

**BUKU TUTORIAL OBJECT DETECTION**

**“DETEKSI OBJEK DAN PENGENALAN KARAKTER  
PLAT NOMOR KENDARAAN INDONESIA BERBASIS  
PYTHON”**

Buku ini dibuat untuk memenuhi persyaratan kelulusan  
matakuliah Program Internship I



Dibuat Oleh,  
**1.16.4.046 Miftahul Hasanah**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA  
POLITEKNIK POS INDONESIA  
BANDUNG  
2020**

## **KATA PENGANTAR**

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, hidayah serta kekuatan sehingga Buku tutorial ini dengan judul “Deteksi Objek Dan Pengenalan Karakter Plat Nomor Kendaraan Indonesia Menggunakan *Convolutional Neural Network* Berbasis *Python*” dapat terselesaikan.

Banyak kendala yang dihadapi dalam penyusunan Buku ini dan penulis menyadari bahwa penyusunan buku ini masih belum sempurna. Ini mengingat keterbatasan pengetahuan, pengalaman serta kemampuan penulis. Penulis megharapkan kritik dan saran yang sifatnya membangun dari pembaca. Untuk itu penulis mengucapkan terima kasih kepada:

1. Allah SWT, karena dengan Rahmat dan Ridho-Nya penulis dapat menyelesaikan buku intership I.
2. Kedua orang tua dan keluarga penulis yang telah mendorong dan memberi semangat kepada penulis.
3. Nisa Hanum Harani, S.Kom., M.T. selaku dosen pembimbing internship I.
4. Nisa Hanum Harani, S.Kom., M.T. selaku Koordinator Internship I.
5. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Prodi DIV Teknik Informatika.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga buku ini membawa manfaat bagi pengembangan ilmu.

Bandung, 17 Januari

2020

Penulis

# DAFTAR ISI

---

KATA PENGANTAR .....	i
DAFTAR ISI.....	iii
DAFTAR GAMBAR.....	viii
BAB I.....	<b>Error! Bookmark not defined.</b>
1.1 Pengantar <i>Artificial Intelligence</i> .....	1
1.1.1 Definisi <i>Artificial Intelligence</i> .....	1
1.1.2 Sejarah <i>Artificial Intelligence</i> .....	5
1.1.3 Konsep Dasar Artificial Intelligence (Kecerdasan Buatan).....	8
1.1.4 Disiplin Ilmu <i>Artificial Intelligence</i> .....	9
1.1.5 Manfaat <i>Artificial Intelligence</i> .....	10
1.1.6 Kelebihan <i>Artificial Intelligence</i> .....	13
1.1.7 Kelemahan <i>Artificial Intelligence</i> .....	14
1.2 Pengantar <i>Machine Learning</i> .....	15
1.2.1 Definisi <i>Machine Learning</i> .....	15
1.2.2 Sejarah <i>Machine Learning</i> .....	17
1.2.3 Perkembangan <i>Machine Learning</i> .....	18
1.2.4 Bagaimana <i>Machine Learning</i> Bekerja? .....	19
1.2.5 Konsep Dasar <i>Machine Learning</i> .....	21
1.2.6 Penerapan <i>Machine Learning</i> .....	24
1.2.7 Metode Algoritma <i>Machine Learning</i> .....	26
1.2.8 Dampak <i>Machine Learning</i> di Masyarakat .....	27
1.3 Pengantar <i>Neural Network</i> .....	29
1.3.1 Definisi Neural Network.....	29
1.3.2 Sejarah <i>Neural Network</i> .....	30
1.3.3 Konsep <i>Neural Network</i> .....	32

1.1.4 Algoritma Dalam Neural Network.....	36
1.1.5 Arsitektur Neural Network.....	37
1.4 Pengantar <i>Deep Learning</i> .....	39
1.4.1 Definisi <i>Deep Learning</i> .....	39
1.4.2 Sejarah <i>Deep Learning</i> .....	41
1.4.3 Perbedaan <i>Neural Network</i> Dengan <i>Deep Learning</i> .....	42
1.4.4 Jenis-Jenis Algorithma <i>Deep Learning</i> .....	43
BAB II .....	44
2.1 Pengantar <i>Convolutional Neural Network</i> (CNN) .....	1
2.1.1. Definisi <i>Convolutional Neural Network</i> .....	1
2.1.2 Sejarah <i>Convolutional Neural Network</i> .....	4
2.1.2.1 Input Layer.....	7
2.1.2.2 Convolution Layer.....	7
2.1.2.3 Pooling Layer .....	11
2.1.2.4 Activation Function Layer.....	13
2.2 Pengantar <i>Region - Convolution Neural Network</i> (R-CNN) .....	13
2.2.1 Region Proposal.....	14
2.2.1 Masalah Dengan R-CNN .....	16
2.3 Pengantar Fast Region - Convolution Neural Network (Faster R-CNN) .....	17
2.4.1 Faster Region - Convolution Neural Network (Faster R-CNN)	
19	
BAB III .....	25
3.1 Tools Yang Digunakan .....	30
3.1.1 Anaconda .....	30
3.1.2 LabelImg .....	34
3.1.3 Sublime .....	36
3.1.4 CUDA .....	39

3.2 Bahasa Pemrograman Yang Digunakan.....	43
3.2.1 <i>Python</i> .....	43
3.1.2.1 Pengertian Python .....	43
3.1.2.2 Sejarah Python.....	44
3.1.2.3 Aplikasi Penggunaan Python.....	45
BAB IV .....	48
4.1 <i>Tools</i> Yang Digunakan .....	48
4.1.1 Instalasi Anaconda.....	48
4.1.2 Instalasi LabelImg.....	56
4.1.3 Instalasi Sublime.....	58
4.1.4 Instalasi CUDA .....	62
4.2 Bahasa Pemrograman Yang Digunakan .....	65
4.2.1 Instalasi <i>Python</i> .....	65
BAB V.....	74
5.1 Membuat Environment Pada Anacoda.....	75
5.2 Library Tensorflow .....	79
5.2.1 Pengertian Tensor flow .....	79
5.2.2 Cara Instalasi Tensor flow .....	82
5.2.3 Kelebihan Tensorflow .....	84
5.3 Library Pandas .....	85
5.3.1 Pengertian Pandas.....	85
5.3.2 Struktur Data Pandas.....	88
5.3.3 Cara Instalasi Pandas.....	89
5.4 Library <i>Numpy</i> .....	91
5.4.1 Pengertian Numpy .....	91
5.4.2 Kelebihan dan Kekurangan Numpy .....	92
5.4.3 Cara Instalasi Numpy .....	93
5.5 Library <i>OpenCV</i> .....	93

5.5.1 Pengertian OpenCV.....	93
5.5.2 Sejarah OpenCV .....	95
5.5.3 Fitur utama dari OpenCV .....	96
5.5.4 Kelebihan Dan Kekurangan OpenCV .....	98
5.5.5 Cara Instalasi OpenCV .....	98
5.6 Library PyTorch .....	101
5.6.1 Pengertian PyTorch.....	101
5.6.2 Cara Instalasi PyTorch.....	101
5.7 Library Theano.....	103
5.7.1 Pengertian Theano .....	103
5.7.2 Cara Instalasi Theano .....	103
5.8 Library Keras .....	105
5.8.1 Pengertian Keras.....	105
5.8.1 Cara Instalasi Keras .....	107
5.9 Library Sklearn .....	108
5.9.1 Pengertian Sklearn.....	108
5.9.2 Cara Instalasi Sklearn.....	109
5.10 Library Matplotlib .....	109
5.10.1 Pengertian Matplotlib.....	109
5.10.2 Cara Instalasi Matplotlib.....	111
5.11 Library Cudnn .....	112
5.11.1 Pengertian Cudnn.....	112
5.11.2 Cara Instalasi Cudnn .....	113
5.12 Library Pillow .....	118
5.12.1 Pengertian Pillow .....	118
5.12.2 Instalasi Pillow.....	119
5.13 Library protobuf.....	121
5.13.1 Pengertian Protobuf.....	121

5.13.2 Instalasi Pillow.....	121
5.14 Library Tensorboard.....	122
5.14.1 Pengertian Tensorboard .....	122
5.14.2 Instalasi Tensor board .....	123
5.15 Metodologi <i>CRISP-DM</i> .....	124
5.15.1 Pengertian CRISP-DM .....	124
5.1.5.2 Tahapan-Tahapan Cross-Industry Process For Data Mining	
	126

# **DAFTAR GAMBAR**

---

Gambar 1. 1 Perceptron .....	31
Gambar 1. 2 Struktur Neuron pada otak manusia.....	33
Gambar 1. 3 Struktur ANN .....	34
Gambar 1. 4 Arsitektur Neural Network.....	38
Gambar 1.5 Perbedaan antara lapisan layer pada Jaringan Saraf Tiruan(Neural Network) dengan Jaringan Deep Learning .....	42
Gambar 2. 1 Arsitektur MLP Sederhana .....	2
Gambar 2. 2 Proses Konvolusi pada CNN .....	3
Gambar 2. 3 Perbedaan arsitektur antara Neural Network pada umumnya dengan CNN.....	4
Gambar 2. 4 Arsitektur CNN .....	6
Gambar 2. 5 Ilustrasi Input RGB .....	7
Gambar 2. 6 Image RGB Convolutional Layer .....	8
Gambar 2. 7 Operasi Convolution Matrix dot.....	9
Gambar 2. 8 Zero Padding pada Matrix 4x4 menjadi Matrix 6x6 .....	10
Gambar 2. 9 Maximum Pooling Layer .....	12
Gambar 2. 10 Average pooling layer .....	12
Gambar 2. 11 Rumus ReLU .....	13
Gambar 2. 12 Arsitektur R-CNN .....	15
Gambar 2. 13 Proses R-CNN .....	16
Gambar 2. 14 Fast R-CNN.....	17
Gambar 2. 15 Perbandingan Algoritma Objek Deteksi.....	18
Gambar 2. 16 Arsitektur Faster R-CNN .....	19
Gambar 2. 17 Perbandingan Kecepatan Uji Waktu Dari Algoritma Objek Deteksi .....	20

Gambar 2. 18 Faster R-CNN .....	21
Gambar 2. 19 Modul Pertama Inception V2.....	23
Gambar 2. 20 Modul kedua Inception V2 .....	24
Gambar 2. 21 Modul Ketiga Inception V2 .....	25
Gambar 2. 22 Region Proposal Network (RPN) .....	25
Gambar 3. 1 Logo Anaconda .....	30
Gambar 3. 2 grafik implementasi matplotlib pada Python.....	31
Gambar 3. 3 Tampilan Anaconda Navigator.....	32
Gambar 3. 4 Tampilan Anaconda Desktop.....	33
Gambar 3. 5 Logo LabelImg.....	34
Gambar 3. 6 Tampilan LabelImg .....	35
Gambar 3. 7 Logo Sublime Text.....	36
Gambar 3. 8 Tampilan Sublime Text.....	37
Gambar 3. 9 Aliran Pemrosesan CUDA.....	43
Gambar 3. 10 Logo Python.....	43

## **DAFTAR TABEL**

---

## **BAB 1**

---

# **PENGENALAN ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NEURAL NETWORK, DAN DEEP Learning**

---

*Artificial Intelligence, Neural Network, Machine Learning* dan *Deep Learning* merupakan suatu istilah di dalam dunia teknologi informasi yang sangat popular pada saat ini. Pada penulisan buku ini penulis akan menjelaskan materi tentang teknologi tersebut dengan penerapan teknologi *Deep Learning* yaitu pada *object detection* dan pengenalan karakter pada plat nomor kendaraan Indonesia berbasis *Python*.

### **1.1 Artificial Intelligence**

#### **1.1.1 Definisi Artificial Intelligence**

Kecerdasan buatan atau biasa disebut dengan *Artificial Intelligence* merupakan kecerdasan yang ditambahkan pada suatu sistem yang bisa diatur dalam lingkungan ilmiah atau bisa disebut juga intelegensi artifisial (dalam bahasa Inggris dapat disebut dengan *Artificial Intelligence*) atau hanya disingkat AI,

didefinisikan sebagai kecerdasan entitas ilmiah. Andreas Kaplan dan Michael Haenlein menjelaskan kecerdasan buatan merupakan “sebuah kemampuan suatu sistem untuk dapat menganalisis data eksternal secara benar, untuk belajar dari data tersebut, dan menggunakan pembelajaran tersebut untuk mencapai tujuan dan tugas tertentu melalui penyesuaian yang fleksibel”.

Berikut ini merupakan penjelasan kecerdasan buatan bagi para ahli:

- **John McCarthy, 1956**

Menurut John McCarthy, kecerdasan buatan merupakan sebuah usaha untuk memodelkan proses berpikir manusia dan merancang suatu mesin agar dapat menirukan perilaku manusia.

- **Herbert Simon, 1987**

Menurut Herbert Simon, kecerdasan buatan merupakan suatu tempat yang digunakan untuk suatu penelitian, aplikasi dan instruksi yang terpaut dengan pemrograman komputer dalam melaksanakan suatu hal yang menurut pandangan manusia – cerdas.

- **Rich dan Knight, 1991**

Menurut Rich dan Knight, kecerdasan buatan merupakan suatu pembelajaran tentang bagaimana sebuah komputer dapat melakukan hal-hal yang pada saat ini masih bisa dikerjakan lebih baik oleh manusia.

Pengertian lain dari kecerdasan buatan merupakan suatu bagian dari ilmu komputer supaya mesin komputer dapat menyelesaikan suatu pekerjaan seperti dan sebaik pekerjaan

yang dilakukan manusia. Pada awal mulanya komputer diciptakan hanya berfungsi sebagai alat hitung saja. Namun seiring berkembangnya zaman, maka fungsi komputer juga semakin mendominasi kehidupan manusia. Saat ini komputer dapat digunakan untuk menangani sesuatu yang dapat dikerjakan oleh manusia. Supaya komputer bisa bekerja seperti manusia maka komputer juga harus diberi bekal keterampilan dan mempunyai kinerja untuk berpikir.

Sistem ini umumnya dianggap komputer. Kecerdasan diciptakan lalu dimasukkan ke dalam sebuah mesin (komputer) supaya dapat menyelesaikan pekerjaan seperti yang dilakukan oleh manusia. Beberapa kategori aspek yang menerapkan kecerdasan buatan antara lain yaitu seperti pada sistem pakar, pada permainan komputer (*games*), logika *fuzzy*, jaringan saraf tiruan dan robotika.

Banyak hal yang kelihatannya rumit bagi kecerdasan manusia, tetapi bagi Informatika relatif tidak bermasalah. Contohnya mentransformasikan persamaan, memecahkan persamaan integral, menciptakan permainan catur atau *Backgammon*. Di lain sisi hal yang bagi manusia tampaknya menuntut sedikit kecerdasan, hingga saat ini masih sulit untuk diterapkan dalam Informatika. Contohnya seperti pada pengenalan Objek/Muka, bermain sepak bola. Teknologi AI menciptakan cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, studi dan adaptasi yang cerdas pada sebuah mesin.

Secara garis besar, *Artificial Intelligence* (AI) dibagi menjadi dua pemahaman yaitu pemahaman AI konvensional dan pemahaman kecerdasan komputasional. AI konvesional umumnya mengaitkan pada metode-metode yang pada saat ini diklasifikasikan pada pembelajaran mesin, yang ditandai dengan formalism dan analis *statistic*. Metode-metodenya meliputi:

1. Sistem pakar: mengimplementasikan kapabilitas penilaian untuk menggapai kesimpulan. Sebuah sistem pakar bisa mengerjakan sebagian besar informasi yang diketahui dan menyimpan kesimpulan-kesimpulan yang berawal pada informasi-informasi tersebut.
2. Pertimbangan berdasar kasus
3. Jaringan Bayesian
4. AI berpedoman pada tingkah laku: metode modular pada pembuatan sistem AI secara manual

Kecerdasan komputasional menyangkutkan pada peningkatan atau pada pembelajaran berulang, misalnya penalaran parameter seperti pada sistem koneksiis. Pembelajaran ini bersumber pada data empiris dan diasosiasikan dengan AI non-simbolis. AI yang tidak teratur dan perhitungan lunak. Metode-metode intinya yaitu meliputi:

1. Jaringan saraf: sistem dengan keahlian pengenalan pola yang sangat terampil
2. Sistem *fuzzy*: teknik-teknik yang digunakan untuk penilaian dibawah ketidakpastian, telah digunakan secara menyebar pada industri modern dan sistem kendali ciptaan konsumen

### 3. Komputasi evolusioner: mengimplementasikan konsep-konsep yang terinspirasi

Meskipun AI memiliki asosiasi fiksi ilmiah yang unggul, AI juga membangun sisi yang amat sangat penting di dalam ilmu komputer, berhubungan dengan kepribadian, pembelajaran dan penyesuaian yang cerdas pada sebuah mesin. Penyelidikan pada AI melibatkan penciptaan mesin yang berguna untuk mengotomatisasikan tugas-tugas yang memerlukan perilaku cerdas. Contohnya yaitu penyelesaian, perencanaan dan pencatatan, keahlian untuk merespons diagnosa dan pertanyaan customer, serta pemahaman mengenai tulisan tangan, suara dan wajah. Hal-hal seperti itu sudah menjadi disiplin ilmu khusus, yang memfokuskan ketertarikan pada pengadaan solusi masalah kehidupan yang nyata. Sistem AI sekarang ini sering digunakan pada bidang ekonomi, teknik, militer dan obat-obatan, seperti yang sudah dibangun pada beberapa aplikasi perangkat lunak komputer rumah dan video game.

#### **1.1.2 Sejarah *Artificial Intelligence***

Pada awal mula abad ke 17, René Descartes mengutarakan pendapat bahwa tubuh hewan bukanlah apa-apa tapi sebagai hanya mesin-mesin yang rumit. Blaise Pascal menciptakan sebuah mesin penghitung digital mekanis pertama pada tahun 1642. Pada abad ke-19, Charles Babbage dan Ada Lovelace bertugas pada mesin penghitung mekanis yang bisa diprogram.

Bertrand Russell dan Alfred North Whitehead juga membuat *Principia Mathematica*, yang merombak akal sehat secara formal.

Warren McCulloch dan Walter Pitts juga menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang menempatkan pondasi bagi jaringan syaraf.

Pada tahun 1950-an merupakan masa cara aktif pada AI. Program AI perdana yang berfungsi dicatat pada tahun 1951 untuk menyelesaikan pekerjaan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan skenario/teks yang telah ditulis oleh Christopher Strachey dan juga sebuah program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy menciptakan suatu istilah "kecerdasan buatan " pada pertemuan pertama yang telah disediakan untuk inti persoalan ini, pada 1956. Dia juga menciptakan bahasa pemrograman Lisp. Alan Turing juga menginformasikan "Turing test" sebagai sebuah langkah untuk mengoperasionalkan test kepribadian cerdas. Joseph Weizenbaum juga menciptakan ELIZA, sebuah chatterbot yang mengimplementasikan psikoterapi Rogerian.

Sepanjang pada tahun 1960-an dan juga pada tahun 1970-an, Joel Moses mempertunjukkan ketahanan penilaian simbolis untuk menggabungkan masalah di dalam program Macsyma, program berbasis pendidikan yang telah berhasil pertama kali di dalam bagian matematika. Marvin Minsky dan juga Seymour Papert telah menciptakan Perceptrons, yang telah mempertunjukkan batas jaringan syaraf sederhana selain itu Alain Colmerauer menguraikan bahasa komputer Prolog. Ted Shortliffe memperlihatkan kekebalan sistem berbasis aturan untuk representasi pengetahuan dan inferensi dalam diagnosa dan terapi

medis yang kadangkala disebut sebagai sistem pakar pertama. Hans Moravec mengembangkan kendaraan terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri.

Pada tahun 1980-an, jaringan syaraf digunakan secara meluas dengan algoritma perambatan balik, pertama kali diterangkan oleh Paul John Werbos pada 1974. Pada tahun 1982, para ahli fisika seperti Hopfield menggunakan teknik-teknik statistika untuk menganalisis sifat-sifat penyimpanan dan optimasi pada jaringan syaraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model jaringan syaraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritma pembelajaran propagansi balik (Back-Propagation learning). Algoritma ini berhasil diimplementasikan ke dalam ilmu komputer dan psikologi. Tahun 1990-an ditandai perolehan besar dalam berbagai bidang AI dan demonstrasi berbagai macam aplikasi. Lebih khusus Deep Blue, sebuah komputer permainan catur, mengalahkan Garry Kasparov dalam sebuah pertandingan 6 game yang terkenal pada tahun 1997. DARPA menyatakan bahwa biaya yang disimpan melalui penerapan metode AI untuk unit penjadwalan dalam Perang Teluk pertama telah mengganti seluruh investasi dalam penelitian AI sejak tahun 1950 pada pemerintah AS.

Tantangan Hebat DARPA, yang dimulai pada 2004 dan berlanjut hingga hari ini, adalah sebuah pacuan untuk hadiah \$2

juta dimana kendaraan dikemudikan sendiri tanpa komunikasi dengan manusia, menggunakan GPS, komputer dan susunan sensor yang canggih, melintasi beberapa ratus mil daerah gurun yang menantang.

