan menghasilkan respon atau output, outputnya bisa berupa sapaan, lambaan tangan, obrolan, dan hal-hal lainnya yang biasa kita lakukan apabila bertemu dengan seseorang yang kita kenal. Jadi secara ilmiahnya, inputan atau rangsangan yang diterima yaitu berupa sapaan dari orang yang dikenali. Rangsangan ini akan diterima oleh alat indra (mata, telinga, hidung, lidah, dan kulit) melalui sel reseptör, kemudian akan diteruskan dalam bentuk impuls berupa arus listrik yang diteruskan ke sel saraf sensorik melalui sinapsis, lalu melewati sel saraf konektor menuju otak. Pada otak informasi akan diolah terlebi dahulu kemudian dikirimkan ke sel saraf motorik dan memberikan atau menghasilkan reaksi berupa sapaan, gerakan berupa lambaan tangan dll. Inilah proses kerja otak manusia begitupun dengan neural network yang terinspirasi dari cara neuron bekerja pada otak manusia. Siapa sangka mesin yang merupakan benda mati akan bisa berprilaku dan berkerja seperti layaknya manusia. Hingga saat ini sangat banyak diminati dan terus dikembangkan dengan tujuan dapat membantu memudahkan manusia dalam bekerja.

## 2.1.1 Sejarah Neural Network

Neural Network bermula ketika Warren McMulloch yang merupakan seorang neurofisiologi dan Walter Pitts yang merupakan seorang ahli matematika menulis makalah tentang cara kerja dari neuron pada tahun 1943. kemudian diperkuatnya konsep neuron dalam buku yang ditulis oleh Donald Hebb pada tahun 1949 yang berjudul *The Organization of Behavior* yang menunjukkan bahwa jaringan syaraf akan bertambah kuat setiap kali digunakan. kemudian dilanjutkan dengan penelitian di IBM untuk mensimulasikan neural network tahun 1950 yang dipimpin oleh Nathaniel Rochester.

Dengan diadakannya konferensi Dartmouth pada tahun 1956 yang membahas tentang penelitian neural network oleh John McCarthy memperkuat konsep mengenai neural network. Lalu pada tahun 1957, John Von Neumann menyarankan untuk meniru fungsi neuron menggunakan relay telegraf atau tabung vakum. Dengan adanya penemuan two-layer-network yang dikenal dengan perceptron. Perceptron berguna untuk menghitung jumlah input, mengurangi treshold, dan meneruskan salah satu dari dua nilai yang mungkin keluar sebagai hasil, penemuan ini ditemukan oleh Frank Rosenblatt pada tahun 1958.

Pada tahun 1959, diperkenalkan model neural network pertama yang dikenal dengan ADALINE (Adaptive Linear Elements) dan MEDALINE (Multiple Adaptive Linear Elements). Model ini merupakan model pertama yang diterapkan pada permasalahan yang ada di dunia nyata. Model ini berfungsi untuk menghilangkan gema pada saluran telepon. Pengembangan model ini dilakukan oleh Bernard Widrow dan Marcian Hoff dari Stanford.

Pada tahun 1982 dalam makalah yang dipresentasikan pada National Academy of Sciences tentang pendekatan untuk menciptakan perangkat yang berguna, menyenangkan, pandai berbicara dan kharismatik, makalah ini dipresentasikan oleh John Hopfield. Lalu konferensi International Institute of Electrical and Electronics (IEEE) mengadakan konferensi mengenai Neural Network yang dihadiri oleh lebih dari 1.800 peserta pada tahun 1987. Sekarang, Neural Network telah diterapkan pada classification, approximation, prediction, recognition, memory simulation, clustering, dll.

### **2.1.2 Konsep Neural Network**

### **2.1.3 Fungsi Neural Network**

### **2.1.4 Keuntungan dan Penerapan Neural Network**



## BAB 3

---

# MENGENAL PYTHON

---

Dalam praktek membuat whatsapp chatbot ini, hal pertama yang harus dilakukan adalah mengenal bahasa pemrograman Python dan mengetahui tentang software yang digunakan yaitu Anaconda3.

### 3.1 Sejarah Python

Nama python berasal dari acara televisi Monty python's flying circus. Python merupakan bahasa pemrograman yang dikembangkan oleh Guido Van Rossum pada tahun 1990 di CWI, Amsterdam. bahasa ini merupakan lanjutan dari bahasa pemrograman ABC. pada tahun 1995, Guido pindah ke CNRI dan mengeluarkan python versi 1.6. pada tahun 2000, Guido pindah ke BeOpen dan mengeluarkan python versi 2.0 setelah itu Guido dan tim PythonLabs pindah ke DigitalCreations. saat ini Guido dan Python Software Foundation terus melakukan perkembangan hingga python versi 2.6.1 dan python versi 3.0 Python Software Foundation merupakan sebuah organisasi yang memiliki hak atas bahasa pemrograman python, hal ini dilakukan untuk mencegah bahasa pemrograman python dimiliki oleh perusahaan komersial.

### 3.1.1 Perbedaan Python 2 dan 3

Perbedaan Python 2 dan Python 3 dapat dilihat pada gambar berikut.

No.	Perbedaan	Python 2	Python 3
1.	Syntax Print	Print tanpa kurung dan menggunakan kurung dapat dijalankan Print("tanpa kurung") Print("dengan kurung")	Print wajib menggunakan kurung, print tanpa kurung menyebabkan error Print("wajib pakai kurung")
2.	Syntax Input	Menggunakan raw_input Nama = raw_input("Masukkan nama anda: ") Menggunakan input akan menyebabkan error	Menggunakan input Nama = input("Masukkan nama anda") Menggunakan raw_input akan menyebabkan error
3.	Hasil Operator Pembagian	3/2 = 1 3//2 = 1 3/2.0 = 1.5 3//2.0 = 1.0 Apabila tipe data float maka akan menghasilkan float, jika tidak maka akan menghasilkan integer	3/2 = 1.5 3//2 = 1 3/2.0 = 1.5 3//2.0 = 1.0 Tipe data float ataupun integer tidak memiliki pengaruh kepada hasil.

**Gambar 3.1** Perbedaan Python 2 dan Python 3

### 3.1.2 Implementasi dan Penggunaan Python di Perusahaan Dunia

#### 1. spotify

spotify adalah suatu layanan musik streaming yang menggunakan pemrograman python untuk analisis data dan backend. pada backend spotify berkomunikasi dengan 0MQ. 0MQ itu sendiri adalah suatu framework dan library open source untuk networking. untuk analisis data tersebut spotify menggunakan luigi, dan modul python yang sinkron dengan hadoop.

#### 2. Google

Google ini sudah menggunakan bahasa pemrograman python ini sudah sajak dari awal berdirinya. Dan pada saat ini bahasa pemrograman python merupakan salah satu bahasa pemrograman server-side resmi di google. Meskipun ada script yang ditulis untuk google menggunakan bahasa perl dan bash, maka nantinya script tersebut akan diubah ke python terlebih dahulu, karena kemudahan dalam perawatannya.

#### 3. Industrial Light and Magic

Industrial Light and Magic ini merupakan studio special efek yang dibutuhkan untuk film star wars saja. Karena infrastruktur awal industrial light and magic ini menggunakan C dan C++, maka akan lebih mudah mengintegrasikan bahasa pemrograman python ketimbang bahasa pemrograman lainnya. Dengan menggunakan bahasa pemprograman python ini industrial light and magic dengan mudah membungkus komponen software dan dapat meningkatkan aplikasi grafisnya.

#### 4. Netflix

Netflix adalah suatu layanan pemutaran film yang dapat dilakukan oleh pengguna dimanapun dan kapanpun. Pada netflix bahasa pemrograman yang digunakan adalah bahasa pemrograman python, bahasa pemrograman ini digunakan pada Central Alert Gateway yang akan me-reroute alert dan mengirimkannya pada individu yang akan melihatnya serta juga dapat secara otomatis reboot atau menghentikan proses yang dianggap bermasalah. Selain itu python juga digunakan untuk menelusuri riwayat dan perubahan pengaturan keamanan.

#### 5. instagram

Instagram adalah suatu aplikasi mobile berbasis IOS, android dan windows phone, dimana pengguna dapat berbagi foto dan video melalui instagram ini. Pada instagram ini menggunakan bahasa pemrograman python dalam task queuenya atau fitur dimana setiap pengguna dapat berbagi foto atau video ke beberapa social network lainnya seperti facebook, twitter, dan lain-lainnya.

### 3.1.3 Jenis-Jenis Variabel

Variabel merupakan tempat penyimpanan data. Tipe data merupakan jenis data yang tersimpan di dalam variabel. terdapat aturan dalam penulisan Variabel.

1. Nama variabel diawali dengan huruf atau garis bawah, contoh: nama, \_nama, namaKu, nama\_variabel.
2. Karakter selanjutnya dapat berupa huruf, garis bawah atau angka, contoh: \_\_nama, nama1, p1.
3. Karakter bersifat case-sensitive (huruf besar dan huruf kecil dibedakan), contoh: Nama dan NAMA keduanya adalah variabel yang berbeda.
4. Nama variabel tidak boleh menggunakan kata kunci yang ada pada bahasa pemrograman python, contoh: if, else, while
5. Nama variabel tidak boleh diawali dengan angka

Jenis-jenis tipe data pada python. Tipe Data Primitif, dibagi menjadi 3 yaitu:

1. Tipe data integer (angka), penulisannya tidak membutuhkan tanda petik, contoh: 10 atau 15

2. Tipe data string (teks), tipe data string ditandai dengan teks yang diapit oleh tanda petik (""), contoh: "nama saya adalah dinda majesty"
3. Tipe data boolean (memiliki dua nilai yaitu true dan false atau 0 dan 1)

Contoh penulisan variabel dan tipe datanya:

1. angka = 10, angka merupakan nama variabel sedangkan 10 adalah nilai dari variabel yang tipe datanya integer.
2. nama = "Dinda Majesty", nama merupakan nama variabel sedangkan "Dinda Majesty" merupakan nilai dari variabel yang tipe datanya string, ditandai dengan adanya petik ("").
3. makan = True , makan merupakan nama variabel sedangkan True merupakan nilai dari variabel yang tipe datanya boolean.

### 3.1.4 Input dan Output

Berikut kode untuk meminta inputan dari user.

```
1 #input output
2 print("masukkan nama anda : ")
3 nama = input()
4 print("nama saya adalah " + nama)
```

**Listing 3.1** Input dan Output

Perintah input() berguna untuk meminta inputan dari user, sehingga memungkinkan user untuk menginputkan data.

Perintah print() berguna untuk menampilkan output dari data yang diinputkan oleh user, sehingga data yang diinputkan user dapat ditampilkan ke layar.

### 3.1.5 Operator Aritmatika dan Konversi Tipe Data

Operator aritmatika

1. penjumlahan (+)
2. pengurangan (-)
3. perkalian (\*)
4. pembagian (/)
5. sisa bagi/modulus (%)
6. pemangkatan (\*\*)

Cara melakukan perubahan terhadap tipe data string menjadi integer, contoh: variabel = "10". Kita dapat mengubah string "10" menjadi angka 10 dengan menambahkan kode int(variabel), dengan begitu 10 yang awalnya bertipe data string akan

dikonversikan menjadi integer.

Cara melakukan perubahan terhadap tipe data integer menjadi string, contoh: variabel = 150. Kita dapat mengubah integer 150 menjadi string "150" dengan menambahkan kode str(variabel), maka tipe data dari variabel akan dikonversikan menjadi string.

### 3.1.6 Perulangan

perulangan terdiri atas 3 kondisi.

1. While, apabila kondisinya True, maka perulangan akan terus berjalan hingga diperoleh kondisi False. Contoh penggunaan 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!")
```

**Listing 3.2** While Loop

2. For, perulangan for bisa melakukan perulangan terhadap item apapun seperti list atau string. Contoh penggunaan For:

```
1 #perulangan for
2 minum = ["kopi", "susu", "teh"]
3 for minuman in minum:
4     print("Saya suka minum", minuman)
```

**Listing 3.3** For Loop

3. nested, perulangan ini memungkinkan adanya perulangan didalam perulangan. Contoh penggunaan nested:

```
1 #nested loop
2     i = 2
3 while(i < 100):
4     j = 2
5     while(j <= (i/j)):
6         if not(i%j): break
7         j = j + 1
8     if (j > i/j) : print(i, " is prime")
9     i = i + 1
10
11 print ("Good bye!")
```

**Listing 3.4** Nested Loop

Pada penulisan sintaks While dan For harus memperhatikan identasi (baris yang menjorok ke dalam), jika tidak diperhatikan dengan baik maka akan terjadi error terhadap identasi. Untuk menambahkan identasi dapat menggunakan spasi atau tab.

### 3.1.7 Looping For

Seperti di bahasa pemrograman lainnya, Python juga memiliki fungsi for. Bedanya di Python, For tidak hanya untuk perulangan dengan jumlah finite (terbatas), melainkan lebih ke fungsi yang dapat melakukan perulangan pada setiap jenis variabel berupa kumpulan atau urutan. Variabel yang dimaksud bisa berupa list, string, ataupun range. Jika sebuah list atau urutan berisi expression, maka ia akan dievaluasi terlebih dahulu. Kemudian item pertama pada urutan atau list akan diassign sebagai variabel iterating\_var. Setelahnya, blok statement akan dieksekusi, berlanjut ke item berikutnya, berulang, hingga seluruh urutan habis. contoh for:

```
for letter in 'Python': # First Example
    print('Current Letter: {}'.format(letter))
fruits = ['banana', 'apple', 'mango']
for fruit in fruits: # Second Example
    print('Current fruit: {}'.format(fruit))
```

Output:

```
Current Letter : P
Current Letter : y
Current Letter : t
Current Letter : h
Current Letter : o
Current Letter : n
Current fruit : banana
Current fruit : apple
Current fruit : mango
```

Anda juga dapat melakukan perulangan berdasarkan indeks atau range dengan memanfaatkan fungsi len():

```
fruits = ['banana', 'apple', 'mango']
for index in range(len(fruits)):
    print('Current fruit: {}'.format(fruits[index]))
```

Output:

```
Current fruit : banana
Current fruit : apple
Current fruit : mango
```

### 3.1.8 While

While pada bahasa Python digunakan untuk mengeksekusi statement selama kondisi yang diberikan terpenuhi (True). Kondisi dapat berupa expression apapun, dan harap

diingat bahwa True di Python termasuk semua nilai non-zero. Saat kondisi menjadi False, program akan melanjutkan ke baris setelah blok statement.

Seperti for dan semua statement percabangan, blok statement yang mengikuti kondisi while dan memiliki posisi indentasi yang sama, dianggap blok statement yang akan dieksekusi.

Contoh:

```
count = 0
while (count < 5):
    print('The count is: {}'.format(count))
    count = count + 1
```

Output:

```
The count is: 0
The count is: 1
The count is: 2
The count is: 3
The count is: 4
```

Seperti pada bahasa lainnya, eksekusi statement while mungkin bersifat infinit / infinite loop saat sebuah kondisi tidak pernah bernilai False. Contohnya sebagai berikut:

```
var = 1
while var == 1: # This constructs an infinite loop
    num = input('Enter a number: ')
    print('You entered: {}'.format(num))

while True: # This constructs an infinite loop
    num = input('Enter a number: ')
    print('You entered: {}'.format(num))
```

Potongan kode di atas tidak akan pernah bernilai False karena nilai var tidak pernah berubah. Untuk menghentikan infinite loop, gunakan CTRL (atau CMD) - C untuk menghentikannya dan keluar dari program.

Anda juga dapat menyingkat penulisan blok statement While jika statement Anda cukup terwakili oleh satu baris.

```
while (var1): do_something()
```

### 3.1.9 Perulangan Bertingkat

Ada kalanya Anda perlu untuk melakukan perulangan bertingkat, misalnya untuk menghasilkan contoh print-out berikut:

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

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

Contoh:

```
for i in range(0, 5):  
    for j in range(0, 5 - i):  
        print('*', end='')  
    print()
```

### 3.1.10 Break

Pernyataan break menghentikan perulangan kemudian keluar, dilanjutkan dengan mengeksekusi pernyataan (statement) setelah blok perulangan. Salah satu penggunaannya yang paling sering adalah sebuah kondisi eksternal yang membutuhkan program untuk keluar dari perulangan. Jika Anda memiliki perulangan bertingkat, break akan menghentikan perulangan sesuai dengan tingkatnya atau di perulangan mana ia berada. Namun jika ia diletakkan di perulangan dengan kedalaman kedua misalnya, hanya perulangan itu saja yang berhenti, tidak dengan perulangan utama. Contoh 1:

```
for letter in 'Python':  
    if letter == 'h':  
        break  
    print('Current letter: {}'.format(letter))
```

Output contoh 1:

```
Current Letter : P  
Current Letter : y  
Current Letter : t
```

Contoh 2:

```
var = 10  
while var > 0:  
    print('Current variable value: {}'.format(var))  
    var = var - 1  
    if var == 5:  
        break
```

Output contoh 2:

```
Current variable value : 10  
Current variable value : 9  
Current variable value : 8  
Current variable value : 7  
Current variable value : 6
```

### 3.1.11 IF Statement

kondisi if dapat digunakan didalam looping dan dapat digunakan untuk memberikan kondisi tertentu dengan cara mengetikkan if lalu kondisi yang akan terjadi.

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

**Listing 3.5** if Statement

2. 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")
```

**Listing 3.6** Elif

3. 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 a dan b bernilai sama, maka tampilkan a sama dengan b, 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 elif a == b:
7     print("a and b are equal")
8 else:
9     print("a is greater than b")
```

**Listing 3.7** Else

4. 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.")