### **1.1.3 Konsep Dasar Artificial Intelligence (Kecerdasan Buatan)**

Berikut ini 4 dasar kategori pada konsep dasar Ai (Kecerdasan Buatan)

#### *1. Acting Humanly*

*Acting humanly* ialah system yang melakukan pendekatan dengan menirukan tingkah laku seperti manusia yang dikenalkan pada tahun 1950 dengan cara kerja pengujian melalui *teletype* yaitu jika penguji (integrator) tidak dapat membedakan yang mengintrogasai antara manusia dan *computer* maka *computer* tersebut dikatakan lolos(menjadi kecerdasan buatan).

#### *2. Thinking Humanly*

Yaitu system yang dilakukan dengan cara intropesi yaitu penangkapan pemikiran psikologis Manusia pada *computer*, hal ini sering diujikan dengan *neuron* ke *neuron* lainnya atau sel otak dengan sel otak lainnya cara pembelajarannya yaitu melalui experiment-experimen.

#### *3. Thinking Rationally*

Ini merupakan *system* yang sangat sulit ,karena sering terjadi kesalah dala, prinsip dan praktiknya,system ini dikenal dengan penalaran komputasi.

#### *4. Acting Rationally*

Yaitu *system* yang melakukan aksi dengan cara menciptakan suatu robotika cerdas yang menggantikan tugas manusia.

#### **1.1.4 Disiplin Ilmu *Artificial Intelligence***

Seperti yang telah disebutkan di atas bahwa AI merupakan salah satu cabang Ilmu Komputer. Tapi karena kompleksitas area AI maka dibuat sub-sub bagian yang dapat berdiri sendiri dan dapat saling bekerja sama dengan sub bagian lain atau dengan disiplin ilmu lain. Berikut ini beberapa cabang ilmu sub bagian dari AI :

##### *1. Natural Languange Processing (NLP)*

*Natural Languange Processing* (NLP) atau Pemrosesan Bahasa Alami, merupakan salah satu cabang AI yang mempelajari pembuatan sistem untuk menerima masukan bahasa alami manusia. Dalam perkembangannya, NLP berusaha untuk mengubah bahasa alami komputer (*bit* dan *byte*) menjadi bahasa alami manusia yang dapat kita mengerti. NLP merupakan ilmu dasar yang dapat dijadikan jembatan untuk membuat komunikasi antara mesin dengan manusia.

##### *2. Expert System (ES)*

*Expert System* (ES) atau Sistem Pakar, merupakan salah satu cabang AI yang mempelajari pembuatan sebuah sistem yang dapat bekerja layaknya seorang pakar. ES dapat menyimpan pengetahuan seorang pakar dan memberikan solusi berdasarkan pengetahuan yang dimilikinya tadi. ES juga merupakan salah satu cabang AI yang sering melakukan kerja sama dengan disiplin ilmu lain karena sifatnya yang dapat menyimpan pengetahuan.

### *3. Pattern Recognition (PR)*

*Pattern Recognition* (PR) atau Pengenalan Pola, merupakan salah satu cabang AI yang mempelajari pembuatan sebuah sistem untuk dapat mengenali suatu pola tertentu. Misalnya sistem PR untuk mengenali huruf dari tulisan tangan, walaupun terdapat perbedaan penulisan huruf A dari masing-masing orang tetapi PR dapat mengenali bahwa huruf tersebut adalah huruf A. Beberapa aplikasi dari PR antara lain : *voice recognition*, *Fingerprint Identification*, *Face Identification*, *Handwriting Identification*, *Optical Character Recognition*, *Biological Slide Analysis*, *Robot Vision* dan lainnya.

### *4. Robotic*

*Robotic* atau Robotika, merupakan salah satu cabang AI yang menggabungkan cabang-cabang AI yang lain termasuk ketiga cabang di atas untuk membentuk sebuah sistem robotik. Keempat cabang AI di atas merupakan cabang umum yang banyak dipelajari, masih banyak cabang-cabang AI yang lainnya. Seiring perkembangan riset dalam AI, dapat dimungkinkan akan muncul cabang-cabang baru yang melengkapi unsur AI sehingga AI menjadi sebuah sistem lengkap dan akan mencapai goal-nya yang sampai sekarang masih belum sempurna.[1]

#### **1.1. 5 Manfaat Artificial Intelligence**

Manfaat adanya Kecerdasan adalah untuk mengembangkan metode dan sistem untuk menyelesaikan suatu masalah, yang mana masalah tersebut juga dapat diselesaikan oleh manusia. Misalnya pencarian tempat, bidang bisnis, rumah tangga

dan dapat meningkatkan kinerja sistem informasi yang berbasis komputer. Seperti contohnya robot cerdas. Robot ini dapat membantu pekerjaan manusia menggunakan sistem navigasi. Pada bidang kesehatan digunakan untuk merawat orang jompo, konsultasi, dan pendekripsi tumor, membantu melakukan operasi medis dll. Diperkirakan ada 90 startup yang mengembangkan AI di industri kesehatan.

Di dunia kesehatan, AI digunakan pada robot yang mengasistensi dokter. Contohnya, Microsoft telah mengembangkan AI yang dapat membantu para dokter untuk menentukan jenis obat dan perawatan apa yang ideal bagi pasien. Selain itu, ada juga Children's National Medical Center di Washington yang telah sukses menggunakan robot AI untuk membantu tindakan operasi ke pasien. Di rumah sakit, AI sangat membantu dalam hal diagnosis medis dan analisis detak jantung. Robot juga digunakan untuk merawat orang jompo, konsultasi, dan pendekripsi tumor. Diperkirakan ada 90 *startups* yang mengembangkan AI di insustri kesehatan ini.

Di dunia otomotif, perusahaan ternama yang terlibat dengan AI seperti Tesla, Google, Apple, telah melakukan penelitian mobil tanpa pengemudi. Beberapa ahli mengatakan bahwa ke depannya manusia akan terbiasa menggunakan kendaraan tanpa pengemudi.

Di industri ekonomi dan keuangan, AI membantu di berbagai hal seperti investasi keuangan, pencataan keuangan, jual beli saham, penipuan dan kriminalisasi di bank, dan lain-lain.

Di dunia penerbangan, AI digunakan untuk mensimulasikan penerbangan sehingga pilot bisa diasisten dengan memberikan informasi pergerakan yang terbaik, informasi keadaan udara dan tekanan, dan lain-lain.

Di dalam dunia pendidikan, tutor robot telah diperkenalkan di kelas untuk mengajar anak-anak mulai dari pelajaran biologi sampai dengan ilmu komputer, meskipun hal ini belum banyak dilakukan. *Machine Learning* pada AI digunakan untuk menilai hal-hal apa saja yang perlu diperbaiki pada masing-masing siswa dalam proses belajarnya.

Di bidang industri, robot sering dijumpai untuk mengantikan manusia, terutama dalam pekerjaan yang repetitif (berulang-ulang). Lo mungkin *udah* sering *liat* mesin-mesin otomatis di pabrik-pabrik yang membantu proses produksi sebuah produk.

Di bidang *Human Resources*, AI dimanfaatkan untuk proses perekrutan dan pengembangan sumber daya manusia. AI digunakan dalam menentukan pekerjaan yang cocok antara pelamar dan pencari kerja. Di tahun 2016 dan 2017, Unilever menggunakan AI untuk seleksi karyawan yang bersifat entry level (itu loh, posisi pekerjaan yang buat karyawan yang baru masuk *banget* dunia kerja yang *gak* butuh pengalaman *banget*). Unilever menggunakan ilmu neuroscience, analisis wajah dan suara dalam wawancara, dan lain-lain untuk mendapatkan kandidat karyawan yang terbaik. Unilever telah memotong waktu

kerja dari empat bulan menjadi empat minggu dan menghemat 50,000 jam para perekut. [2]

#### **1.1.6 Kelebihan *Artificial Intelligence***

Sangatlah banyak kelebihan yang bisa didapatkan dari teknologi tersebut. Berikut ini adalah beberapa kelebihan dari kecerdasan buatan yaitu:

##### **1. Bersifat permanen**

AI adalah kecerdasan buatan yang bersifat permanen karena memang bisa digunakan berulang-ulang, dimana saja, dan kapan saja. AI selalu dikaitkan dengan penggunaan tenaga manusia. Contohnya adalah dengan adanya AI, nantinya suatu perusahaan tidak akan membutuhkan banyak karyawan. Hal tersebut benar jika dikaitkan dengan keuntungan satu ini. Karyawan bisa keluar kapan saja, sedangkan AI tidak.

##### **2. Menawarkan kemudahan**

Kecerdasan manusia yang dibuat telah disimpan di AI sehingga AI akan memudahkan manusia. Data yang sebelumnya disimpan akan mudah diakses kembali. Kerja *Artificial Intelligence* juga lebih cepat dibanding kerja manusia.

##### **3. Bersifat konsisten dan teliti**

Selain lebih cepat kerjanya, AI juga lebih konsisten dan teliti. Kecerdasannya tidak berkurang dan peluang terjadinya kesalahan sangat kecil. Bisa anda lihat di penghitungan komputer. Sangat jarang terjadi kesalahan bukan? Komputer sangatlah konsisten dan teliti, berbeda dengan manusia.

##### **4. Dapat disimpan**

Seperti yang telah dijelaskan dalam rahasia kemampuan mengolah big data, sehingga menyimpan data sebesar dan sebanyak apapun tidak akan menjadi masalah. Arsip dan data tersebut tentunya bisa disimpan dan digunakan oleh generasi selanjutnya.

### **1.1.7 Kelemahan *Artificial Intelligence***

Buatan manusia pasti masih memiliki sedikit cela meskipun hanya sedikit. *Artificial Intelligence* sendiri memiliki kerugian seperti tidak memiliki *common sense* dan kecerdasannya terbatas. Maksud dari tidak memiliki *common sense* yaitu AI hanya mengolah data dan memutuskan sesuatu sesuai data yang dikumpulkan, berbeda dengan manusia yang masih mempertimbangkan naluri dalam mengambil suatu keputusan.

Selain itu ada informasi yang memang hanya bisa diproses dan dimengerti oleh manusia. Jadi jika nantinya ada suatu robot yang dibentuk menyerupai manusia, robot tersebut tetap tidak akan bisa memiliki insting manusia. Hal tersebut berkaitan dengan kerugian atau kelemahan yang kedua yaitu kecerdasannya terbatas.

Kecerdasan buatan yang nantinya digunakan di suatu alat teknologi. Alat tersebut hanya mampu melakukan pekerjaan sesuai dengan sistem AI yang dimasukkan ke dalamnya. Misalnya yaitu sistem yang dirancang khusus untuk mengenali suara dalam bahasa Indonesia tidak akan bisa mengenali suara dalam bahasa Korea.[3]

## **1.2 Machine Learning**

### **1.2.1 Definisi Machine Learning**

*Machine Learning* adalah metode analisis yang membantu menangani data besar dengan cara mengembangkan algoritma komputer. Dengan menggunakan data, pembelajaran mesin memungkinkan komputer menemukan wawasan tersembunyi tanpa diprogram secara eksplisit saat mencarinya. Dengan adanya *email* baru, algoritma tersebut kemudian akan menghasilkan prediksi apakah email baru itu spam atau tidak.

*Machine Learning* adalah aplikasi *Artificial Intelligence* (AI) yang menyediakan sistem kinerja secara otomatis serta belajar memperbaiki diri dari pengalaman tanpa diprogram secara eksplisit. Pembelajaran mesin berfokus pada pengembangan program komputer yang bisa mengakses data dan menggunakannya untuk belajar sendiri. Penerapan metode *Machine Learning* dalam beberapa tahun terakhir telah berkembang di mana-mana dalam kehidupan sehari-hari.

Proses pembelajaran dimulai dengan observasi data, seperti contoh: pengalaman langsung, atau intruksi untuk mencari pola data dan membuat keputusan yang lebih baik dimasa depan berdasarkan contoh tersebut. Tujuan utamanya adalah membiarkan komputer belajar secara otomatis tanpa intervensi atau bantuan manusia dan menyesuaikan aktivitas yang sesuai. Karena teknologi komputasi, *Machine Learning* saat ini tidak seperti *Machine Learning* di masa lalu.

Sementara, algoritma *Machine Learning* sudah ada sejak lama, kemampuannya secara otomatis menggunakan perhitungan matematis yang kompleks ke data besar dan yang lebih cepat merupakan perkembangan terakhir. *Machine Learning* merupakan salah satu cabang dari disiplin ilmu AI (*Artificial Intelligence*) atau *Kecerdasan Buatan* yang membahas mengenai pembangunan sistem yang berdasarkan data.

Sebenarnya, masih ada banyak lagi situs web dan perangkat modern yang mungkin besar dan berisi beberapa model *Machine Learning* yang mungkin tidak kita sadari. Model tersebut digunakan untuk melakukan klasifikasi atau prediksi terhadap data baru yang memungkinkan kita untuk membuat atau mendukung pengambilan keputusan. Berikut adalah beberapa contoh aplikasi pembelajaran mesin yang dipublikasikan secara luas:

- Mobil *Google* yang sangat *hyped* dan *self-driving*. Inti pembelajaran mesin.
- Penawaran rekomendasi online seperti *Amazon* dan *Netflix*. Aplikasi belajar mesin untuk kehidupan sehari-hari.
- Mengetahui apa yang pelanggan katakan tentang Anda di *Twitter*. Pembelajaran mesin dikombinasikan dengan pembuatan aturan linguistik.
- Deteksi penipuan Salah satu kegunaan yang lebih jelas dan penting di dunia kita saat ini.

*Machine Learning* meningkat karena faktor data mining dan analisis Bayesian lebih populer dari sebelumnya. Pengolahan

komputasi yang lebih murah dan lebih bertenaga termasuk penyimpanan data yang terjangkau merupakan peningkatan. Semua hal ini secara cepat dan otomatis menghasilkan model yang dapat menganalisis data yang lebih besar dan lebih kompleks memberikan hasil yang lebih cepat dan akurat dalam skala yang sangat besar.

Prediksi nilai tinggi bisa mengarah pada keputusan dan tindakan cerdas secara real-time tanpa campur tangan manusia. Salah satu kunci untuk menghasilkan gerakan cerdas secara real-time merupakan pembuatan model otomatis.[4]

### **1.2.2 Sejarah *Machine Learning***

*Machine Learning* bermula di awal abad 20, seorang penemu Spanyol, Torres y Quevedo, membuat sebuah mesin learning setelah ditemukannya komputer digital. Isilah *Machine Learning* pada dasarnya adalah proses komputer untuk belajar dari data (*Learn from data*). Tanpa adanya data, komputer tidak akan bisa belajar apa - apa. Oleh karena itu, jika kita ingin belajar *Machine Learning*, pasti akan terus berinteraksi dengan data. Semua pengetahuan *Machine Learning* pasti akan melibatkan data. Data bisa saja sama, akan tetapi algoritma dan pendekatannya berbeda-beda untuk mendapatkan hasil yang optimal.

*Machine Learning* adalah metode analisis data yang meng-otomatisasi pembuatan model analitis. Ini adalah cabang dari *Artificial Intelligence* (AI) yang berdasarkan pada ide bahwa mesin seharusnya bisa belajar dan beradaptasi melalui pengalaman. Fokus besar penelitian *Machine Learning* adalah bagaimana

menganalisa cara otomatis pola kompleks dan bisa membuat keputusan cerdas berdasarkan data. Mesin dalam hal ini memiliki arti "sistem".[5]

### **1.2.3 Perkembangan *Machine Learning***

Dengan berkembangnya teknologi kecerdasan buatan, muncul salah satu cabang kecerdasan buatan yang memperoleh banyak perhatian dari para peneliti yang disebut *Machine Learning*. *Machine Learning* mempelajari teori agar komputer mampu "belajar" dari data, *Machine Learning* melibatkan berbagai disiplin ilmu seperti statistika, ilmu komputer, matematika dan bahkan neurologi. Salah satu algoritma *Machine Learning* yang menarik adalah jaringan saraf tiruan, seperti namanya jaringan saraf tiruan terinspirasi dari cara kerja otak manusia ( yang disederhanakan ). Secara intuisi mencari inspirasi untuk membuat mesin mampu "berfikir" dari cara kerja otak adalah langkah yang bagus sama halnya seperti ingin membuat alat yang mampu terbang dengan melihat cara kerja burung terbang.

Dalam salah satu model jaringan saraf tiruan yang disebut MLP ( *Multi Layer Perceptron* ) dikenal istilah *layer*, beberapa *neuron* tiruan dikelompokan menjadi satu *layer* kemudian *layer* satu menjadi *input* bagi *layer* yang lain, MLP sebenarnya adalah model ( matematika ) yang terdiri dari komposisi-komposisi fungsi dari vektor ke vektor, model ini biasanya di-*train* menggunakan algortima optimisasi berbasis gradien seperti *gradient descent*, berbagai masalah muncul ketika model jaringan saraf tiruan memiliki banyak *layer*, salah satu

masalah yang terkenal disebut *the vanishing gradient*, masalah ini muncul karena jaringan saraf tiruan dengan banyak *layer* sebenarnya adalah fungsi yang terdiri dari banyak komposisi fungsi sehingga ketika menghitung gradien terhadap parameter dari fungsi tersebut, kita harus menggunakan aturan rantai yang menyebabkan gradien parameternya bernilai kecil sehingga algoritma *gradient descent* berjalan lambat.[6]

#### **1.2.4 Bagaimana *Machine Learning* Bekerja?**

*Machine Learning* memiliki dua jenis teknik: *Supervised Learning*, yang melatih model pada data *input* dan output yang diketahui sehingga dapat memprediksi keluaran masa depan dan *UnSupervised Learning*, yang menemukan pola tersembunyi atau struktur intrinsik pada data masukan.

Penerapan metode *Machine Learning* dalam beberapa tahun terakhir telah berkembang di mana-mana dalam kehidupan sehari-hari. *Machine Learning* bukanlah hal baru dalam lanskap ilmu komputer. *Machine Learning* mengaitkan proses struktural dimana setiap bagian menciptakan versi mesin yang lebih baik. Berikut adalah penjelasan dari *Supervised Learning* dan *UnSupervised Learning*:

##### **1. *Supervised Learning***

Pembelajaran mesin yang diawasi menciptakan model yang melancarkan prediksi berdasarkan bukti adanya ketidakpastian. Algoritma pembelajaran yang diawasi memerlukan seperangkat data masukan dan tanggapan yang diketahui terhadap data (output) dan melatih model untuk menghasilkan prediksi yang

masuk akal untuk respons terhadap data baru. Gunakan pembelajaran ini jika Anda ingin mengetahui data output yang ingin Anda prediksi. Pembelajaran ini diawasi menggunakan teknik klasifikasi dan regresi untuk mengembangkan model prediktif.

Teknik klasifikasi memprediksi respons diskrit – misalnya, apakah email itu asli atau spam, atau apakah tumor itu kanker atau tidak. Model klasifikasi mengklasifikasikan data masukan ke dalam kategori tersebut. Aplikasi yang umum termasuk pencitraan medis. Misalnya aplikasi untuk pengenalan tulisan, maka anda harus menggunakan klasifikasi untuk mengenali huruf dan angka.

Jika Anda bisa melakukannya, Anda memiliki landasan yang dapat Anda gunakan pada satu dataset ke dataset yang akan dicoba lagi selanjutnya. Anda bisa mengisi waktu seperti mempersiapkan data lebih lanjut dan memperbaiki hasilnya nanti, begitu Anda lebih percaya diri. Dalam pengolahan citra dan penglihatan komputer, teknik pengenalan pola tanpa pemeriksaan digunakan untuk deteksi objek dan segmentasi. *Algoritme* yang umum mengadakan klasifikasi yang meliputi dukungan mesin vektor (SVM).

## 2. *UnSupervised Learning*

Ini menemukan pola tersembunyi atau struktur intrinsik dalam data. Ini digunakan untuk menarik kesimpulan dari kumpulan data yang terdiri dari data masukan tanpa respon berlabel. Clustering adalah teknik belajar tanpa pengamatan yang umum. Ini digunakan untuk analisis data eksplorasi dalam

menemukan pola atau pengelompokan tertutup dalam data. Aplikasi untuk analisis cluster meliputi analisis urutan gen, riset pasar dan pengenalan objek.

Misalnya, jika sebuah perusahaan telepon seluler ingin mengoptimalkan lokasi di mana mereka membangun menara telepon seluler, mereka dapat menggunakan pembelajaran mesin untuk memperkirakan jumlah kelompok orang yang bergantung pada menara mereka. Telepon hanya bisa berbicara dengan satu menara sekaligus, sehingga tim menggunakan algoritma pengelompokan untuk merancang peletakan menara seluler terbaik dalam mengoptimalkan penerimaan sinyal bagi kelompok dan dari pelanggan mereka.

Algoritma yang umum mengadakan clustering meliputi k-means dan k-medoids, hirarki clustering, model campuran Gaussian, model Markov tersembunyi, peta pengorganisasian sendiri, clustering *fuzzy c-means* dan clustering subtraktif.

### **1.2.5 Konsep Dasar Machine Learning**

Konsep tersebut meliputi kemampuan suatu individu dalam meningkatkan kecerdasan tersebut untuk belajar tanpa terkecuali pada sebuah mesin. Mesin yang mampu belajar, akan meningkatkan produktivitas manusia. Maka ia juga akan memiliki kekuatan yang mungkin tidak dimiliki mesin lainnya.

#### **1. Manfaat pembelajaran mesin dalam memprediksi**

Jika Anda hanya mengenal wajah teman Anda dalam gambar, berarti Anda tidak memakai model *pembelajaran mesin*. Inti pembelajaran mesin adalah meramalkan hal-hal

berdasarkan pola dan faktor lain yang telah dilatih. Apa yang menjadikan sesuatu itu jadi lebih mudah untuk di kuasai. Pengenalan dilakukan dengan cara yang simple tanpa menghabiskan banyak waktu.

## 2. Pembelajaran mesin membutuhkan pelatihan

Anda harus memberi tahu model *pembelajaran mesin* apa yang akan diprediksikannya. Pikirkan bagaimana anak manusia belajar. Ini adalah penyederhanaan yang berlebihan sedikit karena saya meninggalkan bagian dimana Anda juga harus mengatakannya bahwa itu bukan pisang dan tunjukkan berbagai jenis pisang, warna yang berbeda, gambar dari perspektif dan sudut yang berbeda, dll.

## 3. Ketepatan 80% dianggap sukses

Teknologi ini tidak mengetahui dimana platform pembelajaran mesin akan mencapai akurasi 100% dengan mengidentifikasi pisang dalam gambar. Tapi tidak apa-apa, ternyata manusia juga tidak 100% akurat. Aturan yang dikatakan dalam industri ini adalah bahwa model dengan akurasi 80% adalah sebuah kesuksesan. Jika Anda memikirkan betapa bergunanya untuk mengidentifikasi 800.000 gambar dengan benar di koleksi Anda, sementara mungkin tidak mendapatkan 200.000 yang benar, Anda masih menyimpan 80% dari waktu Anda.

Itu merupakan perspektif nilai yang sangat besar. Jika saya bisa melambaikan tongkat sihir dan meningkatkan produktivitas

Anda sebanyak itu, anda akan memberi saya banyak uang. Nah, ternyata saya bisa melakukannya dengan mesin pembelajaran.

#### 4. Pembelajaran mesin berbeda dengan AI

Kebanyakan orang mengatakan hal ini sama dan sangat sederhana. Namun, kenyataan yang di dapat dari para ahli, ini memiliki perbedaan. Perbedaannya sebagai berikut:

- a. *AI - Artificial Intelligence*: Yang berarti komputer lebih baik dari manusia untuk melakukan tugas tertentu. Seperti robot yang bisa membuat keputusan berdasarkan banyaknya masukan, tidak seperti Terminator atau C3PO. Sebenarnya istilah yang sangat luas itu tidak terlalu berguna.
- b. *ML - Mesin belajar* adalah metode untuk mencapai AI. Ini berarti membuat prediksi tentang sesuatu berdasarkan pelatihan dari kumpulan data parsing. Ada banyak cara yang berbeda di platform ML yang dapat menerapkan perangkat pelatihan untuk memprediksi sesuatu.
- c. *NL - Jaringan syaraf tiruan* adalah salah satu cara model pembelajaran mesin untuk memprediksi sesuatu. Jaringan saraf bekerja sedikit seperti otak Anda, dengan menyesuaikan diri dan banyak berlatih untuk memahaminya. Anda akan menciptakan lapisan simpul yang sangat dalam.

#### 5. Memberikan struktur yang jelas terhadap AI

Sebagian besar model *Machine Learning* bergantung pada manusia untuk melakukan apa yang akan dikerjakan mesin pembelajaran. Inilah yang membuat anda selalu bergantung

dengan teknologi tersebut, karena sesuatu yang ingin anda kerjakan. Dan bahkan saat Anda memberikan instruksi yang jelas, biasanya itu masih saja salah. Anda harus begitu eksplisit dengan sistem ini sehingga kesempatan itu tiba-tiba menjadi lebih mudah.

Bahkan halaman web sederhana yang menunjukkan sebuah kotak dengan sebuah kata di dalamnya mengharuskan Anda untuk memberi tahu persis di mana kotak itu muncul, seperti apa bentuknya, warna apa itu, bagaimana cara bekerja pada peramban yang berbeda, bagaimana ditampilkan dengan benar pada perangkat yang berbeda.

Ada banyak cara menghalangi jaringan *syaraf* yang sangat dalam untuk mengambil alih dunia dan mengubah kita agar terlihat lebih kuat, terutama karena semua yang akan kita lakukan tidak segampang dan semudah yang kita pikirkan.[4]

### **1.2.6 Penerapan *Machine Learning***

Perkembangan teknologi semakin hari semakin cepat. *Machine Learning* merupakan salah satu bidang ilmu pengetahuan yang berperan besar di dalamnya. Secara tidak sadar, hampir semua orang menggunakan produk yang dihasilkan oleh penerapan *Machine Learning*. Berikut 10 manfaat dan contoh penerapan *Machine Learning* dalam kehidupan sehari-hari.

#### **1. Hasil Pencarian Search Engine**

Google dan mesin pencari lainnya seperti Bing dan Yandex sudah dari dulu menerapkan *Machine Learning* untuk melakukan peringkingan laman suatu website. Setiap mesin pencari mempunyai resep tersendiri pada algoritma pencarian yang

digunakan. Dalam praktiknya ketika kita mengetik kata kunci, Google akan menampilkan hasil pencarian yang paling mendekati kata kunci tersebut. Apabila kita memilih suatu halaman dan menghabiskan banyak waktu pada halaman tersebut, Google akan mendeteksi bahwa halaman tersebut sesuai dengan kata kunci yang kita masukkan.

## 2. Rekomendasi Produk *Marketplace*

Perkembangan teknologi menyebabkan penggunaan marketplace semakin diminati baik dari sisi penjual dan pembeli. Hampir semua orang menggunakan marketplace, misalnya *Tokopedia*, *Bukalapak*, dan *Shopee*. Tiap detik bisa dipastikan terjadi transaksi antar pedagang dan pembeli. Untuk meningkatkan pengalaman pengguna, setiap akun perlu menampilkan rekomendasi produk yang sesuai dengan minat pembelian pembeli. Untuk melakukan ini secara otomatis dan real time, tentunya *Machine Learning* sangat menentukan keakuratan rekomendasi produk tiap pembeli di akun nya.

## 3. Pengalaman Pengguna Sosial Media

Tidak bisa di pungkiri, walaupun tidak dapat mengalahkan mesin pencari sebagai situs terpopuler. Sosial media merupakan situs yang paling lama diakses oleh penggunanya berdasarkan data Alexa. Sosial media seperti *Facebook* dan *Twitter* menggunakan *Machine Learning* sehingga penggunanya betah untuk terus melihat layar gadget. Berikut beberapa peningkatan pengalaman pengguna yang merupakan aplikasi dari *Machine Learning*.

#### 4. Konten yang Ditampilkan Periklanan Digital

Google Adword adalah media periklanan digital terpopuler di dunia. Adword menampilkan iklan-iklan pada situs web yang menjadi publisher di Google Adsense. Iklan yang ditampilkan adalah iklan yang bersifat dinamis atau berubah-ubah. Google Adword mengumpulkan data situs berdasarkan topik, kemudian menampilkan iklan-iklan yang relevan dengan topik tersebut. Disamping itu Google Adword juga menggunakan cookies, sebagai referensi aktivitas pengunjung suatu website terkait situs-situs yang dikunjungi sebelumnya. Adword melakukan akumulasi terhadap 2 faktor tersebut, sehingga dapat ditampilkan iklan yang sesuai dengan pengunjung website.

#### 5. Asisten Pribadi Virtual

Gadget telah dilengkapi dengan asisten pribadi virtual, baik laptop maupun smartphone, misalnya: Cortana di Microsoft Windows, Siri di iPhone, dan Google Now di Android. Asisten virtual ini dapat membantu penggunanya untuk melakukan pencarian di internet, menanyakan jalan, cuaca, melakukan panggilan telepon, hingga membuka aplikasi. Di Windows, Cortana bahkan mempelajari penggunanya untuk memberikan rekomendasi perintah yang akan dilakukan.[7]

##### **1.2.7 Metode Algoritma *Machine Learning***

###### 1. Supervised *Machine Learning* algorithms

Supervised *Machine Learning* adalah algoritma *Machine Learning* yang dapat menerapkan informasi yang telah ada pada data dengan memberikan label tertentu, misalnya data yang telah

diklasifikasikan sebelumnya (terarah). Algoritma ini mampu memberikan target terhadap output yang dilakukan dengan membandingkan pengalaman belajar di masa lalu.

## 2. Unsupervised *Machine Learning* algorithms

Unsupervised *Machine Learning* adalah algoritma *Machine Learning* yang digunakan pada data yang tidak mempunyai informasi yang dapat diterapkan secara langsung (tidak terarah). Algoritma ini diharapkan mampu menemukan struktur tersembunyi pada data yang tidak berlabel.

## 3. Semi-supervised *Machine Learning* algorithms

Semi-supervised *Machine Learning* adalah algoritma yang digunakan untuk melakukan pemebelajaran data berlabel dan tanpa label. Sistem yang menggunakan metode ini dapat meningkatkan efesiensi output yang dihasilkan.

## 4. Reinforcement *Machine Learning* algorithms

Reinforcement *Machine Learning* adalah algoritma yang mempunyai kemampuan untuk bertinteraksi dengan proses belajar yang dilakukan, algoritma ini akan memberikan poin (*reward*) saat model yang diberikan semakin baik atau mengurangi poin (*error*) saat model yang dihasilkan semakin buruk. Salah satu penerapannya adalah pada mesin pencari.

### **1.2.8 Dampak *Machine Learning* di Masyarakat**

Penerapan teknologi *Machine Learning* mau tidak mau pasti telah dirasakan sekarang. Dampak-dampak yang dirasakan dapat berupa positif maupun negatif.

#### 1. Dampak Positif

Menjadi peluang bagi para wirausahawan dan praktisi teknologi untuk terus berkarya dalam mengembangkan teknologi *Machine Learning*. Terbentuknya aktivitas yang harus dilakukan manusia pun menjadi salah satu dampak positif *Machine Learning*. Memang tidak semua masalah bisa dipecahkan dengan program *Machine Learning*.

Beberapa contoh program *Machine Learning* yang telah digunakan dalam kehidupan sehari-hari:

- Pendekripsi Spam
- Pendekripsi Wajah
- Rekomendasi Produk
- Diagnosa Medis
- Pendekripsi Penipuan Kartu Kredit
- Pengenal Digit
- Perdagangan Saham
- Segmentasi Pelanggan
- Mobil yang bisa Mengendarai Sendiri
- Sistem Ejaan Ms. Word

## 2. Dampak Negatif

Selain sangat membantu, *Machine Learning* juga menimbulkan dampak negatif, seperti adanya pemotongan tenaga kerja karena pekerjaan telah digantikan oleh alat teknologi *Machine Learning*. Ditambah dengan ketergantungan terhadap teknologi akan semakin terasa. Manusia akan lebih terlena oleh kemampuan gadget-nya sehingga lupa belajar untuk melakukan suatu aktivitas tanpa bantuan teknologi.[8]

## **1.3 Neural Network**

### **1.3.1 Definisi Neural Network**

Jaringan syaraf tiruan (JST) atau yang biasa disebut *Artificial Neural Network* (ANN) atau *Neural Network* (NN) saja, merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup. *Neural Network* berupa suatu model sederhana dari suatu syaraf nyata dalam otak manusia seperti suatu unit threshold yang biner.

*Neural Network* merupakan sebuah mesin pembelajaran yang dibangun dari sejumlah elemen pemrosesan sederhana yang disebut neuron atau node. Setiap neuron dihubungkan dengan neuron yang lain dengan hubungan komunikasi langsung melalui pola hubungan yang disebut arsitektur jaringan. Bobot-bobot pada koneksi mewakili besarnya informasi yang digunakan jaringan. Metode yang digunakan untuk menentukan bobot koneksi tersebut dinamakan dengan algoritma pembelajaran. Setiap neuron mempunyai tingkat aktivasi yang merupakan fungsi dari *input* yang masuk padanya. Aktivasi yang dikirim suatu neuron ke neuron lain berupa sinyal dan hanya dapat mengirim sekali dalam satu waktu, meskipun sinyal tersebut disebarluaskan pada beberapa neuron yang lain.[9]

Cabang ilmu kecerdasan buatan cukup luas, dan erat kaitannya dengan disiplin ilmu yang lainnya. Hal ini bisa dilihat dari berbagai aplikasi yang merupakan hasil kombinasi dari berbagai ilmu. Seperti halnya yang ada pada peralatan medis yang berbentuk aplikasi. Sudah berkembang bahwa aplikasi yang dibuat

merupakan hasil perpaduan dari ilmu kecerdasan buatan dan juga ilmu kedokteran atau lebih khusus lagi yaitu ilmu biologi.

*Neural Network* merupakan kategori ilmu Soft Computing. *Neural Network* sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan output. Output diperoleh dari variasi stimulasi dan proses yang terjadi di dalam otak manusia. Kemampuan manusia dalam memproses informasi merupakan hasil kompleksitas proses di dalam otak. Misalnya, yang terjadi pada anak-anak, mereka mampu belajar untuk melakukan pengenalan meskipun mereka tidak mengetahui algoritma apa yang digunakan. Kekuatan komputasi yang luar biasa dari otak manusia ini merupakan sebuah keunggulan di dalam kajian ilmu pengetahuan.

Fungsi dari *Neural Network* diantaranya adalah:

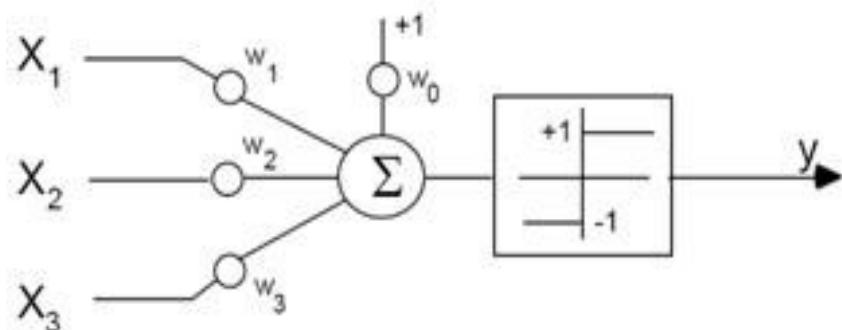
1. Pengklasifikasian pola
2. Memetakan pola yang didapat dari *input* ke dalam pola baru pada *output*
3. Penyimpan pola yang akan dipanggil kembali
4. Memetakan pola-pola yang sejenis
5. Pengoptimasi permasalahan
6. Prediksi [10]

### **1.3.2 Sejarah *Neural Network***

Perkembangan ilmu *Neural Network* sudah ada sejak tahun 1943 ketika Warren McCulloch dan Walter Pitts memperkenalkan perhitungan model *Neural Network* yang pertama kalinya. Mereka

melakukan kombinasi beberapa *processing unit* sederhana bersama-sama yang mampu memberikan peningkatan secara keseluruhan pada kekuatan komputasi.

Hal ini dilanjutkan pada penelitian yang dikerjakan oleh Rosenblatt pada tahun 1950, dimana dia berhasil menemukan sebuah *two-layer network*, yang disebut sebagai *perceptron*. Perceptron memungkinkan untuk pekerjaan klasifikasi pembelajaran tertentu dengan penambahan bobot pada setiap koneksi antar-*network*.



Gambar 1. 1 Perceptron

Pada gambar 1.1 di atas merupakan *perception* pada *neural network*. *Perception* sendiri merupakan sepotong bagian dari jaringan NN yang terdiri dari *input*, *synapses*, *neuron* dan *output*, perhitungan dilakukan disetiap perceptron ini, hasilnya tergantung dari jumlah *input* dan outputnya serta menghitung hasil akhir tergantung dari fungsi aktivasi yang digunakan. Keberhasilan perceptron dalam pengklasifikasian pola tertentu ini tidak sepenuhnya sempurna, masih ditemukan juga beberapa keterbatasan didalamnya. Perceptron tidak mampu untuk

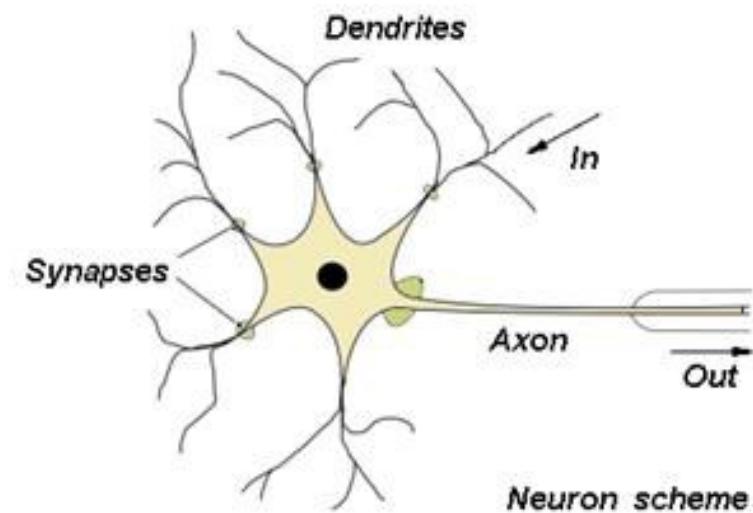
menyelesaikan permasalahan XOR (*exclusive-OR*). Penilaian terhadap keterbatasan *Neural Network* ini membuat penelitian di bidang ini sempat mati selama kurang lebih 15 tahun. Namun demikian, perceptron berhasil menjadi sebuah dasar untuk penelitian-penelitian selanjutnya di bidang *Neural Network*. Pengkajian terhadap *Neural Network* mulai berkembang lagi selanjutnya di awal tahun 1980-an. Para peneliti banyak menemukan bidang interest baru pada domain ilmu *Neural Network*. Penelitian terakhir diantaranya adalah mesin Boltzmann, jaringan Hopfield, model pembelajaran kompetitif, *multilayer network*, dan teori model resonansi adaptif.

Untuk saat ini, *Neural Network* sudah dapat diterapkan pada beberapa task, diantaranya classification, recognition, approximation, prediction, clusterization, memory simulation dan banyak task-task berbeda yang lainnya, dimana jumlahnya semakin bertambah seiring berjalannya waktu.[11]

### **1.1.3 Konsep *Neural Network***

#### **1. Proses Kerja Jaringan Syaraf Pada Otak Manusia**

Ide dasar *Neural Network* dimulai dari otak manusia, dimana otak memuat sekitar  $10^{11}$  neuron. Neuron ini berfungsi memproses setiap informasi yang masuk. Satu neuron memiliki 1 *akson*, dan minimal 1 *dendrite*. Setiap sel syaraf terhubung dengan syaraf lain, jumlahnya mencapai sekitar  $10^4$  *synapsis*. Masing-masing sel itu saling berinteraksi satu sama lain yang menghasilkan kemampuan tertentu pada kerja otak manusia.



Gambar 1. 2 Struktur Neuron pada otak manusia

Pada gambar 1.2 di atas merupakan struktur *neuron* pada otak manusia, bisa dilihat ada beberapa bagian dari otak manusia, yaitu:

1. Dendrit (*Dendrites*) berfungsi untuk mengirimkan impuls yang diterima ke badan sel syaraf.
2. Akson (*Axon*) berfungsi untuk mengirimkan impuls dari badan sel ke jaringan lain
3. Sinapsis berfungsi sebagai unit fungsional di antara dua sel syaraf.

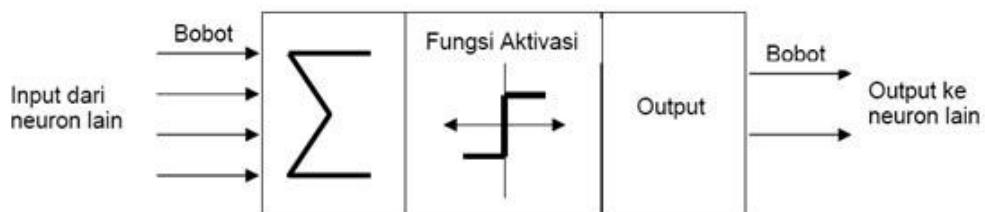
Proses yang terjadi pada otak manusia adalah:

Sebuah neuron menerima impuls dari neuron lain melalui dendrit dan mengirimkan sinyal yang dihasilkan oleh badan sel melalui akson. Akson dari sel syaraf ini bercabang-cabang dan berhubungan dengan dendrit dari sel syaraf lain dengan cara mengirimkan impuls melalui sinapsis. Sinapsis adalah unit

fungsional antara 2 buah sel syaraf, misal A dan B, dimana yang satu adalah serabut akson dari neuron A dan satunya lagi adalah dendrit dari neuron B. Kekuatan sinapsis bisa menurun/meningkat tergantung seberapa besar tingkat propagasi (penyiaran) sinyal yang diterimanya. Impuls-impuls sinyal (informasi) akan diterima oleh neuron lain jika memenuhi batasan tertentu, yang sering disebut dengan nilai ambang (threshold).

## 2. Struktur Neural Network

Dari struktur neuron pada otak manusia, dan proses kerja yang dijelaskan di atas, maka konsep dasar pembangunan *Neural Network* buatan (*Artificial Neural Network*) terbentuk. Ide mendasar dari *Artificial Neural Network* (ANN) adalah mengadopsi mekanisme berpikir sebuah sistem atau aplikasi yang menyerupai otak manusia, baik untuk pemrosesan berbagai sinyal elemen yang diterima, toleransi terhadap kesalahan/error, dan juga *parallel processing*.