```

**Listing 3.8** Nested If

contoh if statement:

```

var1 = 100
if var1:
    print ("1 - Got a true expression value")
    print (var1)
var2 = 0
if var2:
    print ("2 - Got a true expression value")
    print (var2)

```

Output:

```

1 - Got a true expression value
100

```

### 3.1.12 ELSE statement

Statement Else dapat dikombinasikan dengan IF Statement, sebagai jalan keluar saat kondisi / hasil evaluasi bernilai False. Else bersifat opsional dan tunggal. contoh else statement:

```

amount = int(input("Enter amount: "))
if amount<1000:
    discount = amount*0.05
    print ("Discount",discount)
else:
    discount = amount*0.10
    print ("Discount",discount)

print ("Net payable:",amount-discount)

```

Output true:

```
Enter amount: 600
Discount 30.0
Net payable: 570.0
```

Output false:

```
Enter amount: 1200
Discount 120.0
Net payable: 1080.0
```

### 3.1.13 Elif

Alternatif untuk Switch/Case dan IF bertingkat di python. Elif adalah kependekan dari else if, dan merupakan alternatif untuk if bertingkat atau switch/case di beberapa bahasa pemrograman lain. Sebuah IF Statement dapat diikuti satu atau lebih statement elif (opsional dan tidak dibatasi). contoh elif:

```
amount = int(input("Enter amount: "))
if amount<1000:
    discount = amount*0.05
    print ("Discount",discount)
elif amount<5000:
    discount = amount*0.10
    print ("Discount",discount)
else:
    discount = amount*0.15
    print ("Discount",discount)
print ("Net payable:",amount-discount)
```

Output true if:

```
Enter amount: 600
Discount 30.0
Net payable: 570.0
```

Output true elif:

```
Enter amount: 3000
Discount 300.0
Net payable: 2700.0
```

Output false:

```
Enter amount: 6000
Discount 900.0
Net payable: 5100.0
```

## 3.2 Praktek Pertama Python

### 3.2.1 Modulus

Praktek kali ini kita akan mencoba praktek menggunakan modulus atau sisa bagi, kita membuat inputan terlebih dahulu menggunakan perintah input. Kemudian buatlah variabel untuk menampung hasil sisa bagi dari jumlah yang diinputkan. Misalnya, teman-teman menginputkan nilai yaitu 1184011, maka hasil modulus 1184011 mod 3 adalah 1 maka print 1184011 menggunakan tanda pagar. jika hasil modulus adalah 0 maka print 1184011 menggunakan tanda bintang.

```

1 #Modulus
2 print("Soal no 1")
3
4 print("Masukkan NPM anda: ")
5 NPM = input()
6
7 npm = int(NPM) % 3
8 print(npm)
9
10 print("###    ###    #####    ###    ###    #####    ###    ###")
11 print("###    ###    #####    ###    ###    #####    ###    ###")
12 print("###    ###    ###    ###    ###    ###    ###    ###")
13 print("###    ###    ###    ###    ###    ###    ###    ###")
14 print("###    ###    #####    #####    ###    ###    ###")
15 print("###    ###    ###    ###    ###    ###    ###    ###")
16 print("###    ###    ###    ###    ###    ###    ###    ###")
17 print("###    ###    #####    ###    #####    ###    ###")
18 print("###    ###    #####    ###    #####    ###    ###")

```

**Listing 3.9** Modulus

### 3.2.2 Hello NPM

Setelah selesai praktek modulus kita akan menampilkan output berupa kalimat "Hello 1184011 Apa Kabar" sebanyak 2 digit belakang angka, yaitu angka 11, maka akan berulang sebanyak 11 kali.

```

1 #Hello NPM
2 print("Soal No 2")
3
4 Loop = NPM[5:7]
5
6 for x in range(int(Loop)):
7     print("Hallo " + NPM + " Apa Kabar?")

```

**Listing 3.10** Hello NPM

### 3.2.3 Hello NPM (3 Digit Belakang)

Jika telah selesai, maka selanjutnya kita akan melakukan praktek untuk menampilkan output berupa kalimat "Hallo 011 Apa Kabar?" sebanyak angka keenam ditambah

angka ketujuh atau 1 ditambah 1, sehingga kalimat tersebut akan berulang sebanyak 2 kali.

```

1 #3 digit belakang NPM
2 print("Soal No 2")
3
4 Loop = NPM[4:7]
5
6 total = int(NPM[5]) + int(NPM[6])
7
8 for x in range(total):
9     print("Hallo " + Loop + " Apa Kabar?")

```

**Listing 3.11** 3 Digit Belakang

### 3.2.4 Hello NPM (Digit ke-3)

Kemudian kita akan melakukan praktek untuk menampilkan output berupa kalimat "Hallo 0 Apa Kabar?".

```

1 #digit ke 3
2 print("Soal No 3")
3
4 Loop = NPM[4]
5 print("Hello " + Loop + " Apa Kabar?")

```

**Listing 3.12** Digit ke-3

### 3.2.5 Variabel Alfabet

Teman-teman, sekarang mari kita coba menambahkan variabel pada angka 1184011, tambahkan abcdefg kedalam sebuah variabel dan tambahkan variabel index dengan nilai 0, buatlah perulangan agar variabel huruf menyesuaikan dengan variabel angka sehingga menjadi a=1, b=1, c=8, d=4, e=0, f=1 ,g=1.

```

1 #variabel alfabet
2 print("Soal no 5")
3
4 var = "abcdefg"
5 index = 0
6
7 for i in var:
8     print(i + " = " + NPM[index])
9     index += 1

```

**Listing 3.13** Variabel Alfabet

### 3.2.6 Penjumlahan NPM

Praktek selanjutnya adalah menjumlahkan angka 1184011 dengan menerapkan perulangan dan penambahan. Apabila nilai 1 telah didapatkan maka akan ditambahkan

dengan nilai 1 sehingga menjadi 2, kemudian nilai 2 ditambahkan lagi dengan nilai selanjutnya yaitu 8 sehingga menjadi 10, begitu seterusnya.

```

1 #penjumlahan NPM
2 print("Soal no 6")
3
4 index = 0
5 angka = 0
6
7 for i in NPM:
8     jumlah = int(NPM[index]) + int(angka)
9     angka = jumlah
10    index += 1
11
12 print(jumlah)

```

**Listing 3.14** Penjumlahan NPM

### 3.2.7 Perkalian NPM

Praktek selanjutnya adalah mengalikan angka 1184011 dengan menerapkan perulangan dan perkalian. Apabila nilai 1 telah didapatkan maka akan dikalikan dengan nilai 1 sehingga menjadi 1, kemudian nilai 1 dikalikan lagi dengan nilai selanjutnya yaitu 8 sehingga menjadi 8, begitu seterusnya.

```

1 #perkalian NPM
2 print("Soal no 7")
3
4 index = 0
5 angka = 0
6
7 for i in NPM:
8     jumlah = int(NPM[index]) * int(angka)
9     angka = jumlah
10    index += 1
11
12 print(jumlah)

```

**Listing 3.15** Perkalian NPM

### 3.2.8 Print Vertical

Melakukan print secara vertikal hanya perlu menambahkan perulangan terhadap nilai 1184011.

```

1 #print vertical
2 print("Soal no 8")
3
4 for i in NPM:
5     print(i)

```

**Listing 3.16** Print Vertical

### 3.2.9 Digit Genap NPM

Selanjutnya, mari kita lakukan print hanya terhadap digit genap pada angka 1184011 dengan memanfaatkan perulangan, if statement dan operator logika. Logika yang akan diterapkan adalah masing-masing angka 1184011 akan di cek terlebih dahulu, apakah angka tersebut memiliki angka yang apabila dibagi 2 akan menghasilkan sisa bagi sama dengan 0 dan angka tersebut tidak boleh sama dengan 0, karena angka 0 bukan merupakan angka ganjil maupun genap.

```
1 #digit genap NPM
2 print("Soal no 9")
3
4 index = 0
5 for i in NPM:
6     if (int(NPM[index])%2 == 0) & (int(NPM[index]) != 0):
7         print(NPM[index])
8     index += 1
```

**Listing 3.17** Digit Genap NPM

### 3.2.10 Digit Ganjil NPM

Selanjutnya, mari kita lakukan print hanya terhadap digit ganjil pada angka 1184011 dengan memanfaatkan perulangan, if statement dan operator logika. Logika yang akan diterapkan adalah masing-masing angka 1184011 akan di cek terlebih dahulu, apakah angka tersebut memiliki angka yang apabila dibagi 2 akan menghasilkan sisa bagi tidak sama dengan 0 dan angka tersebut tidak boleh sama dengan 0, karena angka 0 bukan merupakan angka ganjil maupun genap.

```
1 #digit ganjil NPM
2 print("Soal no 10")
3
4 index = 0
5 for i in NPM:
6     if (int(NPM[index])%2 != 0) & (int(NPM[index]) != 0):
7         print(NPM[index])
8     index += 1
```

**Listing 3.18** Digit Ganjil NPM

### 3.2.11 Bilangan Prima NPM

Praktek terakhir adalah menampilkan hasil dari bilangan prima angka 1184011 dengan cara apabila angka 1184011 memiliki angka yang merupakan angka prima maka akan ditampilkan, logika yang akan diterapkan adalah apabila angka kecil sama dengan angka 1 maka angka tersebut bukan bilangan prima. Jika angka 1184011 dibagi 2 setelah itu dibagi dengan angka itu sendiri dan menghasilkan sisa bagi sama dengan 0, maka angka tersebut bukan bilangan prima

```
1 #bilangan prima NPM
2 print("Soal no 11")
```

```

3 index = 0
4
5 for i in NPM:
6     prima = True
7     var=int(NPM[ index ])
8     if(var<=1):
9         prima=False
10    for i in range (2,var):
11        if(var%i==0):
12            prima=False
13    if(prima==True):
14        print(var,"Prima")
15    else:
16        print(var,"bukan prima")
17 index += 1

```

**Listing 3.19** Bilangan Prima NPM

### 3.3 Peringatan Error dan Cara Mengatasinya

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

```

nama = "Dinda Majesty"
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 eror SyntaxError: invalid syntax. error ini dapat diatasi dengan memperhatikan tata cara penulisan kode pada bahasa pemrograman python.
3. TypeError, terjadi apabila kode melakukan operasi atau fungsi terhadap tipe data yang tidak sesuai. Contoh: melakukan penjumlahan terhadap tipe data string dan integer. eror ini dapat diatasi dengan mengubah tipe data string menjadi integer.