Gambar 1. 3 Struktur ANN

Pada gambar 1.3 merupakan struktur dari ANN dimana karakteristik dari ANN dilihat dari pola hubungan antar neuron, metode penentuan bobot dari tiap koneksi, dan fungsi aktivasinya. Gambar di atas menjelaskan struktur ANN secara mendasar, yang dalam kenyataannya tidak hanya sederhana seperti itu.

1. *Input* , berfungsi seperti *dendrite*
2. *Output*, berfungsi seperti akson
3. Fungsi aktivasi, berfungsi seperti sinapsis

*Neural Network* dibangun dari banyak node/unit yang dihubungkan oleh link secara langsung. Link dari unit yang satu ke unit yang lainnya digunakan untuk melakukan propagasi aktivasi dari unit pertama ke unit selanjutnya. Setiap link memiliki bobot numerik. Bobot ini menentukan kekuatan serta penanda dari sebuah koneksi.

Proses pada ANN dimulai dari *input* yang diterima oleh neuron beserta dengan nilai bobot dari tiap-tiap *input* yang ada. Setelah masuk ke dalam neuron, nilai *input* yang ada akan dijumlahkan oleh suatu fungsi perambatan (*summing function*), yang bisa dilihat seperti pada di gambar dengan lambang sigma ( $\Sigma$ ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap neuron, disini akan dibandingkan hasil penjumlahan dengan *threshold* (nilai ambang) tertentu. Jika nilai melebihi *threshold*, maka aktivasi neuron akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, neuron akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Proses ini akan terus berulang pada *input -input* selanjutnya.

ANN terdiri dari banyak neuron di dalamnya. Neuron-neuron ini akan dikelompokkan ke dalam beberapa *layer*. Neuron yang terdapat pada tiap *layer* dihubungkan dengan neuron pada

*layer* lainnya. Hal ini tentunya tidak berlaku pada *layer input* dan output, tapi hanya *layer* yang berada di antaranya. Informasi yang diterima di *layer input* dilanjutkan ke *layer-layer* dalam ANN secara satu persatu hingga mencapai *layer* terakhir/*layer* output. *Layer* yang terletak di antara *input* dan output disebut sebagai *hidden layer*. Namun, tidak semua ANN memiliki *hidden layer*, ada juga yang hanya terdapat *layer input* dan output saja.[10]

#### **1.1.4 Algoritma Dalam Neural Network**

##### **1. Algoritma *Backpropagation***

Salah satu algoritma pelatihan jaringan syaraf tiruan yang banyak dimanfaatkan dalam bidang pengenalan pola adalah *Backpropagation*. Algoritma ini umumnya digunakan pada jaringan syaraf tiruan yang berjenis multi-layer *feed-forward*, yang tersusun dari beberapa lapisan dan sinyal dialirkan secara searah dari *input* menuju output. Algoritma pelatihan *Backpropagation* pada dasarnya terdiri dari tiga tahapan [Fausett, 1994], yaitu:

1. *Input* nilai data pelatihan sehingga diperoleh nilai output.
2. Propagasi balik dari nilai *error* yang diperoleh.
3. Penyesuaian bobot koneksi untuk meminimalkan nilai *error*.

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai *error* yang diinginkan. Setelah training selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan syaraf tiruan tersebut. Secara matematis [Rumelhart, 1986], ide dasar dari algoritma *Backpropagation* ini

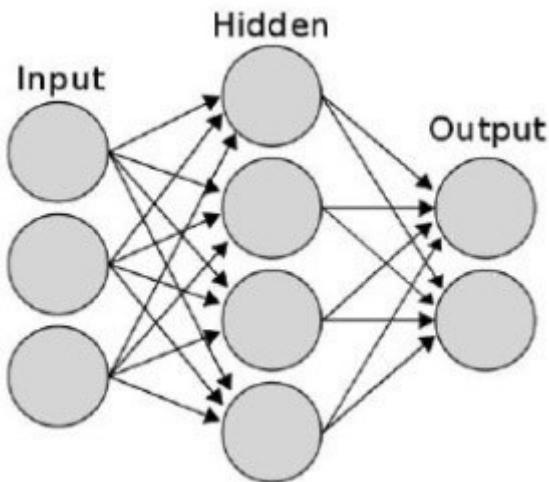
sesungguhnya adalah penerapan dari aturan rantai (chain rule) untuk menghitung pengaruh masing-masing bobot terhadap fungsi *error*.

## 2. Algoritma *Quickprop*

Pada algoritma *Quick prop* dilakukan pendekatan dengan asumsi bahwa kurva fungsi *error* terhadap masing-masing bobot penghubung berbentuk parabola yang terbuka ke atas, dan gradien dari kurva *error* untuk suatu bobot tidak terpengaruh oleh bobot-bobot yang lain [Fahlman, 1988]. Dengan demikian perhitungan perubahan bobot hanya menggunakan informasi lokal pada masing-masing bobot. Perubahan bobot pada algoritma Quickprop dirumuskan sebagai berikut: Pada eksperimen dengan masalah XOR dan *encoder/decoder* [Fahlman, 1988], terbukti bahwa algoritma Quick prop dapat meningkatkan kecepatan training. Eksperimen dari [Schiffmann, 1993] juga menunjukkan peningkatan kecepatan training dan unjuk kerja yang signifikan.

### 1.1.5 Arsitektur Neural Network

Secara umum, Arsitektur JST terdiri atas beberapa lapisan, yaitu lapisan masukan (*Input Layer*), lapisan tersembunyi (*hidden layer*), dan lapisan keluaran (*output layer*). Masing-masing lapisan mempunyai jumlah node atau neuron yang berbeda-beda. Arsitektur JST tersebut dapat diilustrasikan sebagai gambar berikut ini:



*Gambar 1. 4 Arsitektur Neural Network*

Pada gambar 1.4 di atas merupakan arsitektur dari *Neural Network* adalah gambaran susunan komponen *layer* dan *neuron* pada *input* , *hidden* dan *output* yang terhubung dengan bobot, fungsi aktivasi dan fungsi pembelajaran (*learning function*). Berikut adalah penjelasan dari masing-masing lapisan pada neural network:

#### 1. Lapisan Masukan (*Input Layer*)

Lapisan masukan merupakan lapisan yang terdiri dari beberapa neuronyang akan menerima sinyal dari luar dan kemudian meneruskan ke neuron-neuron lain dalam jaringan. Lapisan ini dillhami berdasarkan cirri-ciri dancara kerja sel-sel saraf sensori pada jaringan saraf biologi.

#### 2. Lapisan tersembunyi (*hidden layer*)

Lapisan tersembunyi merupakan tiruan dari sel-sel syaraf konektor pada jaringan saraf bilogis. Lapisan tersembunyi berfungsi meningkatkan kemampuan jaringan dalam memecahkan

masalah. Konsekuensi dari adanya lapisan ini adalah pelatihan menjadi makin sulit atau lama.

### 3. Lapisan keluaran (*output layer*)

Lapisan keluaran berfungsi menyalurkan sinyal-sinyal keluaran hasil pemrosesan jaringan. Lapisan ini juga terdiri dari sejumlah *neuron*. Lapisan keluaran merupakan tiruan dari sel saraf motor pada jaringan saraf biologis.[12]

## **1.4 Deep Learning**

### **1.1.4. Definisi Deep Learning**

Apa itu *Deep Learning*? Oke kita kenalan dulu ya! Karena ada pepatah yang mengatakan seperti ini, “Tak kenal maka sayang”. *Deep Learning* (Pembelajaran Dalam) atau sering dikenal dengan istilah Pembelajaran Struktural Mendalam (Deep Structured Learning) atau Pembelajaran Hierarki (Hierarchical learning) adalah salah satu cabang dari ilmu pembelajaran mesin (*Machine Learning*) yang terdiri algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linear yang ditata berlapis-lapis dan mendalam. Teknik dan algoritma dalam Pembelajaran dalam dapat digunakan baik untuk kebutuhan pembelajaran terarah (*Supervised Learning*), pembelajaran tak terarah (*UnSupervised Learning*) dan semi-terarah (*semi-Supervised Learning*) dalam berbagai aplikasi seperti pengenalan citra, pengenalan suara, klasifikasi teks, dan sebagainya[4]. *Deep Learning* disebut sebagai Deep (dalam) karena struktur dan jumlah jaringan saraf pada algoritme nya sangat banyak bisa mencapai hingga ratusan lapisan.

*Deep Learning* adalah salah satu jenis algoritma jaringan saraf tiruan yang menggunakan metadata sebagai *input* dan mengolahnya menggunakan sejumlah lapisan tersembunyi (*hidden layer*) transformasi non linier dari data masukan untuk menghitung nilai *output*. Algortima pada *Deep Learning* memiliki fitur yang unik yaitu sebuah fitur yang mampu mengekstraksi secara otomatis. Hal ini berarti algoritma yang dimilikinya secara otomatis dapat menangkap fitur yang relevan sebagai keperluan dalam pemecahan suatu masalah. Algortima semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan, algortima ini dapat digunakan untuk memecahkan permasalahan yang perlu pengawasan (*supervised*), tanpa pengawasan (*unsupervised*), dan semi terawasi (*semi supervised*).

Dalam jaringan saraf tiruan tipe *Deep Learning* setiap lapisan tersembunyi bertanggung jawab untuk melatih serangkaian fitur unik berdasarkan output dari jaringan sebelumnya. Algortima ini akan menjadi semakin kompleks dan bersifat abstrak ketika jumlah lapisan tersembunyi (*hidden layer*) semakin bertambah banyak. Jaringan saraf yang dimiliki oleh *Deep Learning* terbentuk dari hierarki sederhana dengan beberapa lapisan hingga tingkat tinggi atau banyak lapisan (*multi-layer*). Berdasarkan hal itulah *Deep Learning* dapat digunakan untuk memecahkan masalah kompleks yang lebih rumit dan terdiri dari sejumlah besar lapisan transformasi non linier.[13]

### **1.4.2 Sejarah *Deep Learning***

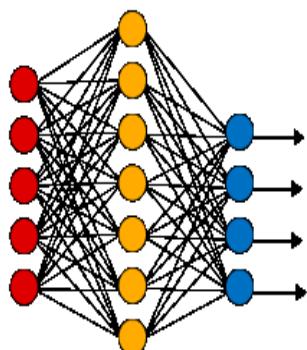
Pada tahun 2006, Geoffrey Hinton memperkenalkan salah satu varian jaringan saraf tiruan yang disebut deep belief nets, ide untuk men-train model jaringan saraf tiruan ini adalah dengan men-train dua *layer* kemudian tambahkan satu *layer* di atasnya, kemudian train hanya *layer* teratas dan begitu seterusnya. Dengan strategi ini kita dapat men-train model jaringan saraf tiruan dengan *layer* lebih banyak dari model-model sebelumnya. Paper ini merupakan awal populernya istilah *Deep Learning* untuk membedakan arsitektur jaringan saraf tiruan dengan banyak *layer*.

Setelah istilah *Deep Learning* populer, *Deep Learning* belum menjadi daya tarik yang besar bagi para peneliti karena jaringan saraf tiruan dengan banyak *layer* memiliki kompleksitas algoritma yang besar, sehingga membutuhkan komputer dengan spesifikasi tinggi, dan tidak efisien secara komputasi saat itu. Hingga pada tahun 2009 Andrew Ng dkk memperkenalkan penggunaan GPU untuk *Deep Learning* melalui paper yang berjudul Large-scale Deep *UnSupervised Learning* using Graphics Processors. Dengan menggunakan GPU jaringan saraf tiruan dapat berjalan lebih cepat dibanding dengan menggunakan CPU. Dengan tersedianya hardware yang memadai perkembangan *Deep Learning* mulai pesat, dan menghasilkan produk-produk yang dapat kita nikmati saat ini seperti pengenal wajah, self-driving car, pengenal suara, dan lain lain.[14]

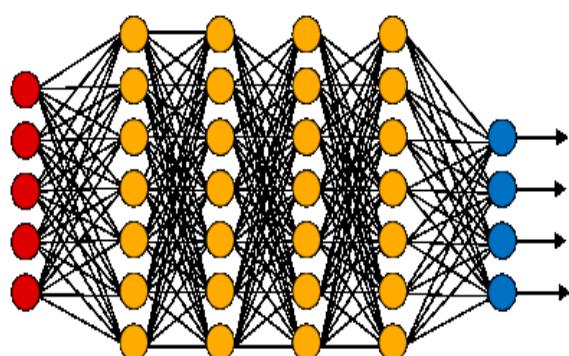
### 1.4.3 Perbedaan Neural Network Dengan Deep Learning

Jaringan Saraf Tiruan (*Neural Network*) adalah jaringan saraf yang biasanya menggunakan jaringan seperti umpan maju (feed forward) atau recurrent network yang hanya memiliki 1 atau 2 lapisan tersembunyi. Tetapi, jika lapisan jaringan sarafnya lebih dari 2 *layer* ke atas atau bahkan mencapai ratusan lapisan itulah yang disebut sebagai *Deep Learning*. Pada Jaringan *Syaraf* Tiruan arsitektur jaringan yang dimilikinya kurang kompleks dan membutuhkan lebih banyak informasi tentang data *input* sehingga dapat menentukan algortima mana yang dapat digunakan. Dalam Jaringan Saraf Tiruan terdiri dari beberapa algoritma yaitu Model Hebb, Perceptron, Adeline, *Propagation* Maju, dll. Sedangkan pada algortima jaringan saraf *Deep Learning* tidak memerlukan informasi apapun terhadap data yang akan dipelajarinya, dan algoritme nya dapat secara mandiri melakukan tuning (penyetelan) dan pemilihan model yang paling optimal.

**Simple Neural Network**



**Deep Learning Neural Network**



● Input Layer      ● Hidden Layer      ● Output Layer

Gambar 1.5 Perbedaan antara lapisan layer pada Jaringan Saraf Tiruan(*Neural Network*) dengan Jaringan Deep Learning

#### **1.4.4 Jenis-Jenis Algorithma Deep Learning**

Berikut ini adalah beberapa jenis algoritma *Deep Learning* berdasarkan kegunaannya.

1. Pembelajaran dengan menggunakan label (*Supervised Learning*)

*Tabel 1. 1Pembelajaran dengan menggunakan label (Supervised Learning)*

Pengolahan data berupa tulisan - analisa <i>sentiment</i> - penguraian kalimat	<i>Recurrent Network,</i> <i>Recurrent Network Tensor Network</i>
Pengolahan citra digital	<i>Deep Belief Network,</i> <i>Convolutional Network</i>
Pengenalan obyek	<i>Convolutional Network,</i> <i>Recurrent Network Tensor Network</i>
Pengolahan suara	<i>Recurrent Network</i>

2. Pembelajaran tanpa menggunakan label (*UnSupervised Learning*)

Bila ingin mencoba mengenali pola atau mengekstrak data, gunakan *Restricted Boltzman Machines* atau *Autoencoder*.

3. Berdasarkan obyektif

- Bila anda ingin melakukan klasifikasi data, gunakan algoritma *Multi Layer Perceptron* dengan *Rectified Linear Unit*, atau *Deep Belief Network*
- Jika anda ingin melakukan *Deep Learning* dengan data yang dapat berubah terhadap waktu, gunakan *Recurrent Neural Network*. [15]

## BAB 2

---

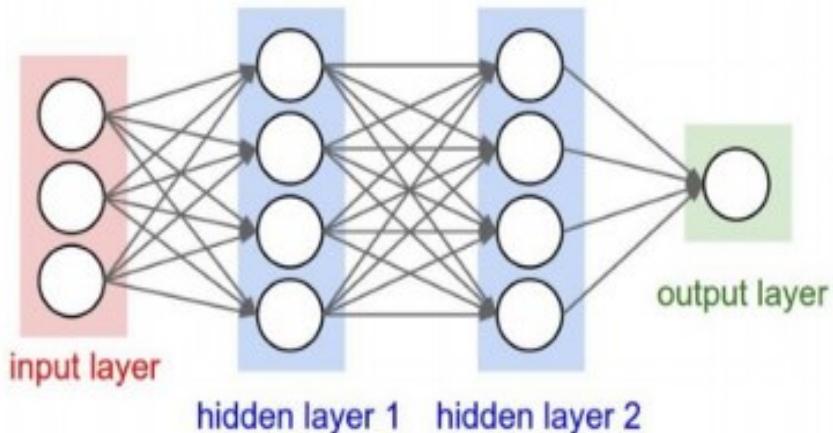
# PENGENALAN CNN, FAST R-CNN, FASTER R-CNN, CROSS VALIDATION, DAN CONFUSION MATRIX

---

### **2.1 Convolutional Neural Network (CNN)**

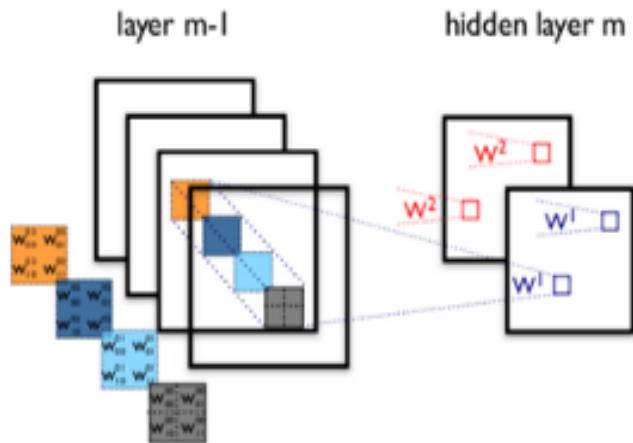
#### **2.1.1. Definisi Convolutional Neural Network**

*Convolutional Neural Network* adalah salah satu metode *Machine Learning* dari pengembangan *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. CNN memiliki dua metode; yakni klasifikasi menggunakan feedforward dan tahap pembelajaran menggunakan *Backpropagation*. Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi.



*Gambar 2. 1 Arsitektur MLP Sederhana*

Gambar 2.1 di atas merupakan arsitektur dari *Multi Layer Perceptron* (MLP). Sebuah MLP memiliki *Input Layer* (kotak merah dan biru) dengan masing-masing *layer* berisi neuron (lingkaran putih). MLP menerima *input* data satu dimensi dan mempropagasi data tersebut pada jaringan hingga menghasilkan output. Setiap hubungan antar neuron pada dua *layer* yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data *input* pada *layer* dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi. Data yang dipropagasi pada CNN adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Operasi linear pada CNN menggunakan operasi konvolusi, dengan bobot yang tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan *kernel* konvolusi. Dimensi bobot pada CNN adalah:

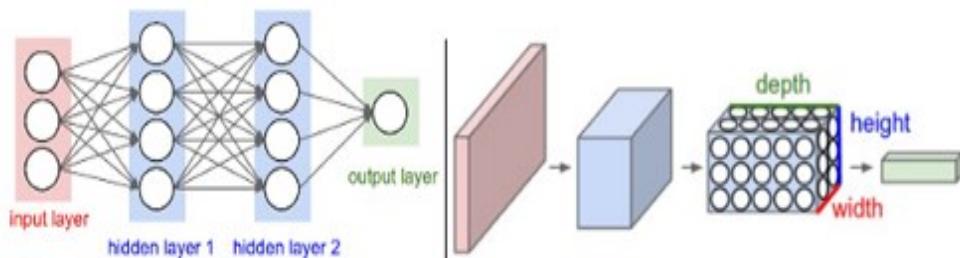


*Gambar 2. 2 Proses Konvolusi pada CNN*

Gambar 2.2 diatas merupakan proses konvolusi pada CNN. CNN merupakan pengembangan lebih lanjut dari MLP karena menggunakan metode yang mirip dengan dimensi yang lebih banyak. Di algoritma CNN ini, *input* dari *layer* sebelumnya bukan *array* 1 dimensi melainkan *array* 2 dimensi. Jika di analogikan dengan fitur dari wajah manusia, *layer* pertama merupakan refleksi goresan-goresan berbeda arah, pada *layer* kedua fitur seperti bentuk mata, hidung, dan mulut mulai terlihat, hal ini karena di lakukan pooling/penggabungan dari *layer* pertama yang masih berupa goresan-goresan, pada *layer* ketiga akan terbentuk kombinasi fitur-fitur mata, hidung, dan mulut yang nantinya akan disimpulkan dengan wajah orang tertentu.

Keseluruhan skala dalam objek sangat penting agar *input* tidak kehilangan informasi spasialnya yang akan diekstraksi fitur dan diklasifikasikan. Hal ini akan menambah tingkat akurasi dan optimum algoritma CNN. Seperti pada kubus yang memiliki skala pada panjang, lebar, dan tinggi. Jika hanya menggunakan *Neural*

*Network* biasa, mungkin hanya memuat skala panjang dan tinggi. Namun CNN bisa memuat semua informasi dari keseluruhan skala yang bisa mengklasifikasikan objek dengan lebih akurat karena bisa menggunakan skala lebarnya juga (yang mungkin tidak akan terlihat oleh *Neural Network* lainnya yang berdimensi dua).



*Gambar 2. 3 Perbedaan arsitektur antara Neural Network pada umumnya dengan CNN*

Gambar 2.3 merupakan perbedaan pada arsitektur antara *neural network* dengan CNN. CNN terdiri dari berbagai lapisan yang dimana setiap lapisan memiliki *Application Program Interface* (API) alias antarmuka program aplikasi sederhana. CNN dengan *input* awal balok tiga dimensi akan ditransformasikan menjadi *output* tiga dimensi dengan beberapa fungsi diferensiasi yang memiliki atau tidak memiliki parameter. CNN membentuk neuron-neuronnya ke dalam tiga dimensi (panjang, lebar, dan tinggi) dalam sebuah lapisan.[16]

### **2.1.2 Sejarah Convolutional Neural Network**

*Convolutional Neural Network* (CNN) pertama kali diperkenalkan oleh Yann LeCun pada tahun 1988. CNN merupakan salah satu metode yang mengawali kemunculan dan kesuksesan *Deep Learning*.

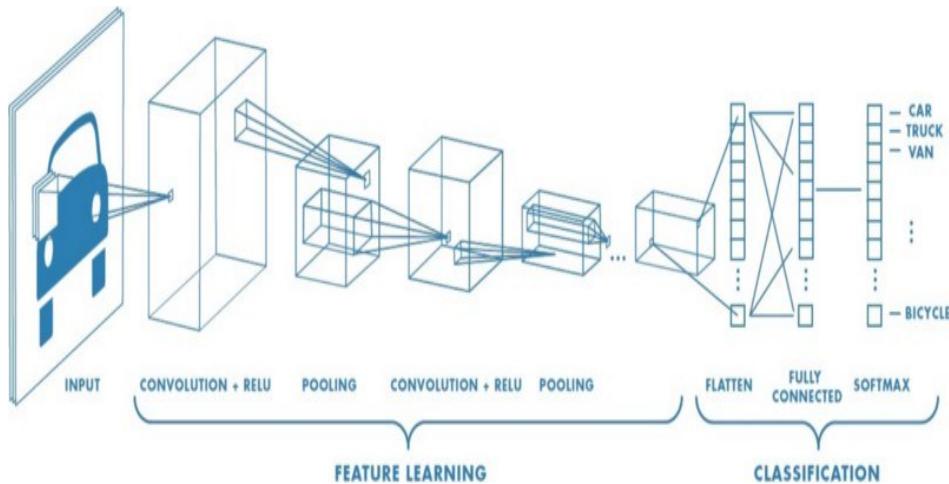
Pada tahun 1950-an Hubel dan Wisel melakukan eksperimen pada salah satu bagian otak kucing yaitu visual cortex. mereka menemukan bahwa visual cortex memiliki bagian kecil berupa sel-sel yang sensitif terhadap area tertentu pada pandangan mata. Hubel dan Wisel menemukan 2 tipe visual cortex, yaitu simple cell dan complex cell. Berdasarkan hasil pengamatan ini, pada tahun 1980-an Kunihiko Fukushima merancang Neocognitron yang merupakan model *Hierarchical Multilayered Neural Network*. Model tersebut telah digunakan pada beberapa kasus seperti klasifikasi karakter dari tulisan tangan (Handwritten Character Recognition). model inilah yang menjadi inspirasi dari *Convolutional Neural Network*.

CNN memiliki kesamaan struktur dengan artificial *Neural Network*. Pada kasus klasifikasi citra, CNN menerima citra *input* atau masukan kemudian diproses dan diklasifikasi ke kategori tertentu (mis. pesawat, kapal, burung, kucing, sapi).

Yang membedakan CNN dengan ANN adalah CNN memiliki arsitektur tambahan yang dioptimisasi untuk fitur yang ada pada citra *input* . komponen-komponen utama yang ada di dalam *Convolutional Neural Networks* adalah:

1. *Input Layer*
2. *Convolution Layer*
3. *Activation Function*
4. *Pooling Layer*
5. *Fully Connected Layer*

Pada gambar 2.4 di bawah ini bisa dilihat alur dari proses CNN dalam mengolah citra masukan sampai mengklasifikasikan citra tersebut ke kategori tertentu berdasarkan nilai keluarannya.



Gambar 2. 4 Arsitektur CNN

CNN terdiri dari beberapa *layer* dan dirancang untuk pengenalan citra kompleks yang efektif. CNN mempunyai performansi yang baik dalam ekstraksi fitur untuk mencapai confidence rate tinggi layaknya cara kerja otak manusia. Dalam proses identifikasi dan klasifikasi terdapat banyak *layer* yang digunakan untuk mendapatkan hasil identifikasi yang akurat. Kumpulan *layer* tersebut membentuk sebuah arsitektur yang kemudian digunakan untuk mengenali sebuah objek.

Arsitektur dari *Convolutional Neural Network* dibagi menjadi 2 bagian yaitu Feature Extraction Layer (Feature Learning) dan Fully Connected Layer (MLP) atau Classification. Dikatakan Feature Extraction Layer, karena proses yang terjadi pada bagian ini adalah melakukan encoding dari sebuah gambar menjadi features yang

berupa angka-angka yang merepresentasikan gambar tersebut (Feature Extraction).

Berikut penjelasan pada *layer* yang terdapat pada arsitektur Convolution Neural Network:

#### **2.1.2.1 Input Layer**

*Input Layer* menampung nilai piksel dari citra yang menjadi masukan. Untuk citra dengan ukuran  $64 \times 64$  dengan 3 channel warna, RGB(*Red, Green, Blue*) maka yang menjadi masukan akan adalah piksel *array* yang berukuran  $64 \times 64 \times 3$ . Pada gambar 2.5 dibawah ini merupakan ilustrasi *input* RGB pada sebuah gambar.

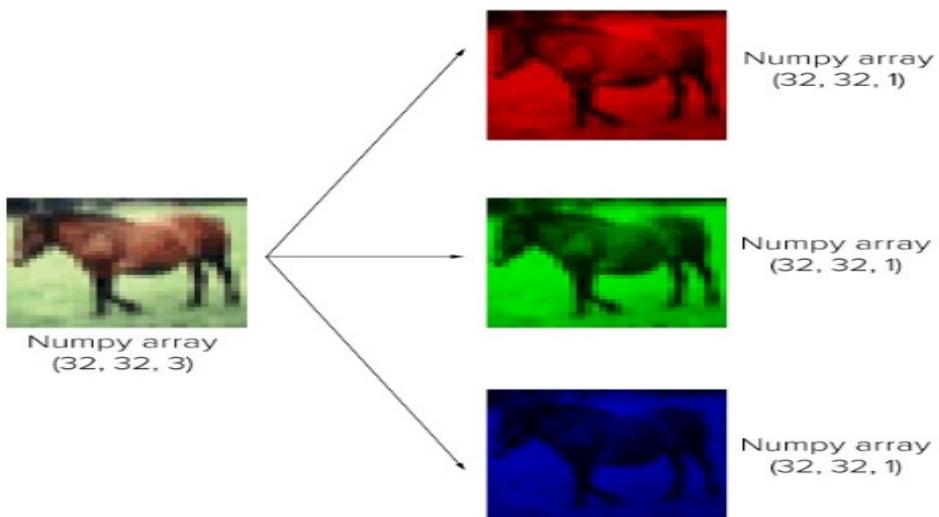


*Gambar 2. 5 Ilustrasi Input RGB*

#### **2.1.2.2 Convolution Layer**

*Convolution Layer* merupakan inti dari CNN. *Convolution Layer* menciptakan citra baru yang menampakkan fitur dari citra *input*. Pada proses tersebut, *Convolution Layer* memerlukan *filter* pada setiap citra yang menjadi masukkan. *Filter* pada *layer* ini yaitu *array* 2 dimensi dapat berukuran  $5 \times 5$ ,  $3 \times 3$  atau  $1 \times 1$ . Proses convolution dengan memakai *filter* pada *layer* ini akan

menghasilkan feature map yang akan digunakan pada *activation layer*.



Gambar 2. 6 Image RGB Convolutional Layer

Gambar 2.6 diatas merupakan alur pada *Convolution Layer RGB (Red, Green, Blue)* dari sebuah gambar yang memiliki ukuran  $32 \times 32$  pixels yang sebenarnya adalah multidimensional array dengan ukuran  $32 \times 32 \times 3$  (3 adalah jumlah channel). *Convolutional layer* terdiri dari beberapa *neuron* yang tersusun sedemikian rupa sehingga akan membentuk sebuah *filter* dengan panjang dan tinggi (*pixels*). Sebagai contoh, *layer* pertama pada *Feature Extraction layer* biasanya adalah *Convolutional layer* dengan ukuran  $5 \times 5 \times 3$ . Panjang 5 pixels, tinggi 5 pixels dan tebal/jumlah 3 buah sesuai dengan *channel* dari gambar tersebut.

Ketiga *filter* ini akan digeser keseluruhan bagian dari gambar. setiap pergeseran yang dilakukan oleh operasi “dot” antara *input* dan nilai dari *filter* tersebut akan menghasilkan sebuah *output* atau yang biasa disebut dengan *Activation map* atau *Feature Map*.

Input	Kernel	Intermediate Output																																																					
<table border="1"> <tr><td>1</td><td>0</td><td>1</td><td>0</td><td>2</td></tr> <tr><td>1</td><td>1</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>3</td><td>2</td><td>1</td><td>3</td></tr> <tr><td>0</td><td>2</td><td>0</td><td>1</td><td>0</td></tr> </table>	1	0	1	0	2	1	1	3	2	1	1	1	0	1	1	2	3	2	1	3	0	2	0	1	0	<table border="1"> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>2</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> </table>	0	1	0	0	0	2	0	1	0	<table border="1"> <tr><td>7</td><td>5</td><td>3</td></tr> <tr><td>4</td><td>7</td><td>5</td></tr> <tr><td>7</td><td>2</td><td>8</td></tr> </table>	7	5	3	4	7	5	7	2	8										
1	0	1	0	2																																																			
1	1	3	2	1																																																			
1	1	0	1	1																																																			
2	3	2	1	3																																																			
0	2	0	1	0																																																			
0	1	0																																																					
0	0	2																																																					
0	1	0																																																					
7	5	3																																																					
4	7	5																																																					
7	2	8																																																					
<table border="1"> <tr><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>0</td><td>1</td><td>2</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>3</td><td>0</td></tr> <tr><td>0</td><td>3</td><td>0</td><td>3</td><td>2</td></tr> <tr><td>1</td><td>0</td><td>3</td><td>2</td><td>1</td></tr> </table>	1	0	0	1	0	2	0	1	2	0	3	1	1	3	0	0	3	0	3	2	1	0	3	2	1	<table border="1"> <tr><td>2</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>3</td><td>0</td></tr> </table>	2	1	0	0	0	0	0	3	0	<table border="1"> <tr><td>5</td><td>3</td><td>10</td></tr> <tr><td>13</td><td>1</td><td>13</td></tr> <tr><td>7</td><td>12</td><td>11</td></tr> </table>	5	3	10	13	1	13	7	12	11	<table border="1"> <tr><td>19</td><td>13</td><td>15</td></tr> <tr><td>28</td><td>16</td><td>20</td></tr> <tr><td>23</td><td>18</td><td>25</td></tr> </table>	19	13	15	28	16	20	23	18	25
1	0	0	1	0																																																			
2	0	1	2	0																																																			
3	1	1	3	0																																																			
0	3	0	3	2																																																			
1	0	3	2	1																																																			
2	1	0																																																					
0	0	0																																																					
0	3	0																																																					
5	3	10																																																					
13	1	13																																																					
7	12	11																																																					
19	13	15																																																					
28	16	20																																																					
23	18	25																																																					
<table border="1"> <tr><td>2</td><td>0</td><td>1</td><td>2</td><td>1</td></tr> <tr><td>3</td><td>3</td><td>1</td><td>3</td><td>2</td></tr> <tr><td>2</td><td>1</td><td>1</td><td>1</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>3</td><td>2</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>2</td><td>1</td><td>1</td></tr> </table>	2	0	1	2	1	3	3	1	3	2	2	1	1	1	0	3	1	3	2	0	1	1	2	1	1	<table border="1"> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>2</td></tr> </table>	1	0	0	1	0	0	0	0	2	<table border="1"> <tr><td>7</td><td>5</td><td>2</td></tr> <tr><td>11</td><td>8</td><td>2</td></tr> <tr><td>9</td><td>4</td><td>6</td></tr> </table>	7	5	2	11	8	2	9	4	6										
2	0	1	2	1																																																			
3	3	1	3	2																																																			
2	1	1	1	0																																																			
3	1	3	2	0																																																			
1	1	2	1	1																																																			
1	0	0																																																					
1	0	0																																																					
0	0	2																																																					
7	5	2																																																					
11	8	2																																																					
9	4	6																																																					

Gambar 2. 7 Operasi Convolution Matrix dot

Pada gambar 2.7 merupakan proses konvolusi antara *image input* dengan *kernel (filter)* terdapat parameter yang menentukan hasil output antara lain:

#### a. Stride

Pada proses konvolusi, *Stride* merupakan sebuah parameter dimana dapat menentukan berapa jumlah pergeseran *filter*. Jika stride 2, jadi *Convolutional filter* bergeser sejauh 2 pixels secara horizontal lalu secara vertical. Semakin kecil stride maka akan semakin detail informasi yang kita dapatkan dari sebuah *input*, namun membutuhkan komputasi yang lebih jika dibandingkan dengan stride yang besar.

#### b. Padding

*Padding* atau *Zero Padding* merupakan sebuah parameter dimana dapat menentukan berapa jumlah pixels (berisi nilai 0) yang akan ditambahkan pada setiap sisi dari *input*. Ini kita gunakan dengan tujuan untuk memanipulasi dimensi output dari conv. *layer*.

(Feature Map). Tujuan menggunakan padding adalah agar dimensi output tetap sama dengan dimensi *input* atau tidak berkurang secara drastis. Karena dimensi *output Convolutional layer* selalu lebih kecil dari pada hasil *input* nya dan hasil *output* ini yang akan kita gunakan kembali untuk *input* an dari *Convolutional layer* selanjutnya, sehingga makin banyak informasi terbuang. Dapat meningkatkan performa dari model karena *Convolutional filter* akan fokus pada informasi yang sebenarnya yaitu yang berada diantara *Zero Padding* tersebut.

0	0	0	0	0	0
0	35	19	25	6	0
0	13	22	16	53	0
0	4	3	7	10	0
0	9	8	1	3	0
0	0	0	0	0	0

Gambar 2. 8 Zero Padding pada Matrix 4x4 menjadi Matrix 6x6

Pada gambar 2.8 diatas merupakan *Zero Padding* pada ukuran matrix 4x4 diubah menjadi matrix ukuran 6x6. Untuk menghitung dimensi dari *Feature Map* kita bisa gunakan rumus pada gambar 2.5 seperti dibawah ini:

$$output = \frac{W - N + 2P}{S} + 1$$

Gambar 2. 5 Rumus Menghitung Dimensi Feature Map

(a) W = Panjang/Tinggi *Input*

(b) N = Panjang/Tinggi *Filter*

(c) P = *Zero Padding*

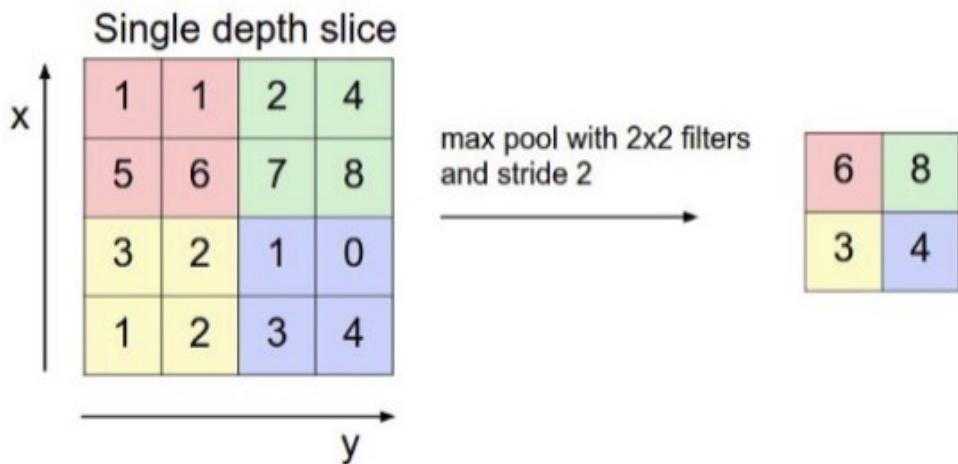
(d) S = *Stride*

c. *Dropout*

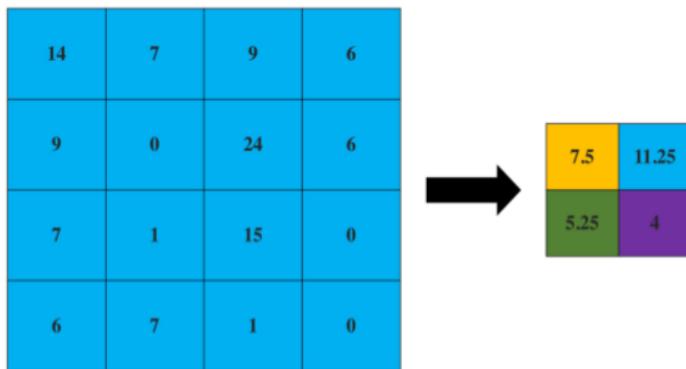
*Dropout* merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* mengacu kepada menghilangkan neuron yang berupa *hidden* maupun *layer* yang *visible* di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. *Neuron* yang akan dihilangkan akan dipilih secara acak. Setiap *neuron* akan diberikan probabilitas yang bernilai antara 0 dan 1.

#### **2.1.2.3 Pooling Layer**

*Pooling Layer* biasanya berada setelah *Convolutional layer*. Pada prinsipnya *Pooling Layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu untuk mengurangi dimensi setiap *Feature Map* tetapi menyimpan informasi yang paling penting. *Pooling* bisa dari berbagai jenis seperti *Max Pooling*, *Average Pooling*, dll. Contoh jika peneliti menggunakan *Max Pooling* dengan ukuran 2x2 dengan *stride* sebanyak 2, maka pada setiap pergeseran *filter* yang dilakukan, nilai maximum pada area ukuran 2x2 pixel tersebut akan dipilih, sedangkan *Average Pooling* akan memilih nilai rata-ratanya. Tujuan kita dalam menggunakan *Pooling Layer* adalah untuk mengurangi dimensi dari *Feature Map* (*down sampling*), sehingga dapat mempercepat komputasi karena lebih mudah dikelola dan dapat mengatasi *overfitting*. Pada gambar 2.9 merupakan proses dari *maximum pooling layer*. Sedangkan pada gambar 2.10 merupakan proses *Average pooling layer*.



Gambar 2. 9 Maximum Pooling Layer



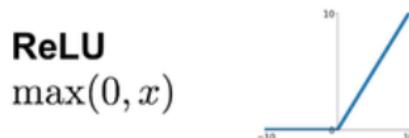
Gambar 2. 10 Average pooling layer

Proses konvolusi dan *pooling* dapat diulang beberapa kali hingga menghasilkan sebuah arsitektur *Convolutional Neural Network* yang diinginkan. Diantara *Convolutional layer* dan *Pooling Layer* umumnya terdapat *Activation layer* (*ReLU*, *softmax*, dll) yang banyak ditemukan pada *artificial Neural Network*. *Activation layer* adalah sebuah node/titik yang mendefinisikan sebuah *output* berdasarkan *input* yang diberikan. Pada CNN *layer* ini dapat didefinisikan sebagai batas (*threshold*) suatu *input* agar

menghasilkan suatu *output*. Jika suatu *input* belum melebihi *threshold* maka tidak akan ada klasifikasi *output*.

#### 2.1.2.4 Activation Function Layer

*Activation Layer* merupakan sebuah *layer* dimana *Feature Map* dimasukkan ke dalam fungsi aktivasi. Fungsi aktivasi digunakan untuk mengubah nilai-nilai pada *Feature Map* pada range tertentu sesuai dengan fungsi aktivasi yang digunakan. *Feature map* bertujuan untuk meneruskan nilai yang menampilkan fitur dominan dari citra yang masuk ke *layer* berikutnya.[18] Aktivasi yang digunakan diantara *convolution* dan *Pooling Layer* yaitu *ReLU* (*Rectified Linear Unit*) dan *Softmax*. *ReLU* adalah metode yang digunakan *Deep Learning network* pada *hidden layer*. Sebuah *ReLU* memiliki *input*  $o$  jika *input* kurang dari  $o$ , jika *input* lebih besar dari  $o$  maka *output* sama dengan *input*. Gambar 2.11 berikut ini merupakan rumus *relu*.



Gambar 2. 11 Rumus *ReLU*

## 2.2 Pengantar Region - Convolution Neural Network (R-CNN)

Masalah yang dicoba diselesaikan oleh sistem R-CNN adalah menemukan objek dalam gambar (deteksi objek). Apa yang Anda lakukan untuk menyelesaikan ini? Anda bisa mulai dengan pendekatan *sliding window*. Saat menggunakan metode ini, Anda cukup melihat keseluruhan gambar dengan ukuran persegi yang berbeda dan melihat gambar yang lebih kecil dalam metode brute

force. Masalahnya adalah Anda akan memiliki sejumlah besar gambar yang lebih kecil untuk dilihat. Untuk keberuntungan kami, orang-orang pintar lainnya mengembangkan algoritma untuk memilih dengan cerdas apa yang disebut proposal kawasan. Untuk menyederhanakan konsep ini: "Proposal kawasan hanyalah sebagian kecil dari gambar asli, yang kami pikir dapat berisi objek yang kami cari."