```

a = "10"
b = 5

print(a + b)

```

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

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

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

Maka akan menghasilkan output eror IndentationError: expected an indented block

### 3.3.1 Except

Try Except merupakan perintah yang bisa digunakan dalam penanganan error pada bahasa pemrograman python. perintah ini biasanya digunakan saat penanganan error input/output, operasi database, pengaksesan indeks suatu list atau dictionary dan berbagai kasus lainnya.

Contoh sederhana penggunaan Try-Except saat menangani NameError

```
1 #try except
2 try:
3     print(x)
4 except NameError:
5     print("Variable x tidak ada")
6 except:
7     print("ada sesuatu yang salah nih")
```

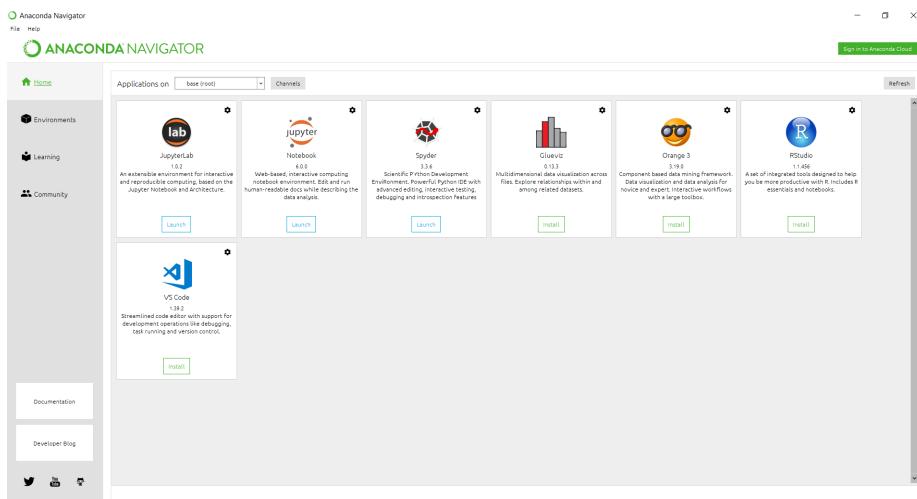
**Listing 3.20** Try Except

Pada contoh diatas, try except akan menghasilkan output Variabel x tidak ada, karena kita tidak mendefinisikan variabel x sebelum kita menampilkan output x ke layar. Dengan menggunakan try except kode yang kita buat terhindar dari error dan kita bisa mengetahui kesalahan yang terjadi ketika terdapat error pada kode kita.

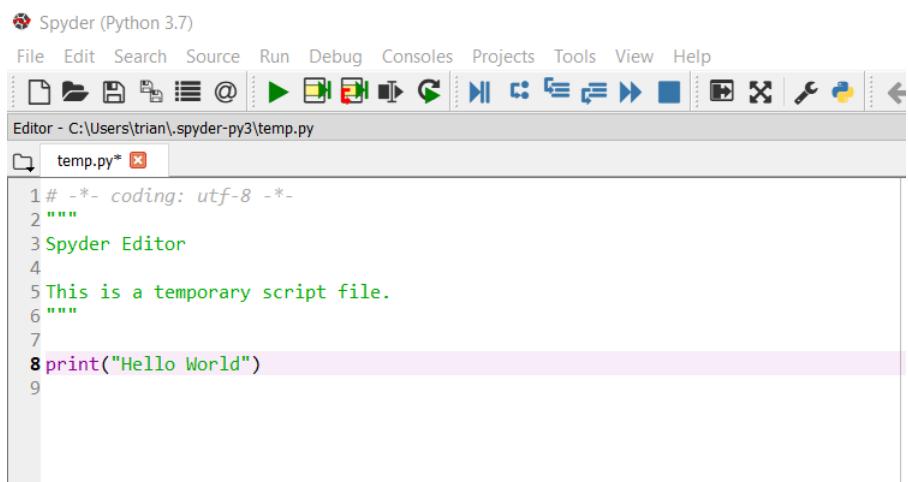
### 3.3.2 Run Script Hello World di Spyder

Jika teman-teman telah berhasil melakukan instalasi anaconda3 maka teman-teman akan memiliki software spyder, pada software ini teman-teman akan mengetikkan kode program yang akan teman-teman buat nantinya. Sekarang ayo kita coba menjalankan program untuk mencetak kata *hello world*.

1. Buka anaconda navigator, lalu klik launch seperti pada gambar 3.2.
2. ketikkan `print("Hello World")` dan run spyder dengan cara mengklik tombol berwarna hijau yang terletak ditengah toolbar, untuk lebih jelas dapat dilihat pada gambar 3.3.
3. hasilnya akan tampak seperti pada gambar 3.4



**Gambar 3.2** *Launch Spyder*



**Gambar 3.3** *Print Hello World*

---

```

Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.6.1 -- An enhanced Interactive Python.

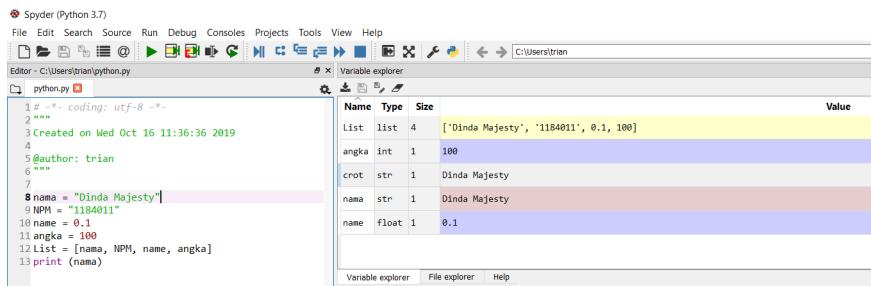
In [1]: runfile('C:/Users/trian/.spyder-py3/temp.py', wdir='C:/Users/trian/.spyder-py3')
Hello World

```

**Gambar 3.4** *Hello World*

### 3.3.3 Pemakaian Variable Explorer

Variable explorer akan secara otomatis terisi ketika kita membuat variable, pada variable explorer kita bisa melihat nama variable, tipe data, length, dan value dari variable tersebut. Contoh penggunaan variabel explorer dapat teman-teman lihat pada gambar 3.5.



Gambar 3.5 Variable Explorer

## 3.4 Indentasi

### 3.4.1 Penjelasan Indentasi

Indentasi adalah bagian paragraf yang menjorok ke dalam pada baris-baris paragraf. Mengatur indentasi dapat menggunakan tab atau spasi. Indentasi digunakan oleh bahasa pemrograman python sebagai pengganti briket () untuk membuka dan menutup fungsi. Error indentasi dapat terjadi apabila syntax tidak menggunakan tab atau space. Contoh yang benar (menggunakan tab/spasi sebagai indentasi):

```
# blok percabangan if
if username == 'petanikode':
    print("Selamat Datang Admin")
    print("Silahkan ambil tempat duduk")
```

```
# blok percabangan for
for i in range(10):
    print i
```

Contoh yang salah (tidak menggunakan tab/spasi):

```
# blok percabangan if
if username == 'petanikode':
print("Selamat Datang Admin")
print("Silahkan ambil tempat duduk")
```

```
# blok percabangan for
for i in range(10):
    print i
```

### 3.4.2 Jenis-Jenis Error Indentasi

IndentationError: unexpected indent. Error diatas terjadi apabila syntax kekurangan tab atau spasi. Contoh error identasi dapat dilihat pada gambar 3.6. Apabila di

```
Spyder (Python 3.7)
File Edit Search Source Run Debug Consoles Projects Tools View Help
Editor - C:\Users\trian\send_whatsapp\coba.py
WA.py x coba.py x
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 16 18:13:23 2019
4
5 @author: trian
6 """
7 username = input()
8 # blok percabangan if
9 if username == 'petanikode':
10 print("Selamat Datang Admin")
11     print("Silahkan ambil tempat duduk")
12
13 # blok percabangan for
14 for i in range(10):
15     print (i)
```

Gambar 3.6 *Indentasi*

running akan memunculkan error seperti pada gambar 3.7.

### 3.4.3 Cara Membaca Error

Jika terjadi error maka cari di line berapa error terjadi, pada gambar 3.7 terdapat error indentasi pada line 10.

### 3.4.4 Cara Menangani Error

Menangani error indentasi dapat dilakukan dengan cara menambahkan tab atau space pada line yang error. Untuk lebih jelasnya dapat teman-teman lihat pada gambar 3.8.