### **2.2.1 Region Proposal**

Ada berbagai algoritma proposal wilayah yang dapat kita pilih. Ini adalah algoritma "normal" yang bekerja di luar kotak. Kami tidak harus melatih mereka atau apa pun. Dalam kasus makalah ini, mereka menggunakan metode pencarian selektif untuk menghasilkan proposal wilayah. Saya menemukan penjelasan yang sangat bagus dan terperinci tentang bagaimana, algoritma bekerja di sini . Namun perlu diingat:

R-CNN adalah agnostik terhadap metode proposal wilayah. Anda dapat memilih metode apa pun yang Anda suka dan itu akan berhasil baik. Ini akan membuat hampir 2.000 wilayah berbeda yang harus kita perhatikan. Ini terdengar seperti jumlah yang besar, tetapi masih sangat kecil dibandingkan dengan pendekatan jendela geser kasar.

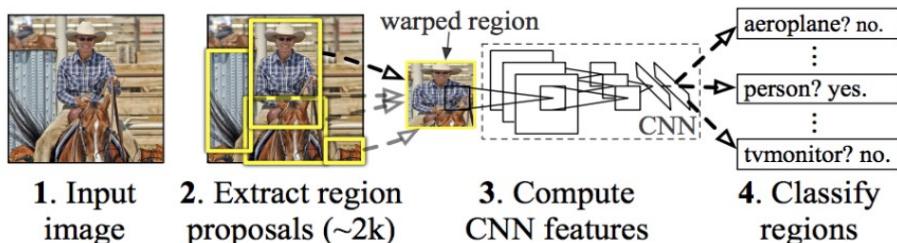
Untuk menghindari masalah memilih sejumlah besar daerah, Ross Girshick et al mengusulkan metode di mana kami menggunakan pencarian selektif untuk mengekstrak hanya 2000 wilayah dari gambar dan dia menyebutnya proposal wilayah. Karena itu, sekarang, alih-alih mencoba mengklasifikasikan

sejumlah besar wilayah, Anda dapat bekerja dengan 2000 wilayah. Proposal 2000 wilayah ini dihasilkan menggunakan algoritma pencarian selektif yang ditulis di bawah ini.

Pencarian Selektif :

- Menghasilkan sub-segmentasi awal, kami menghasilkan banyak kandidat daerah
- item Menggunakan algoritma serakah untuk secara rekursi menggabungkan daerah yang sama menjadi yang lebih besar
- Menggunakan daerah yang dihasilkan untuk menghasilkan proposal daerah kandidat akhir

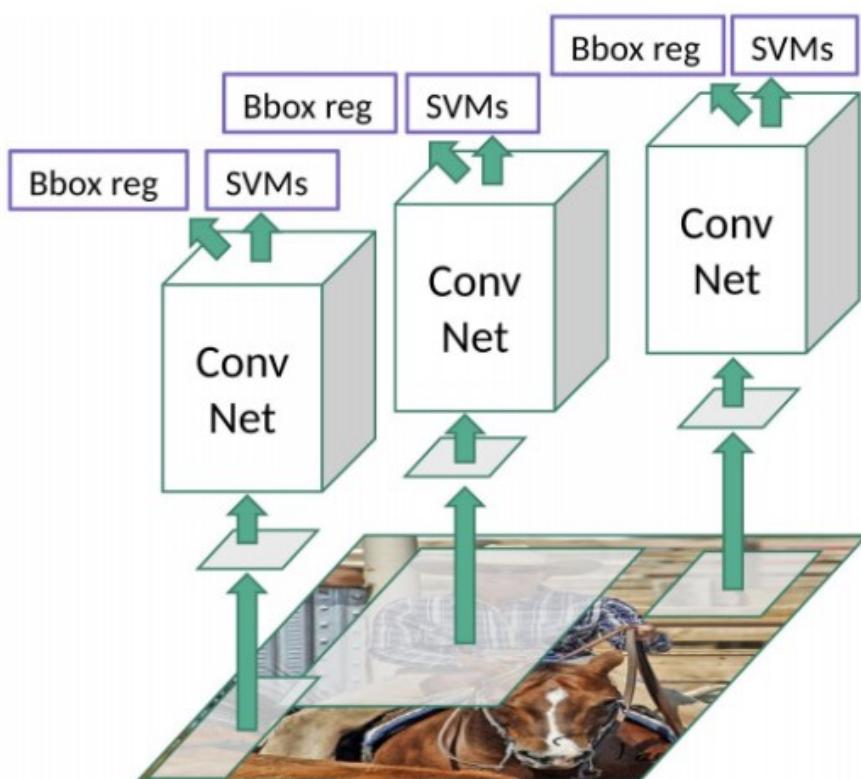
### R-CNN: *Regions with CNN features*



Gambar 2. 12 Arsitektur R-CNN

Gambar 2.12 diatas merupakan arsitektur dari R-CNN. Proposal wilayah kandidat 2000 ini dibengkokkan menjadi kotak dan dimasukkan ke dalam jaringan saraf *Convolutional* yang menghasilkan vektor fitur 4096-dimensi sebagai output. CNN bertindak sebagai ekstraktor fitur dan lapisan padat output terdiri dari fitur yang diekstraksi dari gambar dan fitur yang diekstraksi dimasukkan ke dalam SVM untuk mengklasifikasikan keberadaan objek dalam proposal wilayah kandidat. Selain memprediksi keberadaan objek dalam proposal wilayah, *algoritme* juga

memprediksi empat nilai yang merupakan nilai offset untuk meningkatkan ketepatan kotak pembatas. Misalnya, mengingat proposal wilayah, algoritme akan memperkirakan keberadaan seseorang tetapi wajah orang tersebut di dalam proposal wilayah tersebut dapat dipotong setengahnya. Oleh karena itu, nilai offset membantu menyesuaikan kotak pembatas proposal wilayah. Pada gambar merupakan proses pada R-CNN.



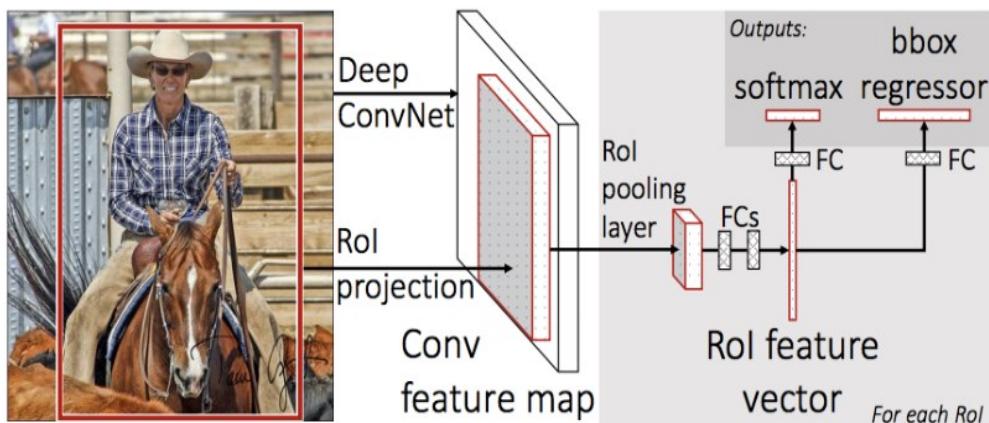
Gambar 2. 13 Proses R-CNN

### 2.2.1 Masalah Dengan R-CNN

Berikut beberapa masalah pada metode R-CNN :

- Masih membutuhkan banyak waktu untuk melatih jaringan karena Anda harus mengklasifikasikan 2.000 proposal wilayah per gambar.
- Itu tidak dapat diimplementasikan real time karena dibutuhkan sekitar 47 detik untuk setiap gambar uji.
- Algoritme pencarian selektif adalah algoritma tetap. Karena itu, tidak ada pembelajaran yang terjadi pada tahap itu. Hal ini dapat mengarah pada pembuatan proposal daerah kandidat yang buruk.

### 2.3 Pengantar Fast Region - Convolution Neural Network (Fater R-CNN)

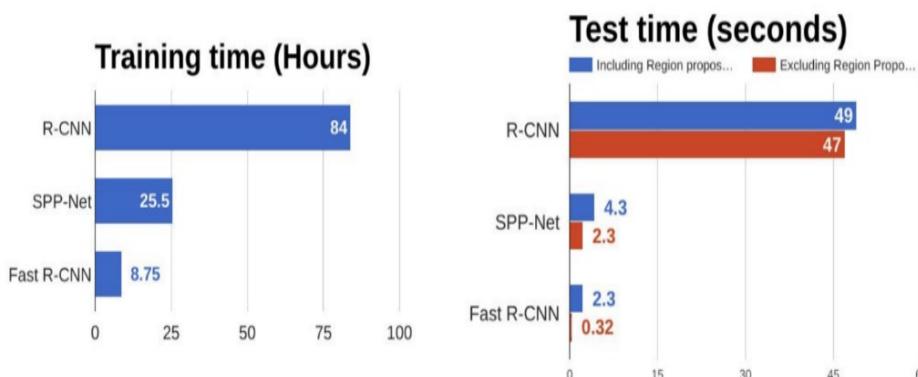


*Gambar 2. 14 Fast R-CNN*

Pada gambar 2.14 di atas merupakan struktur dari proses Fast R-CNN. R-CNN memecahkan beberapa kelemahan R-CNN untuk membangun algoritma deteksi objek yang lebih cepat dan disebut Fast R-CNN. Pendekatannya mirip dengan algoritma R-CNN. Tapi, alih-alih mengumpulkan proposal wilayah ke CNN, kami memberi makan gambar *input* ke CNN untuk menghasilkan peta fitur *Convolutional*. Dari peta fitur konvolusional, kami mengidentifikasi wilayah proposal dan membungkukannya ke dalam kotak dan

dengan menggunakan lapisan pool RoI kami membentuk kembali menjadi ukuran tetap sehingga dapat dimasukkan ke dalam lapisan yang terhubung sepenuhnya.

Dari vektor fitur RoI, kami menggunakan lapisan softmax untuk memprediksi kelas wilayah yang diusulkan dan juga nilai offset untuk kotak pembatas. Alasan "Fast R-CNN" lebih cepat daripada R-CNN adalah karena Anda tidak harus memasukkan 2.000 proposal wilayah ke jaringan saraf *Convolutional* setiap waktu. Sebaliknya, operasi konvolusi dilakukan hanya sekali per gambar dan peta fitur dihasilkan darinya.

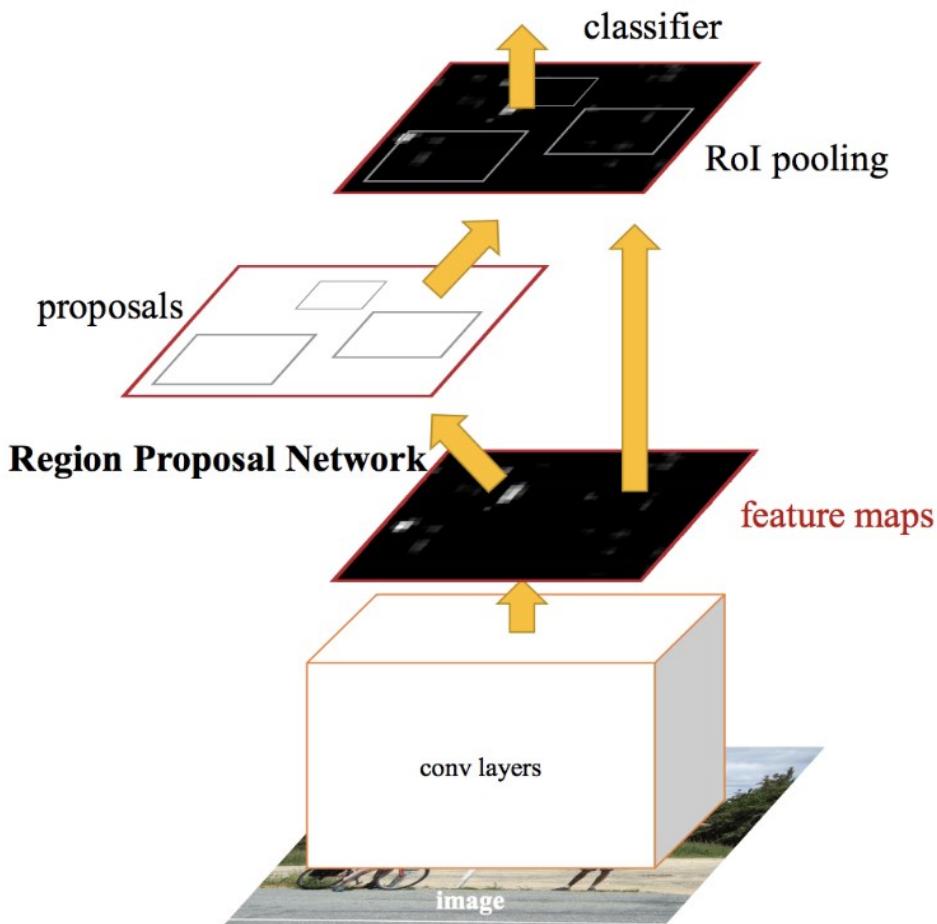


Gambar 2. 15 Perbandingan Algoritma Objek Deteksi

Pada gambar 2.15 diatas merupakan perbandingan algoritme pada objek deteksi. Dari gambar grafik 2.14 dapat disimpulkan bahwa Fast R-CNN secara signifikan lebih cepat dalam sesi pelatihan dan pengujian dibandingkan R-CNN. Ketika Anda melihat kinerja *Fast R-CNN* selama waktu pengujian, termasuk proposal wilayah memperlambat algoritma secara signifikan bila dibandingkan

dengan tidak menggunakan proposal wilayah. Oleh karena itu, proposal kawasan menjadi hambatan dalam algoritma Fast R-CNN yang memengaruhi kinerjanya.

#### **2.4.1 Faster Region - Convolution Neural Network (Faster R-CNN)**

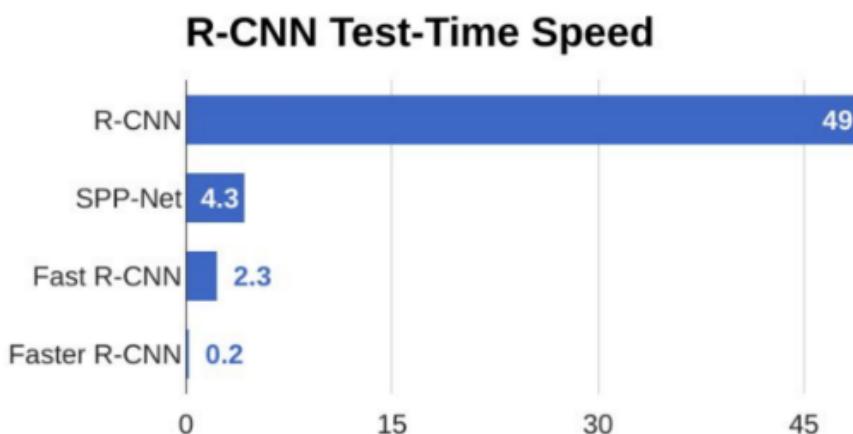


*Gambar 2. 16 Arsitektur Faster R-CNN*

Pada gambar 2.16 diatas merupakan arsitektur dari Faster R-CNN. Kedua algoritma di atas (R-CNN dan Cepat R-CNN) menggunakan pencarian selektif untuk mengetahui proposal kawasan. Pencarian selektif adalah proses yang lambat dan

memakan waktu yang mempengaruhi kinerja jaringan. Oleh karena itu, Shaoqing Ren et al datang dengan algoritma deteksi objek yang menghilangkan algoritma pencarian selektif dan memungkinkan jaringan mempelajari proposal wilayah.

Mirip dengan Fast R-CNN, gambar disediakan sebagai *input* ke jaringan *Convolutional* yang menyediakan peta fitur *Convolutional*. Alih-alih menggunakan algoritma pencarian selektif pada peta fitur untuk mengidentifikasi proposal wilayah, jaringan yang terpisah digunakan untuk memprediksi proposal wilayah. Proposal wilayah yang diprediksi kemudian dibentuk kembali menggunakan lapisan pool RoI yang kemudian digunakan untuk mengklasifikasikan gambar dalam wilayah yang diusulkan dan memprediksi nilai offset untuk kotak pembatas.



*Gambar 2. 17 Perbandingan Kecepatan Uji Waktu Dari Algoritma Objek Deteksi*

Dari grafik pada gambar 2.17 di atas, Anda dapat melihat bahwa R-CNN lebih cepat jauh lebih cepat daripada pendahulunya. Oleh karena itu, bahkan dapat digunakan untuk deteksi objek real-

time. Pada awalnya metode R-CNN bertujuan untuk fokus pada mengembangkan kehandalan akurasi dalam mendeteksi objek. Sebelum faster R-CNN diperkenalkan, berikut perkembangan metode R-CNN seiring berjalannya waktu.



Gambar 2. 18 Faster R-CNN

Metode faster *RCNN* menggunakan *RPN* (*Region Proposal Network*) untuk menentukan kandidat *boundary box*. Setiap kandidat *boundary box* akan masuk tahap klasifikasi untuk mendeteksi kelas yang terdapat pada *boundary box* tersebut. RPN ini merupakan pengganti Selective Search pada metode sebelumnya yaitu *Fast R-CNN*. Dibandingkan *SSD* dan *YOLO*, *Faster R-CNN* ini dinilai lebih lambat karena terlalu banyak kandidat box yang perlu masuk bagian klasifikasi.

Tahapan Faster R-CNN adalah sebagai berikut :

1. Kalkulasi kandidat *bounding box* menggunakan RPN (*Region Proposal Network*)

Berikut tahapan RPN :

- *Input* gambar pertama kali akan masuk kedalam CNN untuk mereduksi dimensi gambar serta untuk memperoleh feature map
- Pada *layer* terakhir dari CNN diatas dilakukan konvolusi lagi menggunakan sliding window dengan ukuran 3x3
- Untuk setiap sliding window menghasilkan sejumlah n tetap anchor box

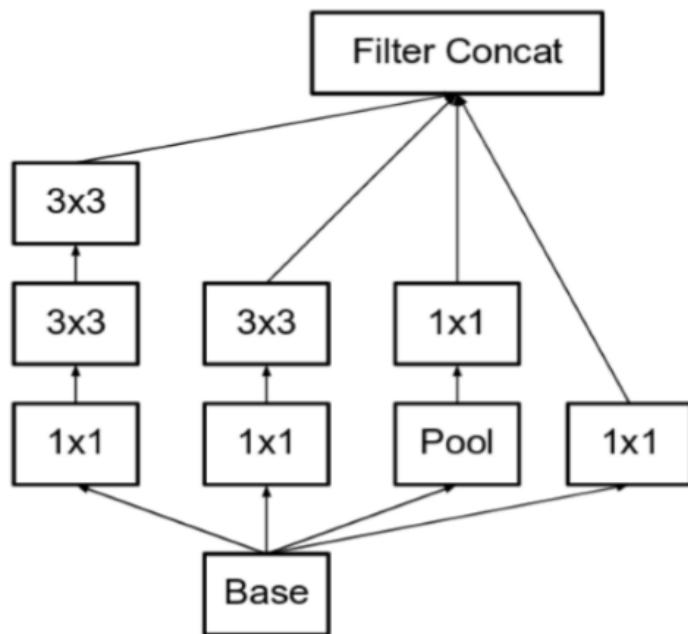
- Lalu dilakukan seleksi untuk setiap box yang dihasilkan sesuai dengan skor adanya objek pada box tersebut
2. Jalankan CNN untuk setiap bounding box yang dihasilkan dari RPN
  3. Setiap output yang dihasilkan CNN menjadi *input* SVM (*Support Vector Machine*) untuk klasifikasi kelasnya dan juga sebagai *input* untuk linear regression untuk memperbaiki bounding box agar lebih presisi

Faster RCNN merupakan salah satu metode *Deep Learning* yang digunakan untuk mengenali suatu suatu objek pada citra. Pengenalan dilakukan dengan menelusuri ciri-ciri yang dimiliki oleh objek pada citra. Penelusuran dilakukan melalui sejumlah *layer* (seperti yang dilakukan pada *Neural Network*) melalui proses konvolusi atau yang lebih dikenal dengan nama *Convolutional Neural Network* (CNN).

CNN memiliki berbagai arsitektur, salah satunya dalam penelitian ini adalah *Inception V2*. Arsitektur dari *Inception V2* dirancang untuk mengurangi kompleksitas CNN, yang dilakukan dengan cara menyusun arsitektur yang lebih melebar dari pada mendalam. *Inception V2* memiliki 3 modul yang ditunjukkan oleh Gambar 4. Modul pertama (Gambar 4.a) menggantikan konvolusi  $5 \times 5$  menjadi  $3 \times 3$ . Selanjutnya pemfaktoran konvolusi dilakukan. Terakhir modul diubah lebih melebar untuk mengurangi kompleksitas jaringan konvolusi .