```
In [6]: runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/send_whatsapp')
Traceback (most recent call last):

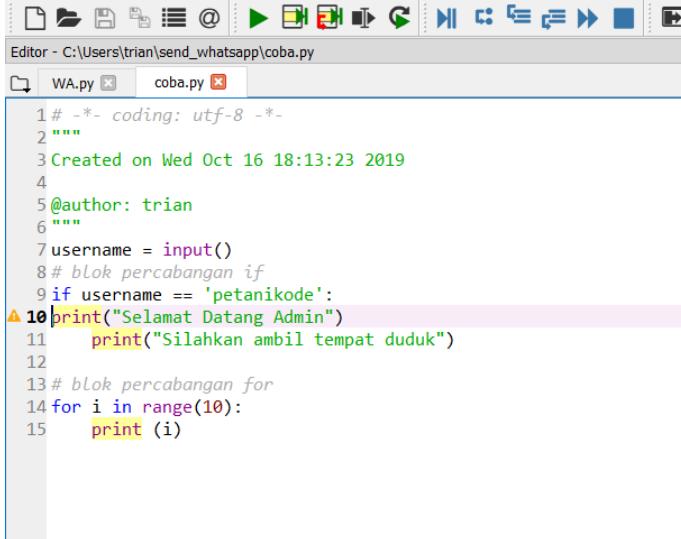
  File "C:\Users\trian\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 3325, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)

  File "<ipython-input-6-0a8945c5cd46>", line 1, in <module>
    runfile('C:/Users/trian/send_whatsapp/coba.py', wdir='C:/Users/trian/send_whatsapp')

  File "C:/Users/trian/Anaconda3/lib/site-packages\spyder_kernels\customize\spydercustomize.py", line 827, in runfile
    execfile(filename, namespace)

  File "C:/Users/trian/Anaconda3/lib/site-packages\spyder_kernels\customize\spydercustomize.py", line 110, in execfile
    exec(compile(f.read(), filename, 'exec'), namespace)

  File "C:/Users/trian/send_whatsapp/coba.py", line 10
    print("Selamat Datang Admin")
           ^
IndentationError: expected an indented block
```

**Gambar 3.7 Error Indentasi**


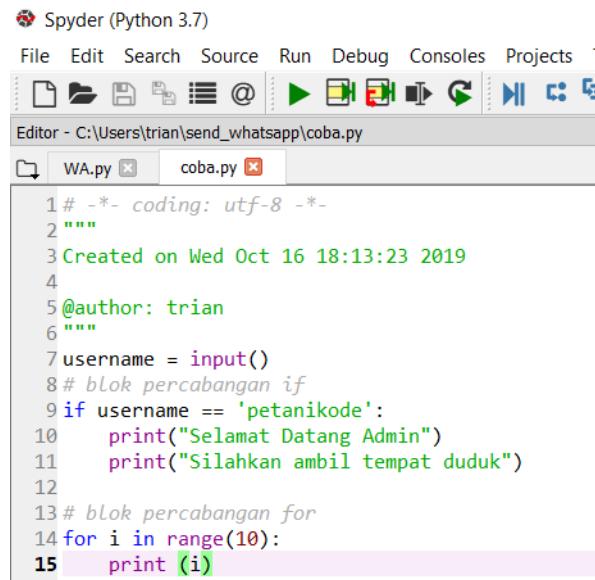
The screenshot shows the Spyder Python 3.7 IDE interface. The menu bar includes File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, and Help. Below the menu is a toolbar with various icons. The main area is titled 'Editor - C:/Users/trian/send\_whatsapp/coba.py'. There are two tabs: 'WA.py' and 'coba.py'. The 'coba.py' tab is active and contains the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 16 18:13:23 2019
4
5 @author: trian
6 """
7 username = input()
8 # blok percabangan if
9 if username == 'petanikode':
10    print("Selamat Datang Admin")
11    print("Silahkan ambil tempat duduk")
12
13 # blok percabangan for
14 for i in range(10):
15    print (i)
```

A syntax error is highlighted at line 10, where the 'print' statement is missing its opening parenthesis. A yellow arrow points to the error, and the line is highlighted in pink.

**Gambar 3.8 Syntax Error**

Penulisan syntax identasi yang benar dapat dilihat pada gambar 3.9.



```

Spyder (Python 3.7)

File Edit Search Source Run Debug Consoles Projects Help
Editor - C:\Users\trian\send_whatsapp\coba.py
WA.py coba.py

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Oct 16 18:13:23 2019
4
5 @author: trian
6 """
7 username = input()
8 # blok percabangan if
9 if username == 'petanikode':
10     print("Selamat Datang Admin")
11     print("Silahkan ambil tempat duduk")
12
13 # blok percabangan for
14 for i in range(10):
15     print(i)

```

**Gambar 3.9** Syntax yang Telah Diperbaiki

### 3.4.5 Quiz: 1

1. Siapakah pencipta Python?
  - a) Guido van Rossum
  - b) Guido van Persie
  - c) Guido van Linguini
  - d) Guido van Laptop
  
2. Pada tahun berapa python dikembangkan?
  - a) 1999
  - b) 1995
  - c) 1945
  - d) 1990
  
3. Apa itu indentasi?
  - a) Bagian paragraf yang menjorok ke dalam
  - b) Bagian paragraf yang menjorok ke sungai
  - c) Bagian paragraf yang menjorok ke laut
  - d) Bagian paragraf yang menjorok ke hati
  
4. Apa itu try except?
  - a) Perintah untuk penanganan masalah hidup
  - b) Perintah untuk penanganan masalah error
  - c) Perintah untuk penanganan masalah galau
  - d) Perintah untuk penanganan masalah sakit perut

5. Apa itu if statement?

- a) Kondisi yang digunakan untuk percabangan logika
- b) Kondisi yang digunakan untuk percabangan pohon
- c) Kondisi yang digunakan untuk percabangan tunas
- d) Kondisi yang digunakan untuk percabangan akar



## BAB 4

---

# INSTALASI ANACONDA

---

### 4.1 Pengenalan Anaconda

Anaconda merupakan sebuah software yang mendistribusikan bahasa pemrograman python dan R untuk keperluan komputasi ilmiah seperti data science, machine learning, data processing skala-luas, analisis prediksi, dan lain sebagainya. Anaconda memiliki anaconda navigator yang didalamnya terdapat software-software yang dapat digunakan untuk membuat program python dan R. Salah satu software yang akan kita gunakan yaitu Spyder. Sebelum kita membuat program kita harus menginstall anaconda terlebih dahulu. Instalasi pada kali ini menggunakan 2 sistem operasi yaitu Windows 10 x64 dan Ubuntu 19.04. Hal yang dibutuhkan adalah laptop dengan os windows atau ubuntu dan internet.

#### 4.1.1 Instalasi Anaconda 3 Windows 10 x64

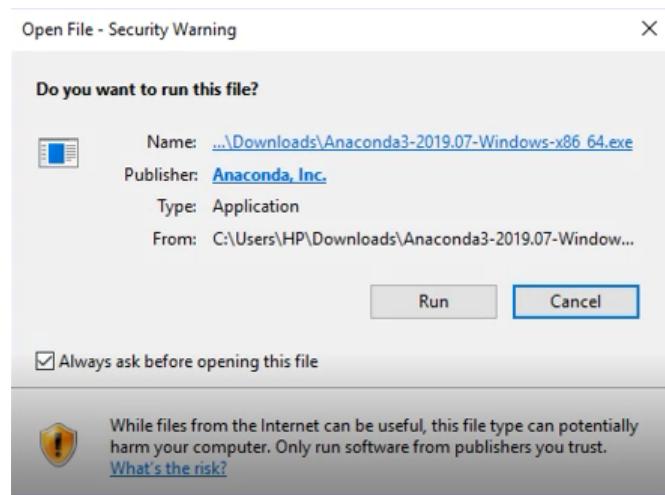
Hal yang harus diperhatikan sebelum melakukan instalasi *Anaconda Python*

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

3. Download file anaconda yang sesuai dengan versi sistem operasi (32bit atau 64bit)

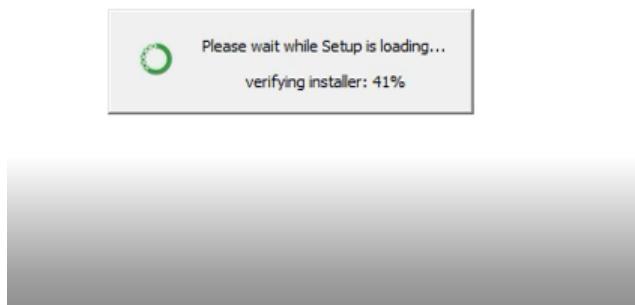
Berikut langkah-langkah instalasi anaconda.

1. Buka aplikasi *installer Anaconda* yang telah didownload lalu akan muncul gambar 4.1, lalu pilih run untuk menjalankan proses instalasi.



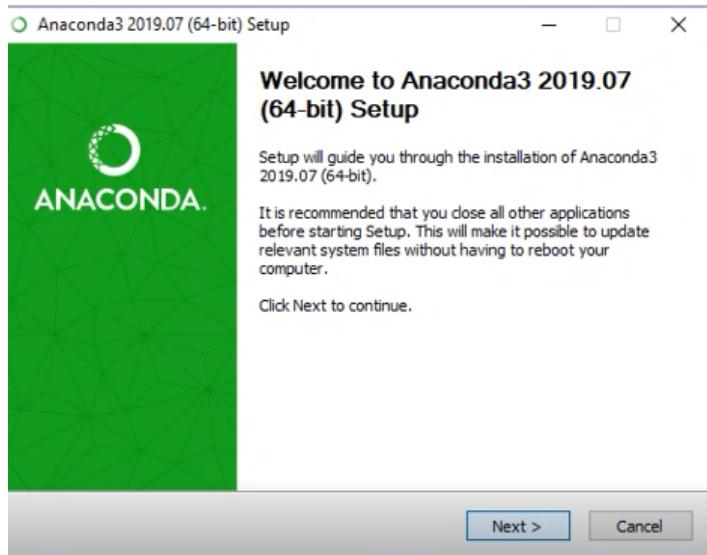
**Gambar 4.1** Run Setup Anaconda

2. Tunggu hingga *setup loading* selesai seperti pada gambar 4.2.



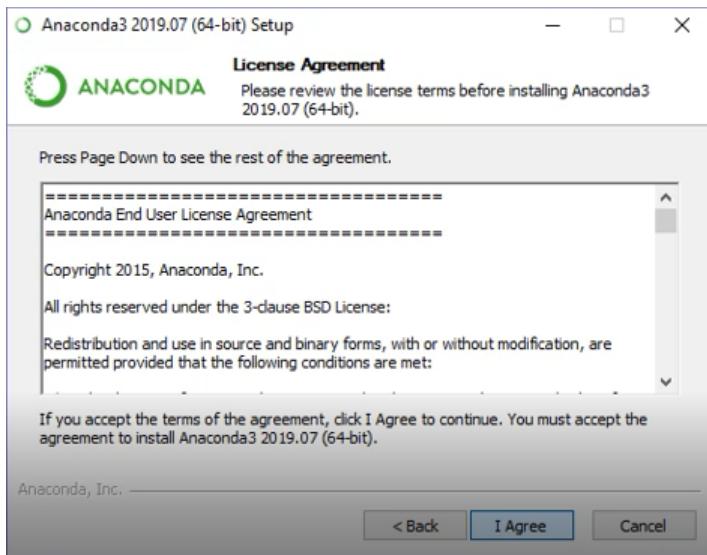
**Gambar 4.2** Setup Loading

3. Jika *setup loading* telah selesai, maka klik *next* seperti pada gambar 4.3.



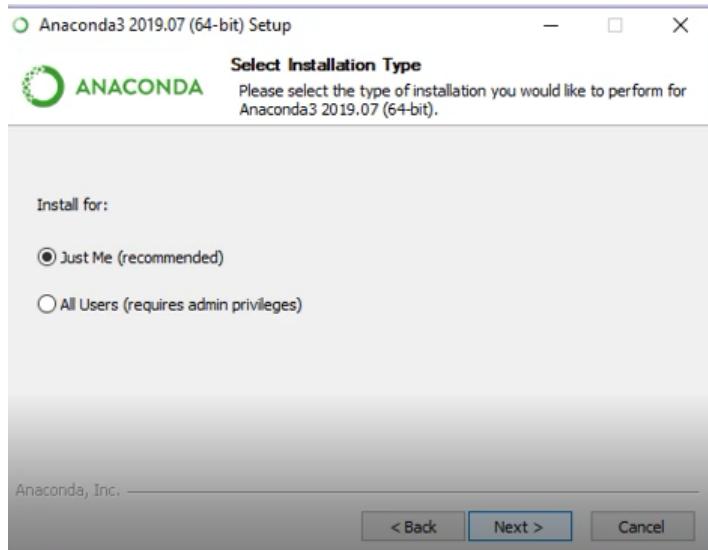
**Gambar 4.3** Welcome to Anaconda Setup

4. Pada *License Agreement* klik *I Agree* karena jika teman-teman tidak menyetujui lisensi anaconda maka teman-teman tidak akan bisa melanjutkan proses instalasi. lakukan langkah ini seperti pada gambar 4.4



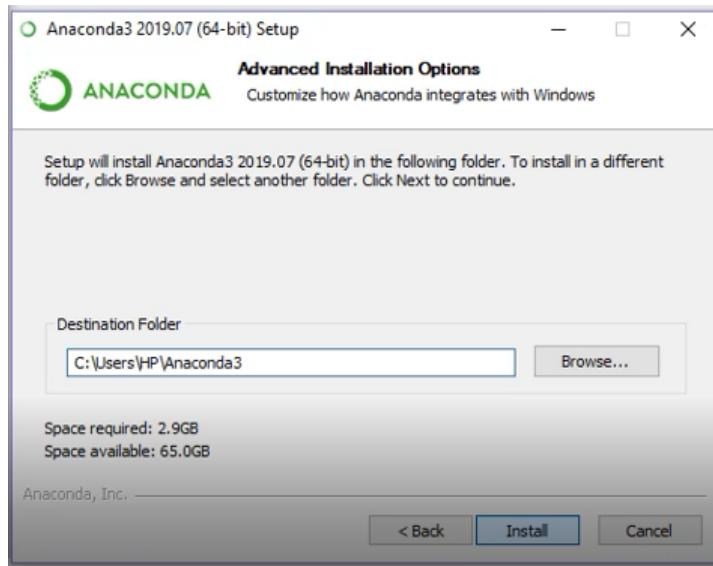
**Gambar 4.4** License Agreement

5. Kemudian pilih *Just Me(Recomended)* agar sesuai dengan komputer yang digunakan, kemudian klik *next* seperti pada gambar 4.5.



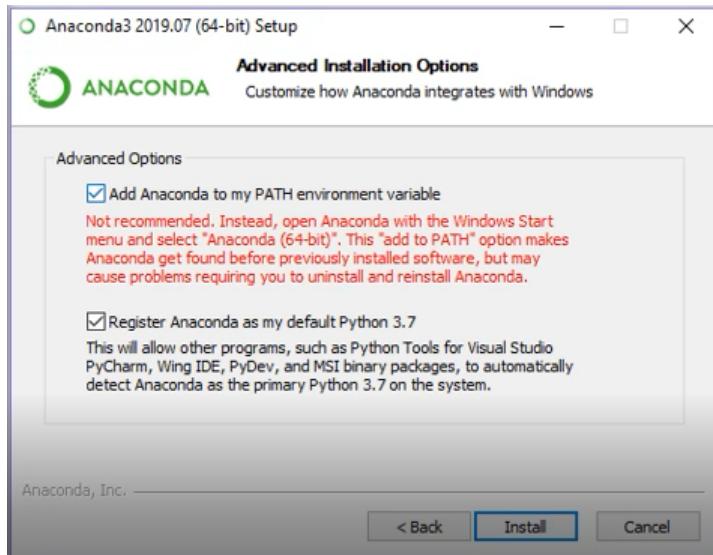
**Gambar 4.5** *Just Me(recomended)*

6. Kemudian pilih direktori tempat kita akan *menginstall anaconda* seperti pada gambar 4.6



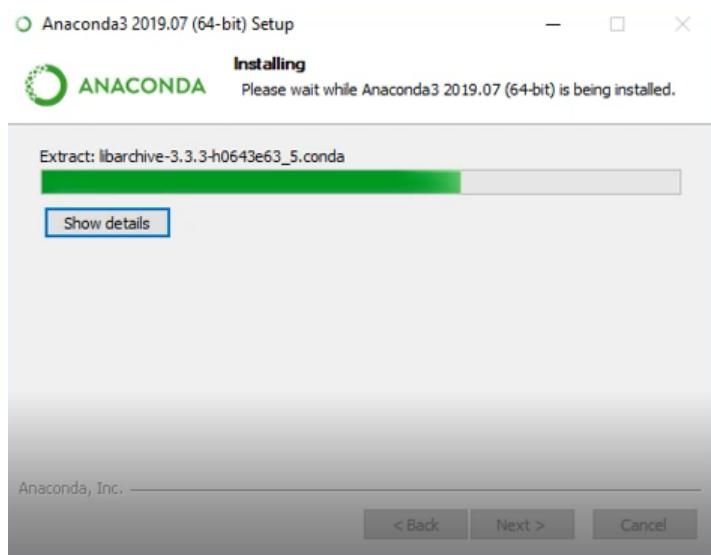
**Gambar 4.6** Pilih lokasi

- Kemudian centang *Add Anaconda to my Path environment variable*, agar saat melakukan instalasi package anaconda, package tersebut akan langsung tertuju ke path anaconda tidak ke aplikasi yang lain. kemudian Klik *install* seperti pada gambar 4.7.



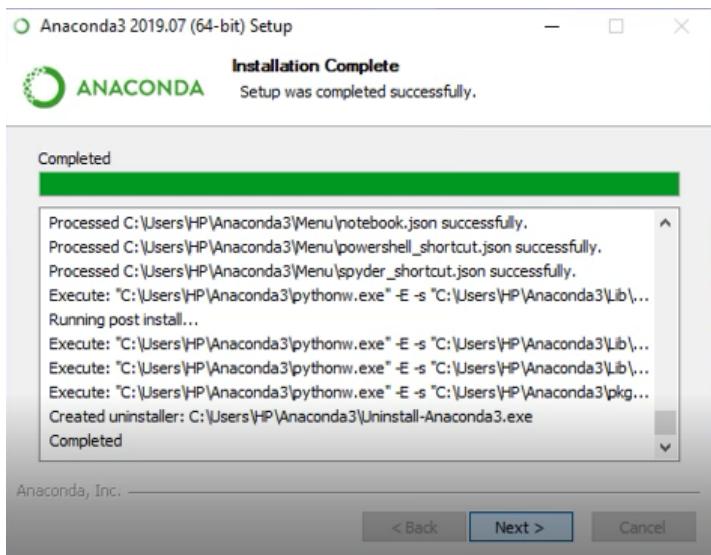
**Gambar 4.7** Centang Anaconda to my PATH

8. Tunggu sampai proses *installasi* selesai seperti pada gambar 4.8.



Gambar 4.8 Waiting Installation Complete

9. Apabila instalasi telah selesai maka akan terlihat seperti gambar 4.9, kemudian klik *next*



Gambar 4.9 Installation Complete

10. apabila muncul gambar 4.10, maka klik *next*



Gambar 4.10    *Anaconda+JetBrains*

11. Jika instalasi telah selesai maka akan ada ucapan terima kasih telah menginstall anaconda 3 seperti pada gambar 4.11, hal ini menandakan bahwa teman-teman telah selesai dan berhasil melakukan instalasi anaconda. Kemudian klik *finish* untuk mengakhiri instalasi.