- a. Modul Pertama *Inception V2*

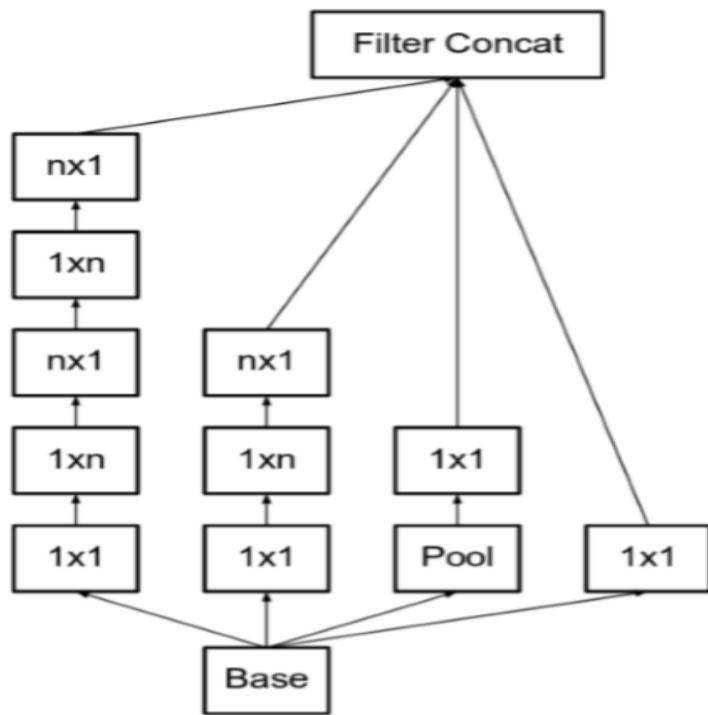
Pada gambar 2.19 dibawah berikut merupakan modul pertama pada inception V2.



Gambar 2. 19 Modul Pertama Inception V2

b. Modul kedua Inception V2

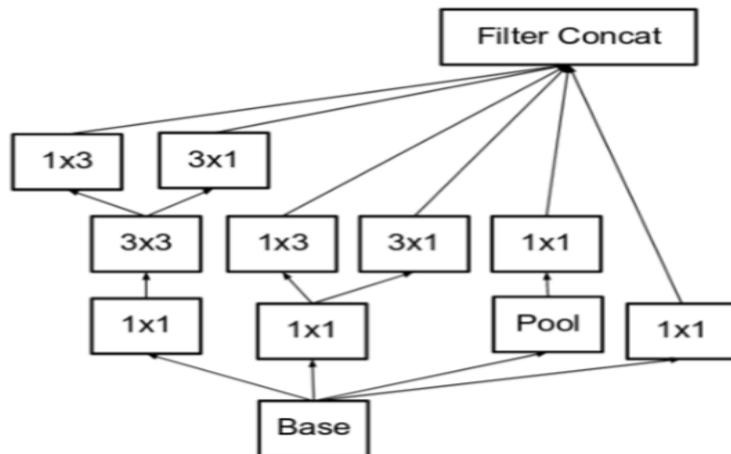
Pada gambar 2.20 dibawah berikut merupakan modul kedua pada inception V2.



Gambar 2. 20 Modul kedua Inception V2

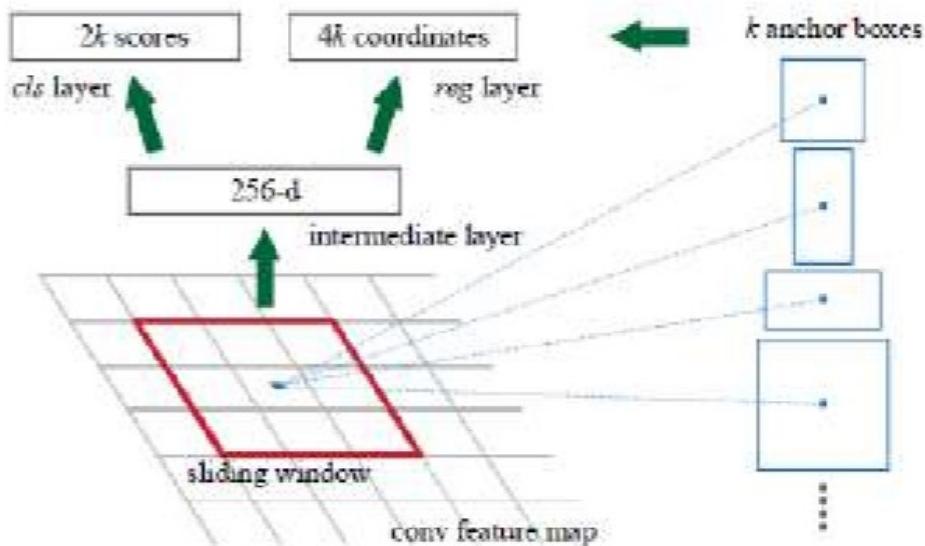
### c. Modul Ketiga Inception V2

Pada gambar 2.21 dibawah berikut merupakan modul ketiga pada inception V2.



Gambar 2. 21 Modul Ketiga Inception V2

Selanjutnya, berbeda dengan metode RCNN sebelumnya, maka *Faster-RCNN* membuat perubahan dengan memunculkan *Region Proposal Network* (RPN) ditunjukkan oleh Gambar 2.22 berikut:



Gambar 2. 22 Region Proposal Network (RPN) [17]

## 2.4 Cross Validation

*Cross Validation*, kadang-kadang disebut estimasi rotasi atau pengujian *out-of-sample*, adalah salah satu dari berbagai teknik validasi model yang serupa untuk menilai bagaimana hasil analisis statistik akan di generalisasi ke set data independen. Hal ini terutama digunakan dalam pengaturan di mana tujuannya adalah prediksi, dan satu ingin memperkirakan berapa akurat suatu model prediktif akan tampil dalam praktik. Pada saat prediksi, sebuah model biasanya diberikan *dataset* data yang diketahui di mana data set untuk digunakan sebagai *dataset* pelatihan dan data set yang tidak diketahui atau biasa disebut dengan *dataset testing* terhadap

model yang diuji (disebut *dataset* validasi atau set pengujian).

Tujuan dari *Cross Validation* adalah mengevaluasi model dalam memprediksi data baru yang tidak digunakan dalam memperkirakannya, untuk menandai masalah seperti *overfitting* atau bias seleksi dan untuk memberikan wawasan tentang bagaimana model akan di generalisasi ke *dataset* independen (yaitu, *dataset* tidak dikenal, misalnya dari masalah nyata).

Satu putaran *cross-validation* melibatkan partisi sebuah sampel dari data yang menjadi pelengkap subset, melakukan analisis pada satu bagian (disebut *training set*), dan memvalidasi analisis pada subset lain (disebut validasi set atau pengujian set). Untuk mengurangi variabilitas, dalam sebagian besar metode, beberapa putaran validasi silang dilakukan menggunakan partisi yang berbeda, dan hasil validasi digabungkan (misalnya di rata-rata) selama putaran untuk memberikan perkiraan kinerja prediksi model.

## 2.5 Confusion Matrix

*Confusion Matrix* adalah salah satu metode yang dapat digunakan untuk mengukur kinerja suatu metode klasifikasi dan pada dasarnya *Confusion Matrix* mengandung informasi yang dapat membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya. Pada dasarnya confusion matrix mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya.

Berdasarkan jumlah keluaran kelasnya, sistem klasifikasi dapat dibagi menjadi 4 (empat) jenis yaitu klasifikasi binary, multi-

class, multi-label dan hierarchical. Pada klasifikasi binary, data masukan dikelompokkan ke dalam salah satu dari dua kelas. Jenis klasifikasi ini merupakan bentuk klasifikasi yang paling sederhana dan banyak digunakan. Contoh penggunaannya antara lain dalam sistem yang melakukan deteksi orang atau bukan, sistem deteksi kendaraan atau bukan, dan sistem deteksi pergerakan atau bukan.

Sementara itu, pada bentuk klasifikasi multi-class, data masukan diklasifikasikan menjadi beberapa kelas. Sebagai contoh sistem yang dapat mengklasifikasikan jenis kendaraan seperti sepeda, sepeda motor, mobil, bus, truk, dan sebagainya. Bentuk klasifikasi multi-label pada dasarnya sama dengan multi-class dimana data dikelompokkan menjadi beberapa kelas, namun pada klasifikasi multi-label, data dapat dimasukkan dalam beberapa kelas sekaligus. Bentuk klasifikasi yang terakhir adalah hierarchical. Data masukan dikelompokkan menjadi beberapa kelas, namun kelas tersebut dapat dikelompokkan kembali menjadi kelas-kelas yang lebih sederhana secara hirarkis. Contohnya dalam penelitian ini, arah pergerakan dikelompokkan menjadi 12 arah pergerakan yang tentunya dapat disederhanakan menjadi 4 arah.

Pada pengukuran kinerja menggunakan confusion matrix, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Nilai True Negative (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan False Positive (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, True

Positive (TP) merupakan data positif yang terdeteksi benar. False Negative (FN) merupakan kebalikan dari True Positive, sehingga data positif, namun terdeteksi sebagai data negatif. Pada jenis klasifikasi binary yang hanya memiliki 2 keluaran kelas, confusion matrix dapat disajikan seperti pada tabel 2.1 berikut ini.

*Tabel 2. 1 Tabel Confusion Matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
<b>Positif</b>	TP (True Positive)	FN (False Negative)
<b>Negatif</b>	FP (False Positive)	TN (True Negative)

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data.

Pada tahap ini akan dijelaskan bagaimana nilai *Confusion Matrix* didapat berdasarkan penggunaan rumus-rumus yang ada dibawah ini, sebagai berikut :

1. Pada tahap ini dapat dijelaskan bahwa hasil dengan nilai 99-100 persen dapat menjadi sangat baik, bagus, biasa-biasa saja, buruk atau mengerikan itu semua tergantung pada masalahnya.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

*Gambar 2. 23 Rumus Perhitungan Nilai Accuracy*

2. Pada tahap ini dapat dijelaskan bahwa *recall* dapat didefinisikan sebagai rasio dari jumlah total contoh positif

yang di klasifikasikan dengan benar dan nilai *recall* yang baik menunjukkan klasifikasi dikenali dengan benar.

$$Recall = \frac{TP}{TP + FN}$$

*Gambar 2. 24 Rumus Perhitungan Nilai Recall*

3. Pada tahap ini dapat dijelaskan bahwa nilai presisi didapatkan dari pembagian jumlah total contoh positif yang diklasifikasikan dengan benar dengan jumlah total contoh positif yang diprediksi.

$$Precesion = \frac{TP}{TP + FP}$$

*Gambar 2. 25 Rumus Perhitungan Nilai Precision*

---

## BAB 3

## **PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN**

---

### **3.1 Tools Yang Digunakan**

#### **3.1.1 Anaconda**

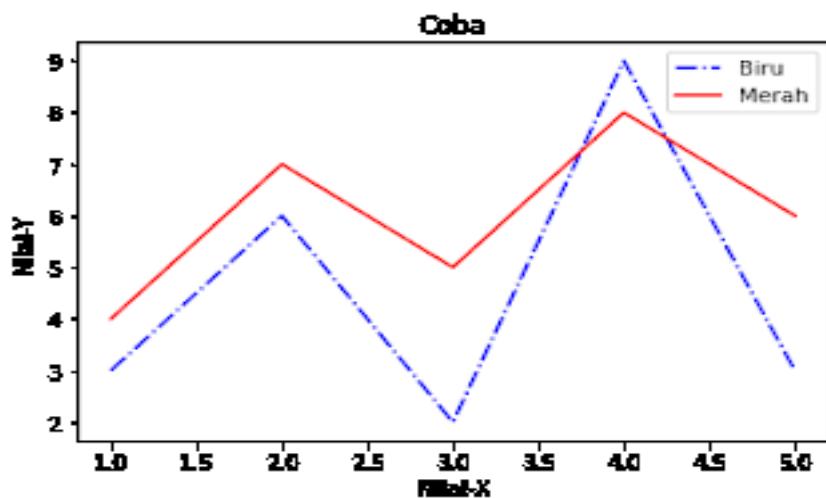


*Gambar 3. 1 Logo Anaconda*

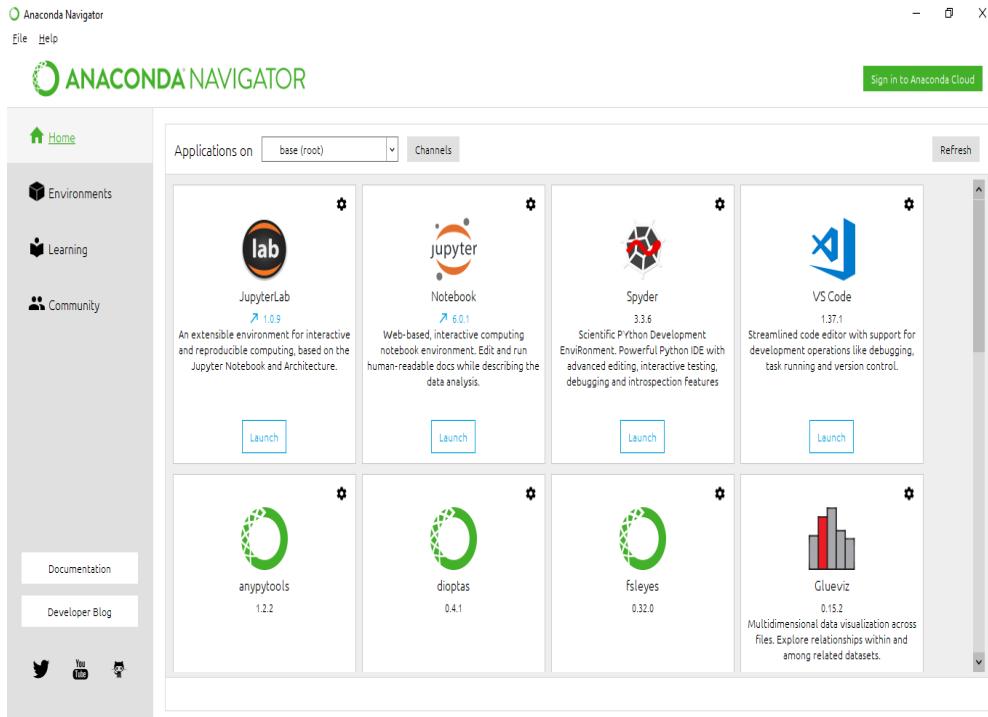
Gambar 3.1 diatas merupakan logo dari *Anaconda*. Tools *Anaconda* merupakan platform ilmu data berbasis di sekitar bahasa pengkodean *Python*. *Anaconda* diciptakan agar mempermudah pengguna me manajemen paket *Python*. Dengan menggunakan *Anaconda*, maka versi dari paket yang ada, di manajemen oleh package management system conda. *Anaconda* telah digunakan oleh lebih dari 12jt pengguna, dan merupakan yang terpopuler sebagai distribusi *Python* dan R, dan memiliki lebih dari 1400 paket data science yang cocok untuk *Windows*, *Linux*, dan *MacOS*.

Sejumlah manfaat inti termasuk peningkatan kolaborasi antar departemen, kemampuan untuk mereproduksi data,

skalabilitas superior dan berbagai lapisan keamanan. Fitur utama lainnya dari paket ini adalah memungkinkan organisasi untuk mengelola dan menafsirkan data besar dengan lebih baik; faktor kunci untuk sukses dalam lingkungan bisnis modern. Pada gambar 3.2 dibawah ini merupakan grafik implementasi matplotlib pada *Python*.



Gambar 3. 2 grafik implementasi matplotlib pada Python



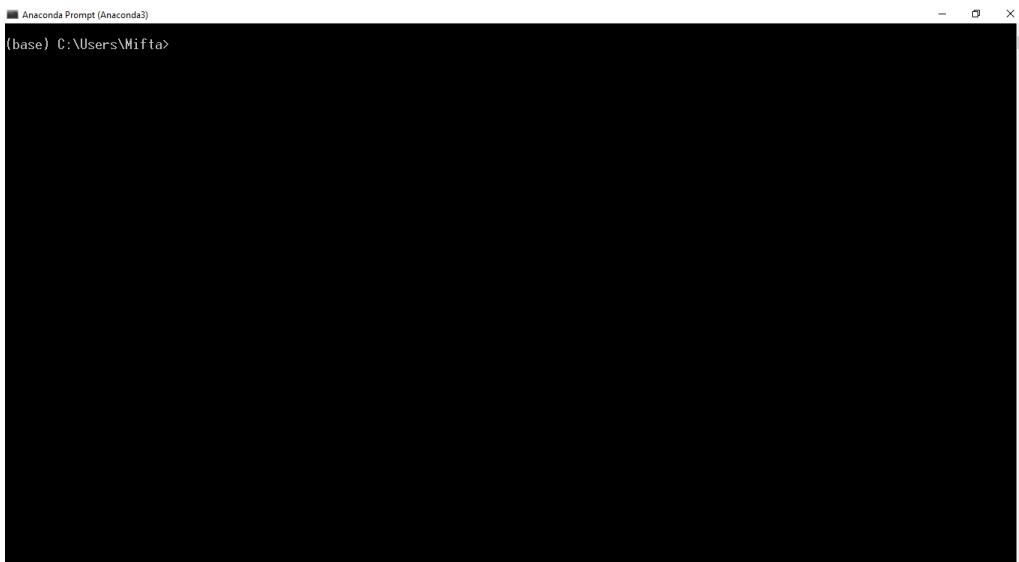
Gambar 3. 3 Tampilan Anaconda Navigator

Gambar 3.3 diatas merupakan tampilan dari *anaconda navigator*. *Anaconda Navigator* adalah antarmuka pengguna dengan grafis desktop (GUI) yang termasuk dalam distribusi *Anaconda*. Hal ini memungkinkan kita untuk menjalankan aplikasi yang disediakan dalam distribusi *Anaconda* dan dengan mudah mengelola paket conda, lingkungan, dan saluran tanpa menggunakan perintah baris-per baris. Ini tersedia untuk Windows, macOS dan Linux.

Aplikasi yang disediakan dalam Distribusi *Anaconda*:

- JupyterLab
- Jupyter Notebook
- QT Console
- Spyder

- VS Code
- Glueviz
- Orange 3
- Rstudio



*Gambar 3. 4 Tampilan Anaconda Desktop*

Gambar 3.4 Diatas merupakan tampilan *Anaconda Desktop*. *Anaconda* merupakan sebuah distribusi bahasa pemrograman *Python* dan *R* untuk keperluan komputasi ilmiah (scientific computing) seperti data science, machine learning, data processing skala-luas, analisis prediksi, dan lain sebaginya. Menilik dari kemampuan *Python* sebagai bahasa pemrograman yang sangat "powerful". Maka, sebagai contoh dengan *Anaconda* kamu dapat menggambarkan grafik - grafik menarik, sebagai penjabaran dari analisa kamu terhadap suatu hal.

*Anaconda* memiliki sifat yang modular, sehingga dapat disesuaikan tergantung kebutuhan organisasi yang bersangkutan. Karena hal itu mendorong kolaborasi waktu nyata, tingkat efisiensi

internal juga akan ditingkatkan. User dapat menikmati dukungan teknis real-time selama penyebaran kode open-source dan karena *Anaconda* sepenuhnya kompatibel dengan bahasa *Python*, kurva pembelajaran secara keseluruhan telah berkurang secara dramatis.

### **Fitur dan Aplikasi yang disediakan dalam distribusi *Anaconda***

Fitur :

- *Anaconda Navigator*
- Pengkompilasian dengan rilisan *Python* terbaru
- Keandalan yang lebih baik
- Peningkatan kinerja CPU
- Lebih dari 230 paket yang diperbarui dan ditambahkan

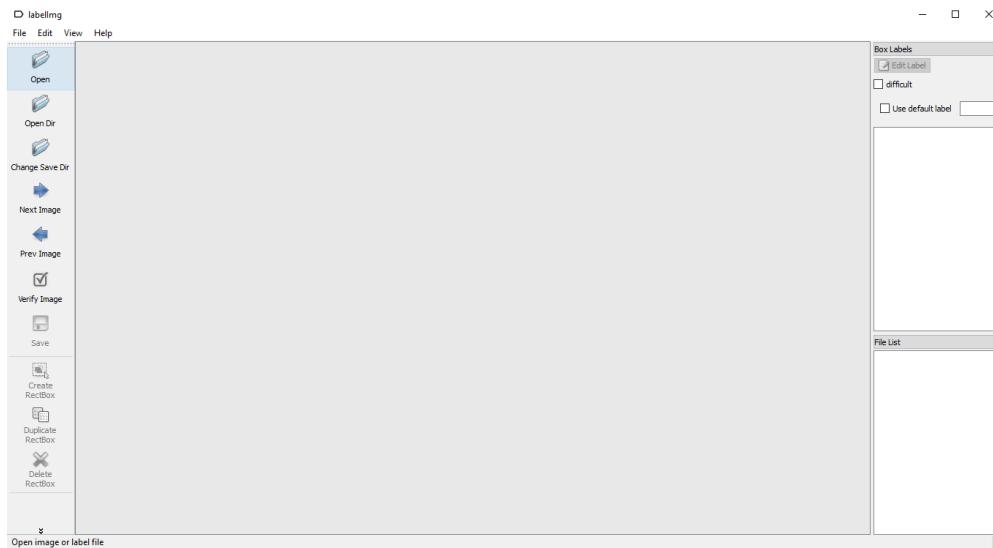
#### **3.1.2 LabelImg**



*Gambar 3. 5 Logo LabelImg*

Pada gambar 3.5 diatas merupakan logo dari aplikasi labelImg yang digunakan untuk memerikan label pada suatu gambar. Salah satu alat Anotasi Gambar untuk Penglihatan Komputer adalah LabelImg. LabelImg adalah alat anotasi gambar grafis open source yang dapat Anda gunakan untuk memberi label pada kotak

pembatas objek dalam gambar. LabelImg adalah alat anotasi gambar grafis. Ini ditulis dalam *Python* dan menggunakan Qt untuk antarmuka grafisnya. Anotasi disimpan sebagai file XML dalam format PASCAL VOC, format yang digunakan oleh *ImageNet*. Selain itu, ia juga mendukung format YOLO.