**Gambar 4.11** *Thanks for install Anaconda*

#### 4.1.2 Update Anaconda dan Spyder

Kenapa kita harus melakukan update anaconda dan spyder? melakukan update diperlukan agar software yang kita gunakan merupakan software yang terbaru, karena versi lama dan versi baru akan memiliki banyak perbedaan dan akan menjadi masalah nantinya ketika kita membuat program atau mengimportkan modul-modul python yang digunakan. Berikut cara mengupdate Spyder:

1. Buka anaconda prompt, lalu ketikkan perintah conda update spyder
2. Konfirmasi update dengan mengetikkan y, lalu tekan enter
3. Tunggu hingga installan selesai

Berikut cara mengupdate Anaconda:

1. Buka anaconda prompt, lalu ketikkan perintah conda update anaconda
2. Konfirmasi update anaconda dengan mengetikkan y dan kemudian tekan enter
3. Tunggu hingga installan selesai

#### 4.1.3 Instalasi Anaconda Ubuntu 19.04

Untuk instalasi python pada Ubuntu 19.04 dibutuhkan sebagai berikut:

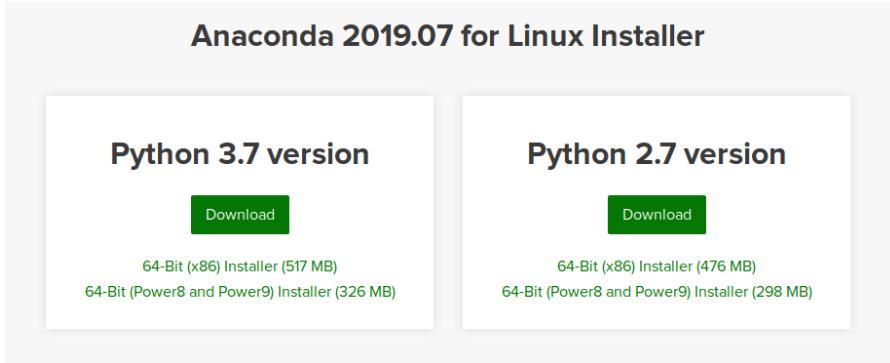
1. Internet

2. Anaconda installer (64bit or 32bit)

3. enter, dan yes atau no

Ikuti langkah berikut:

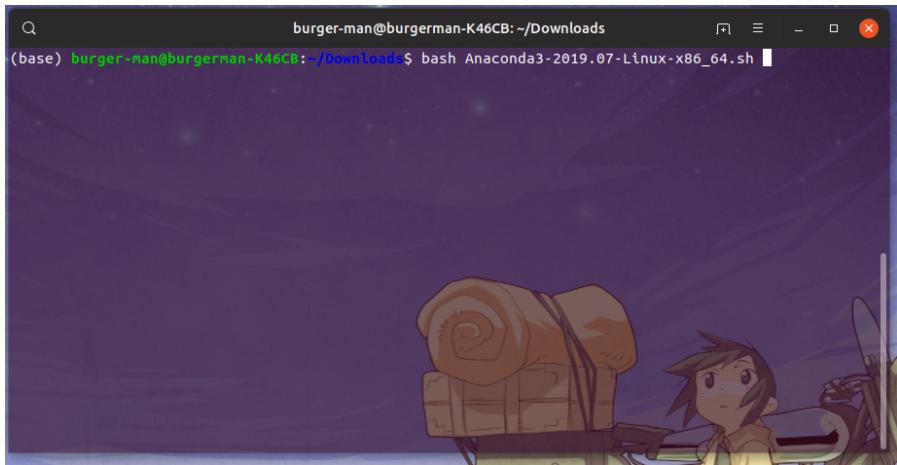
- Pertama kita kunjungi situs <https://www.anaconda.com/distribution/#download-section> seperti gambar 4.12 dan pilih **64-Bit (x86) Installer (517 MB)**



**Gambar 4.12** Gambar halaman download

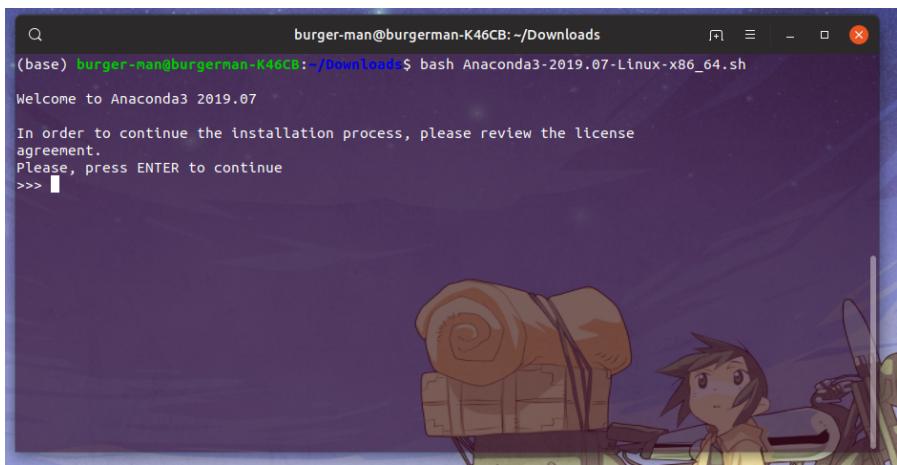
2. Kedua kita buka **terminal** kita lalu arahkan ke direktori kita menyimpan file download anaconda

3. Ketiga kita ketikkan sebagai berikut **bash namafileanaconda.sh** lalu enter, contoh seberti gambar 4.13



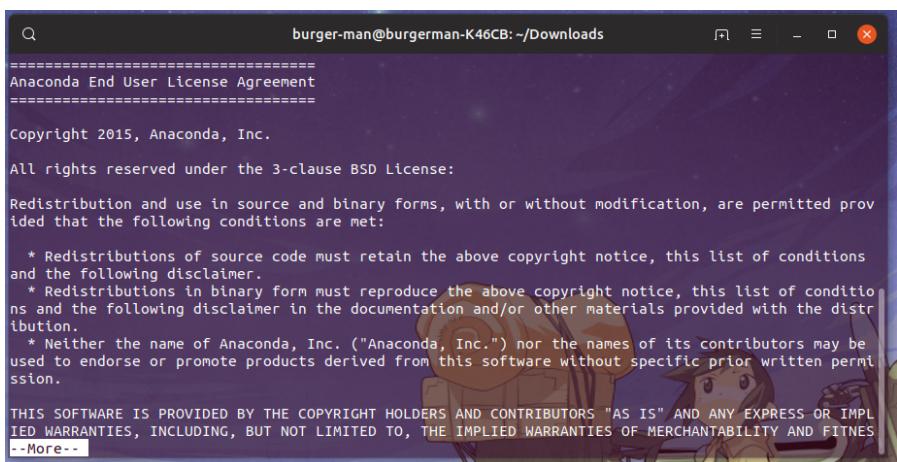
**Gambar 4.13** Gambar install anaconda

- Setelah itu, tekan **ENTER** saja seperti gambar 4.14



**Gambar 4.14** Gambar eksekusi anaconda

- Lalu akan muncul sebuah tulisan **End User License Agreement** seperti gambar 4.15, tekan **ENTER** dan tahan hingga seperti gambar



**Gambar 4.15** Gambar anaconda license agreement

**Gambar 4.16** Gambar perintah yes or no

- Lalu setelah muncur perintah '**yes**' or '**no**' ketik **yes** lalu enter
  - Setelah itu muncul path direktori instalasi anaconda kita seperti gambar 4.17 lalu tekan enter

```
Q burger-man@burgerman-K46CB: ~/Downloads
Please answer 'yes' or 'no':'
>>>
Please answer 'yes' or 'no':'
>>> yes

Anaconda3 will now be installed into this location:
/home/burger-man/anaconda3

- Press ENTER to confirm the location
- Press CTRL-C to abort the installation
- Or specify a different location below

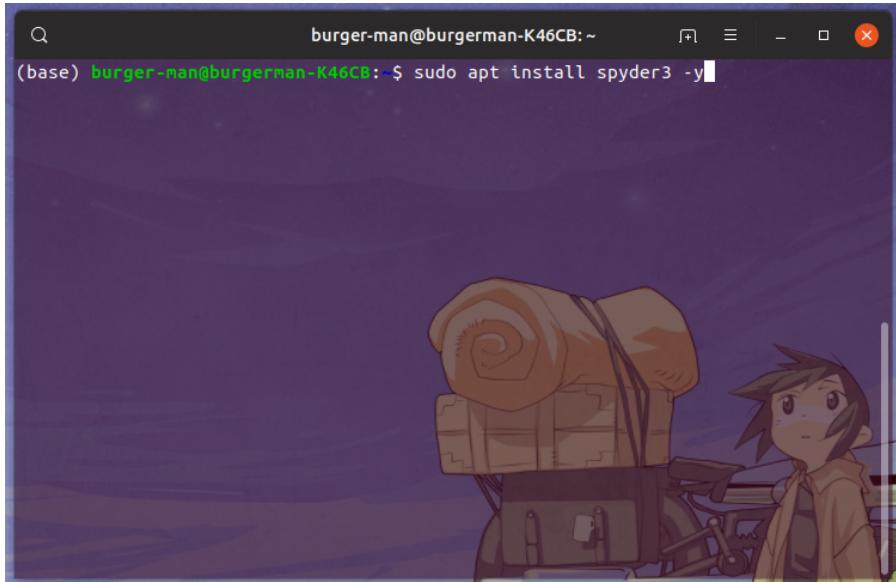
[~/home/burger-man/anaconda3] >>> [REDACTED]
```



**Gambar 4.17** Gambar path anaconda

Setelah kita selesai instalasi anaconda jangan lupa juga untuk menginstal spyder ide, caranya seperti berikut:

- (a) ketikkan perintah `sudo apt install spyder3 -y` seperti gambar 4.18



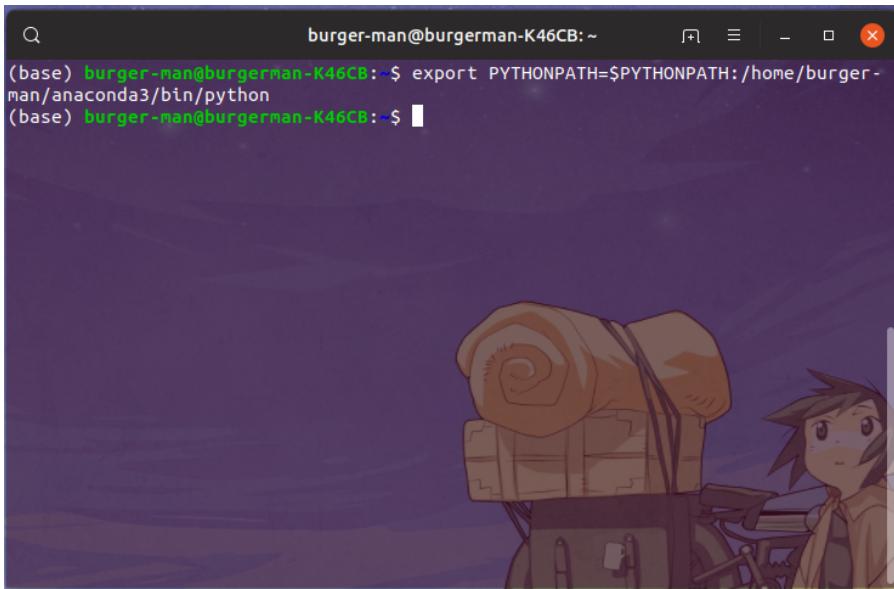
**Gambar 4.18** Gambar perintah install spyder3

(b) lalu jalankan dengan perintah **spyder** atau **spyder3**

#### 4.1.4 Konfigurasi *Python*

Setelah kita selesai instal Anaconda dan Spyder, selanjutnya kita akan mempelajari bagaimana cara setting environments python kita? caranya sebagai berikut

1. pertama kita buka terminal kita lalu ketikkan perintah **export PYTHONPATH=\$PYTHONPATH:\$PWD** contoh seperti gambar 4.19, lalu enter



**Gambar 4.19** Gambar setpath

#### 4.1.5 Quiz 2

1. Sistem operasi apa yang digunakan untuk instalasi pada buku ini?
  - a) Ubuntu dan Windows
  - b) Usus buntu dan Jendela
  - c) Penguin dan Kaca
  - d) Hewan dan Benda
2. Apa yang harus diperhatikan ketika ingin download Anaconda?
  - a) Kamu
  - b) Dia
  - c) Aku
  - d) Versi Sistem Operasi
3. Ubuntu versi berapa pada instalasi buku ini??
  - a) 19.04
  - b) 19.10
  - c) 19.99
  - d) 19.90
4. Apakah berbeda instalasi Anaconda pada windows dan linux? a) Ya  
b) Tidak

5. Apa yang kita memerlukan internet untuk instal Anaconda?
  - a) Ya
  - b) Tidak

## BAB 5

---

# INSTALASI PIP DAN PERINTAH CLI

---

### 5.1 Pip

Pip merupakan modul atau paket yang harus kita miliki apabila kita menggunakan bahasa pemrograman python. Pip digunakan untuk menginstal package-package python yang akan kita gunakan dalam pembuatan kode program.

#### 5.1.1 Instalasi Pip

Cara melakukan instalasi pip pada anaconda CLI:

1. buka anaconda prompt (Start -> Anaconda Prompt)
2. ketikkan conda install -c anaconda pip seperti pada gambar 5.1.

```

Anaconda Prompt (Anaconda3)

(base) C:\Users\trian>conda install -c anaconda pip
Collecting package metadata (current_repodata.json): done
Solving environment: done

=> WARNING: A newer version of conda exists. <=#
  current version: 4.7.10
  latest version: 4.7.12

Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: C:\Users\trian\Anaconda3

added / updated specs:
- pip

The following packages will be downloaded:

  package          | build
ca-certificates-2019.5.15 |      0    166 KB  anaconda
certifi-2019.6.16        | py37_0   155 KB  anaconda

Total:          166 KB

```

**Gambar 5.1** *Install pip*

3. ketik y, lalu enter. Tunggu hingga proses instalasi selesai seperti pada gambar 5.2.

```

Anaconda Prompt (Anaconda3)

Total:          10.9 MB

The following packages will be UPDATED:
  conda           pkgs/main::conda-4.7.10-py37_0 --> anaconda:::conda-4.7.12-py37_0

The following packages will be SUPERSEDED by a higher-priority channel:
  ca-certificates          pkgs/main --> anaconda
  certifi                   pkgs/main --> anaconda
  openssl                  pkgs/main --> anaconda
  pip                      pkgs/main --> anaconda

Proceed ([y]/n)? y

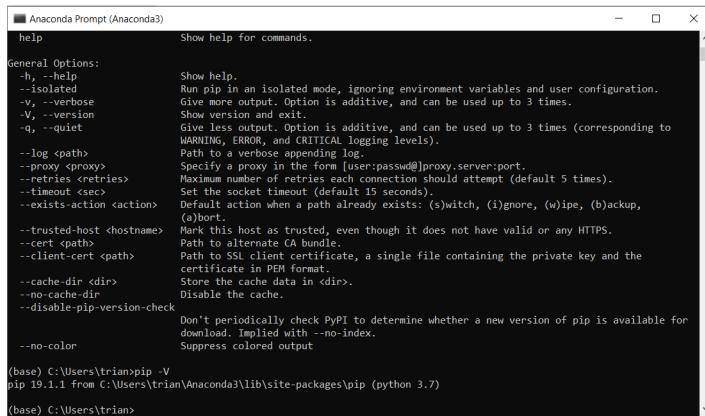
Downloading and Extracting Packages
openssl-1.1.1c           | 5.7 MB | #####| #####| #####| #####| 100%
certifi-2019.6.16         | 155 KB | #####| #####| #####| #####| 100%
ca-certificates-2019       | 166 KB | #####| #####| #####| #####| 100%
pip-19.1.1                | 1.8 MB | #####| #####| #####| #####| 100%
conda-4.7.12              | 3.0 MB | #####| #####| #####| #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

(base) C:\Users\trian>

```

**Gambar 5.2** *Install pip Selesai*

4. jika telah selesai, lakukan pengecekan versi pip dengan mengetikkan pip -V seperti pada gambar 5.3.



```

Anaconda Prompt (Anaconda3)
help           Show help for commands.

General Options:
-h, --help      Show help.
--isolated     Run pip in an isolated mode, ignoring environment variables and user configuration.
-v, --verbose   Give more output. Option is additive, and can be used up to 3 times.
-V, --version   Shows version and exits.
-q, --quiet     Give less output. Option is additive, and can be used up to 3 times (corresponding to
                WARNING, ERROR, and CRITICAL logging levels).
--log <path>   Path to a verbose appending log.
--proxy <proxy> Specify a proxy in the form [user:password@]proxy.server:port.
--retries <retries> Maximum number of retries each connection should attempt (default 5 times).
--timeout <sec> Set the socket timeout (default 5 seconds).
--exists-action <action> Default action when a path already exists: (s)witch, (i)gnore, (w)ipe, (b)ackup,
                    (a)bort.
--trusted-host <hostname> Mark this host as trusted, even though it does not have valid or any HTTPS.
--cert <path>   Path to alternative CA bundle.
--client-cert <path> Path to SSL client certificate, a single file containing the private key and the
                  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 whether a new version of pip is 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>

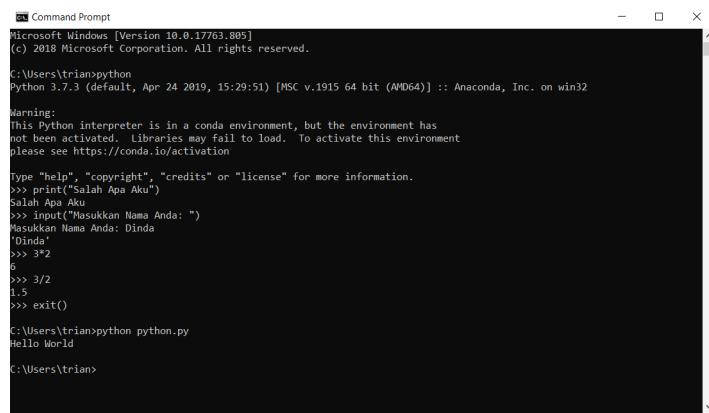
```

**Gambar 5.3 Melihat Versi pip**

### 5.1.2 Command Line Interface/Interpreter

Command line interpreter atau yang biasa teman-teman ketahui sebagai command prompt memungkinkan kita untuk menuliskan baris perintah yang akan dijalankan oleh komputer. Command line interpreter hanya berupa script atau tulisan kode program, berbeda dengan graphic user interface atau GUI yang memungkinkan kita memerintahkan sesuatu kepada komputer dengan hanya menggunakan tombol-tombol dan tampilan yang mudah dipahami. Cara menjalankan program python pada CLI:

1. Buka command prompt lalu ketikkan python seperti pada gambar 5.4.
2. Buatlah perintah print, input, perkalian, dan pembagian seperti pada gambar 5.4.
3. Bisa juga menjalankan file .py yang telah dibuat di IDE dengan cara python namafile.py, lalu klik enter seperti pada gambar 5.4.



The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The window displays the following text:

```
Microsoft Windows [Version 10.0.17763.805]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\trian>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

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
9
>>> 3/2
1.5
>>> exit()

C:\Users\trian>python python.py
Hello World

C:\Users\trian>
```

**Gambar 5.4** *CLI in Command Prompt*