Gambar 3. 6 Tampilan LabelImg

Pada gambar 3.6 merupakan tampilan awal pada aplikasi labelimg. LabelImg adalah alat anotasi gambar grafis. Ini ditulis dalam *Python* dan menggunakan Qt untuk antarmuka grafis nya. Anotasi disimpan sebagai file XML dalam format PASCAL VOC, format yang digunakan oleh *ImageNet*. Untuk mendapatkan aplikasi Labelimg tersebut kita bisa langsung men download nya pada google.

Pada aplikasi LabelImg tersebut terdapat beberapa menu seperti menu untuk membuka file yang akan kita beri label, menu untuk menyimpan hasil dari pekerjaan kita, menu untuk next dan prev

pada gambar, menu untuk membuat *rectBox*, menu untuk men *duplicate RectBox* dan masih banyak lagi menu lainnya.

### 3.1.3 Sublime



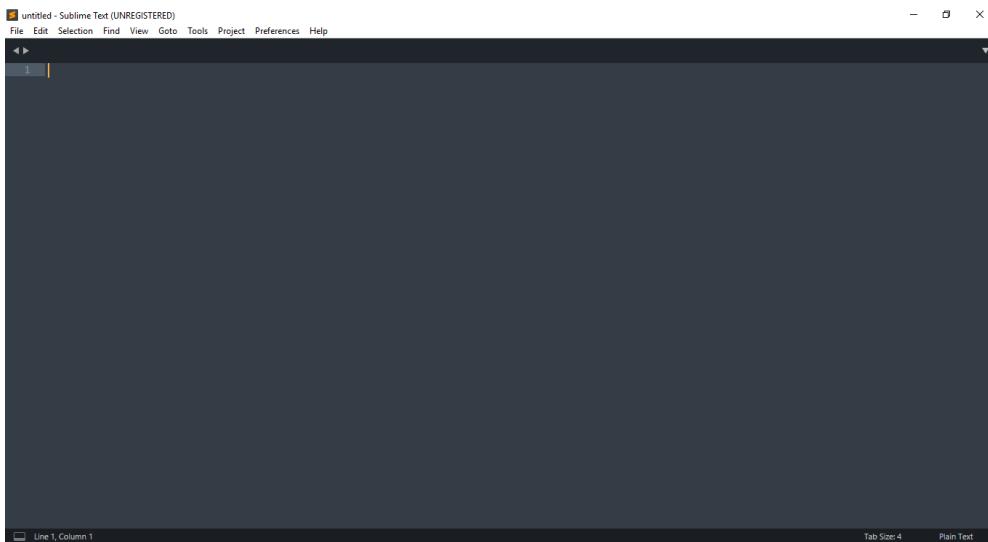
Gambar 3. 7 Logo Sublime Text

Pada gambar 3.7 merupakan logo pada aplikasi sublime text. Aplikasi Sublime Text merupakan aplikasi editor yang digunakan untuk kode dan teks yang dapat berjalan diberbagai platform operating system dengan menggunakan teknologi Phyton API. Aplikasi ni diciptakan karena terinspirasi dari aplikasiVim, Aplikasi ini sangatlah fleksibel dan powerfull. Aplikasi ini dapat dikembangkan dengan menggunakan sublime-packages.

Sublime Text bukanlah aplikasi opensource dan juga aplikasi yang dapat digunakan dan didapatkan secara gratis, namun beberapa fitur pengembangan fungsionalitas (packages) dari aplikasi ini merupakan hasil dari temuan dan memperoleh dukungan penuh dari komunitas serta memiliki linsensi aplikasi gratis. Aplikasi ini mendukung berbagai bahasa pemrograman dan

mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas.

Sublime Text mendukung berbagai bahasa pemrograman dan mampu menyajikan fitur syntax highlight hampir di semua bahasa pemrogramman yang didukung ataupun dikembangkan oleh komunitas seperti; C, C++, C#, CSS, D, Dylan, Erlang, HTML, Groovy, Haskell, Java, JavaScript, LaTeX, Lisp, Lua, Markdown, MATLAB, OCaml, Perl, PHP, Python, R, Ruby, SQL, TCL, Textile and XML. Biasanya bagi bahasa pemrograman yang didukung ataupun belum terdukung secara default dapat lebih dimaksimalkan atau didukung dengan menggunakan add-ons yang bisa didownload sesuai kebutuhan user. Pada gambar 3.8 dibawah ini merupakan tampilan awal pada saat kita membuka aplikasi sublime text.



Gambar 3. 8 Tampilan Sublime Text

Berikut beberapa fitur yang diunggulkan dari aplikasi Sublime Text:

- *Go to Anything*

Ini digunakan untuk membuka file diawali dengan menarik satu project file yang sedang kita kerjakan pada sublime kemudian dengan menekan CTRL+P maka kita dapat mencari file apa yang akan kita buka dengan menuliskan nama file nya, juga kita dapat mencari baris kata dengan keyword # atau @ untuk melompat ke simbol setelah menekan CTRL+P.

- *Multiple Selections*

Berfungsi untuk membuat perubahan pada code pada saat yang sama dalam beberapa baris yang berbeda. Bisa dicoba dengan menekan CTRL+L, atau bisa mengganti kata yang sama dalam baris berbeda dengan memblok kata yang akan diganti kemudian tekan CTRL+D.

- *Command Pallete*

Untuk fungsi yang satu ini jarang digunakan sebenarnya namun dalam fungsi ini banyak sekali hal yang bisa dilakukan seperti menutup semua file, convert case: lower case, remove tag dan masih banyak lagi. Dengan menekan CTRL+SHIFT+P

- *Distraction Free Mode*

Digunakan untuk merubah tampilan menjadi layar penuh, dibutuhkan ketika user akan fokus pada kerjaan yang dikerjakan maka dengan fungsi ini akan sangat membantu dengan menekan SHIFT+F11.

- *Split Editing*

Dapatkan hasil yang maksimal dari monitor layar lebar dengan dukungan editing perpecahan. Melakukan editing di sisi file dengan sisi, atau mengedit dua lokasi di satu file. Anda dapat mengedit dengan banyak baris dan kolom yang user inginkan.

- *Instant Project Switch*

Mengambil seluruh file yang dimasukkan kedalam project pada aplikasi ini terhubung dengan fitur *Go to Anything* untuk menjelajahi semua file yang ada ataupun untuk beralih ke file dalam project lainnya dengan cepat.

- *Plugin API*

Aplikasi ini memiliki plugin API berbasis *Python* sehingga membuat aplikasi ini sangat tangguh.

- *Customize Anything*

Aplikasi sublime mempunyai dan memberikan user fleksibilitas dalam hal pengaturan fungsional dalam aplikasi ini.

- *Cross Platform*

Aplikasi ini dapat berjalan hampir di semua operating system modern seperti Windows, OS X, dan Linux based operating system.

### **3.1.4 CUDA**

*CUDA (Compute Unified Device Architecture)* adalah platform komputasi paralel dan model antarmuka pemrograman aplikasi (API) yang dibuat oleh Nvidia. Ini memungkinkan pengembang perangkat lunak dan insinyur perangkat lunak untuk menggunakan unit pemrosesan grafis (GPU) yang diaktifkan CUDA untuk pemrosesan tujuan umum - suatu pendekatan yang disebut GPGPU

(komputasi Tujuan Umum pada Unit Pemrosesan Grafik). CUDA adalah sebuah Platform yang memiliki lapisan perangkat lunak yang memberikan akses langsung ke set instruksi virtual GPU dan elemen komputasi paralel, untuk eksekusi *kernel* komputasi.

CUDA adalah platform komputasi paralel dan model antarmuka pemrograman aplikasi yang dibuat oleh Nvidia dan memungkinkan pengembang perangkat lunak dapat menggunakan untuk melakukan suatu kesatuan proses grafik yang diaktifkan CUDA. Oleh karena itu dalam model CUDA terdapat CUDA yang digunakan untuk mengalokasikan memori pada perangkat dengan tujuan untuk melakukan peluncuran *kernel* sedangkan CUDA yang digunakan untuk memproses satuan grafik. Berdasarkan referensi tersebut, CUDA digunakan oleh peneliti untuk membantu proses kerja yang terjadi pada *TensorFlow* dalam memproses sebuah grafik yang terdapat pada sebuah gambar.

Platform CUDA dirancang untuk bekerja dengan bahasa pemrograman seperti C, C ++, dan Fortran. Aksesibilitas ini memudahkan spesialis dalam pemrograman paralel untuk menggunakan sumber daya GPU, berbeda dengan API sebelumnya seperti Direct3D dan OpenGL, yang membutuhkan keterampilan tingkat lanjut dalam pemrograman grafis. GPU yang didukung CUDA juga mendukung kerangka kerja pemrograman seperti OpenACC dan OpenCL dan HIP dengan mengkompilasi kode tersebut ke CUDA. Ketika CUDA pertama kali diperkenalkan oleh Nvidia, namanya adalah singkatan dari Compute Unified Device Architecture, tetapi

Nvidia kemudian membatalkan penggunaan umum dari akronim tersebut.

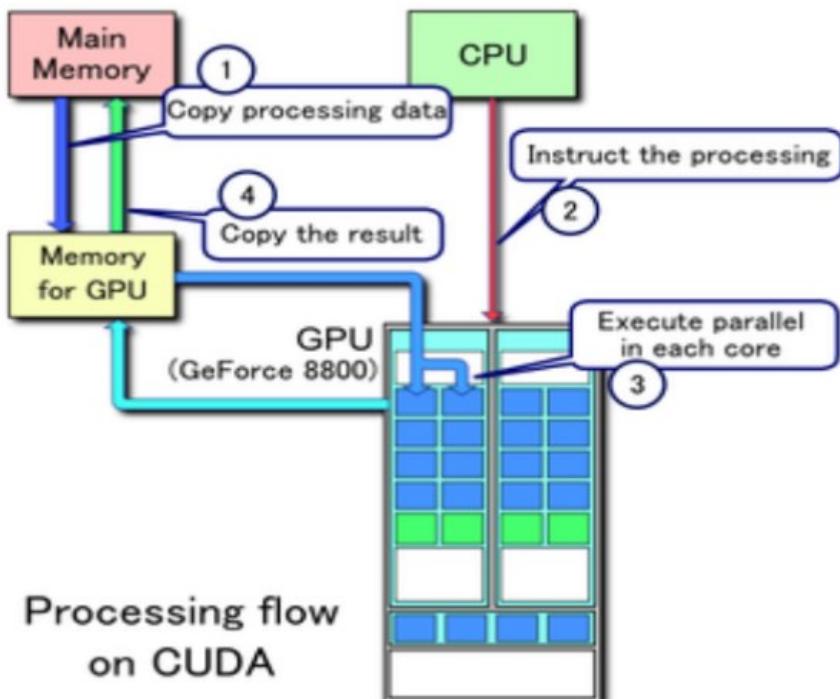
CUDA bisa diakses oleh pengembang perangkat lunak melalui perpustakaan yang dipercepat CUDA, arahan penyusun seperti OpenACC , dan ekstensi ke bahasa pemrograman standar industri termasuk C , C ++ dan Fortran . Pemrogram C / C ++ dapat menggunakan 'CUDA C / C ++', dikompilasi dengan nvcc , kompiler C / C ++ berbasis LLVM Nvidia. Pemrogram Fortran dapat menggunakan 'CUDA Fortran', yang dikompilasi dengan kompiler PGI CUDA Fortran dari The Portland Group.

Selain perpustakaan, arahan kompiler, CUDA C / C ++ dan CUDA Fortran, platform CUDA mendukung antarmuka komputasi lainnya, termasuk OpenCL Khronos Group, Microsoft DirectCompute, OpenGL Compute Shaders, dan C ++ AMP. Pembungkus pihak ketiga juga tersedia untuk *Python*, Perl, Fortran, Jawa, Ruby, Lua, Common Lisp, Haskell, R, MATLAB, IDL, Julia dan dukungan asli di Mathematica.\

ada industri game komputer , GPU digunakan untuk rendering grafik, dan untuk perhitungan fisika game (efek fisik seperti puing-puing, asap, api, cairan); contohnya termasuk PhysX dan Bullet . CUDA digunakan untuk mempercepat aplikasi non-grafis dalam biologi komputasi, kriptografi , dan bidang lainnya dengan urutan besarnya atau lebih.

CUDA memiliki beberapa keunggulan dibandingkan perhitungan tradisional untuk keperluan umum pada GPU (GPGPU) menggunakan API grafik:

- Pembacaan yang tersebar - kode dapat membaca dari alamat yang berubah-ubah dalam memori
- Memori virtual terpadu (CUDA 4.0 dan lebih tinggi)
- Memori terpadu (CUDA 6.0 ke atas)
- Memori bersama - CUDA memperlihatkan wilayah memori bersama cepat yang dapat dibagi di antara utas. Ini dapat digunakan sebagai cache yang dikelola pengguna, memungkinkan bandwidth yang lebih tinggi daripada yang dimungkinkan menggunakan pencarian tekstur
- Unduhan dan pembacaan lebih cepat ke dan dari GPU
- Dukungan penuh untuk operasi integer dan bitwise, termasuk pencarian tekstur integer



*Gambar 3. 9 Aliran Pemrosesan CUDA*

## **3.2 Bahasa Pemrograman Yang Digunakan**

### **3.2.1 Python**



*Gambar 3. 10 Logo Python*

#### **3.1.2.1 Pengertian Python**

Pada gambar 3.10 diatas merupakan logo pada bahasa pemrograman python. Bahasa pemrograman yang interpretatif dan multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode salah satunya adalah Bahasa *Python*. Memiliki fungsi sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas yang memiliki fungsionalitas pustaka standar yang besar serta komprehensif. Bahasa pemograman *Python* juga didukung oleh komunitas yang besar. Bahasa ini banyak digunakan padaprogrammer karena dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami.

Bahasa pemograman ini juga mensupport multi paradigma pemrograman, utamanya namun tidak dibatasi pada pemrograman berorientasi objek, pemrograman imperatif, dan pemrograman fungsional. Beberapa fitur yang tersedia pada Bahasa *Python* adalah sebagai bahasa pemrograman dinamis yang dilengkapi dengan manajemen memori otomatis. *Python* pada umumnya digunakan

sebagai bahasa skrip meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Pemograman *Python* juga dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi.

*Python* saat ini bias dijalankan di berbagai platform system operasi, diantaranya adalah sebagai berikut :

- Linux/Unix
- Windows
- Mac OS X
- Java Virtual Machine
- OS/2
- Amiga
- Palm
- Symbian (untuk produk-produk Nokia)

Bahasa pemograman ini dihubungkan dengan beberapa lisensi yang berbeda dari beberapa versi. Lihat sejarahnya di *Python Copyright*. Namun pada prinsipnya *Python* dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial.

### **3.1.2.2 Sejarah *Python***

*Python* dikembangkan oleh Guido van Rossum pada tahun 1990 di Stichting Mathematisch Centrum (CWI), Amsterdam sebagai kelanjutan dari bahasa pemrograman ABC. Versi terakhir yang dikeluarkan CWI adalah 1.2.

Tahun 1995, Guido pindah ke CNRI di Virginia Amerika sambil terus melanjutkan pengembangan *Python*. Versi terakhir yang dikeluarkan adalah 1.6. Tahun 2000, Guido dan para pengembang inti *Python* pindah ke BeOpen.com yang merupakan sebuah perusahaan komersial dan membentuk BeOpen PythonLabs. *Python* 2.0 dikeluarkan oleh BeOpen. Setelah mengeluarkan *Python* 2.0, Guido dan beberapa anggota tim PythonLabs pindah ke DigitalCreations.

Saat ini pengembangan *Python* terus dilakukan oleh sekumpulan pemrogram yang dikoordinir Guido dan *Python* Software Foundation. *Python* Software Foundation adalah sebuah organisasi non-profit yang dibentuk sebagai pemegang hak cipta intelektual *Python* sejak versi 2.1 dan dengan demikian mencegah *Python* dimiliki oleh perusahaan komersial. Saat ini distribusi *Python* sudah mencapai versi 2.7.14 dan versi 3.6.3

Nama *Python* dipilih oleh Guido sebagai nama bahasa ciptaannya karena kecintaan Guido pada acara televisi Monty *Python's Flying Circus*. Oleh karena itu seringkali ungkapan-ungkapan khas dari acara tersebut seringkali muncul dalam korespondensi antar pengguna *Python*.

### **3.1.2.3 Aplikasi Penggunaan *Python***

*Python* digunakan di berbagai bidang pengembangan. Berikut beberapa aplikasi penggunaan *Python* yang paling populer,

#### **1. Website dan *internet***

Bahasa pemrograman *Python* dapat digunakan sebagai server side yang diintegrasikan dengan berbagai internet

protokol misalnya HTML, JSON, Email Processing, FTP, dan IMAP. Selain itu, *Python* juga mempunyai library untuk pengembangan internet.

## 2. Penelitian ilmiah dan Numerik

*Python* dapat digunakan untuk melakukan riset ilmiah untuk mempermudah perhitungan numerik. Misalnya penerapan algoritma KNN, Naive Bayes, Decision Tree, dan lain-lain.

## 3. Data Science dan Big Data

*Python* memungkinkan untuk melakukan analisis data dari database big data.

## 4. Media pembelajaran pemrograman

*Python* dapat digunakan sebagai media pembelajaran di universitas. *Python* sangat mudah dan hemat untuk dipelajari sebagai Object Oriented Programming dibandingkan bahasa lainnya seperti MATLAB, C++, dan C#.

## 5. Graphical User Interface (GUI)

*Python* dapat digunakan untuk membangun interface sebuah aplikasi. Tersedia library untuk membuat GUI menggunakan *Python*, misalnya Qt, win32extension, dan GTK+.

## 6. Pengembangan Software

*Python* menyediakan dukungan struktur kode untuk mempermudah pengembangan software.

## 7. Aplikasi bisnis

*Python* juga dapat digunakan untuk membuat sistem informasi baik untuk bisnis dan instansi.

## BAB 4

---

### INSTALASI TOOLS YANG DIGUNAKAN

---

#### 4.1 Tools Yang Digunakan

##### 4.1.1 Instalasi Anaconda

Berikut adalah langkah-langkah untuk melakukan instalasi Anaconda:

1. Langkah pertama yaitu melakukan download file .exe pada link berikut ini :"<https://repo.continuum.io/archive/>"

##### Anaconda installer archive [thinkstudioo.blogspot.co.id](http://thinkstudioo.blogspot.co.id)

Filename	Size	Last Modified	MD5
Anaconda2-5.1.0-Linux-ppc64le.sh	267.3M	2018-02-15 09:08:49	e894dcc547a1c7d67deb04f6bba7223a
Anaconda2-5.1.0-Linux-x86.sh	431.3M	2018-02-15 09:08:51	e26fb9d3e53049f6e32212270af6b987
Anaconda2-5.1.0-Linux-x86_64.sh	533.0M	2018-02-15 09:08:50	5b1b5784cae93cf696e11e66983d8756
Anaconda2-5.1.0-MacOSX-x86_64.pkg	588.0M	2018-02-15 09:08:52	4f9c197dfe6d3dc7e50a8611b4d3cfa2
Anaconda2-5.1.0-MacOSX-x86_64.sh	505.9M	2018-02-15 09:08:53	e9845ccf67542523c5be09552311666e
Anaconda2-5.1.0-Windows-x86.exe	419.8M	2018-02-15 09:08:55	a09347a53e04a15ee965300c2b95dfde
Anaconda2-5.1.0-Windows-x86_64.exe	522.6M	2018-02-15 09:08:54	b16d6d6858fc7decf671ac71e6d7cfdb
Anaconda3-5.1.0-Linux-ppc64le.sh	285.7M	2018-02-15 09:08:56	47b5b2b17b7dbac0d4d0f0a4653f5b1c
Anaconda3-5.1.0-Linux-x86.sh	449.7M	2018-02-15 09:08:58	793a94ee85baf64d0ebb67a0c49af4d7
Anaconda3-5.1.0-Linux-x86_64.sh	551.2M	2018-02-15 09:08:57	966406059cf7ed89cc82eb475ba506e5
Anaconda3-5.1.0-MacOSX-x86_64.pkg	594.7M	2018-02-15 09:09:06	6ed496221b843d1b5fe8463d3136b649
Anaconda3-5.1.0-MacOSX-x86_64.sh	511.3M	2018-02-15 09:10:24	047e12523fd287149ecd80c803598429
Anaconda3-5.1.0-Windows-x86.exe	435.5M	2018-02-15 09:10:28	7a2291ab99178a4cdec530861494531f
Anaconda3-5.1.0-Windows-x86_64.exe	537.1M	2018-02-15 09:10:26	83a8b1edcb21fa0ac481b23f65b604c6
Anaconda2-5.0.1-Linux-x86.sh	413.2M	2017-10-24 12:13:07	ae155b192027e23189d723a897782fa3

Gambar 4. 1 Download File Anaconda

Pada gambar 4.1 diatas merupakan tampilan pada saat kita akan mendownload file *anaconda* dari google.

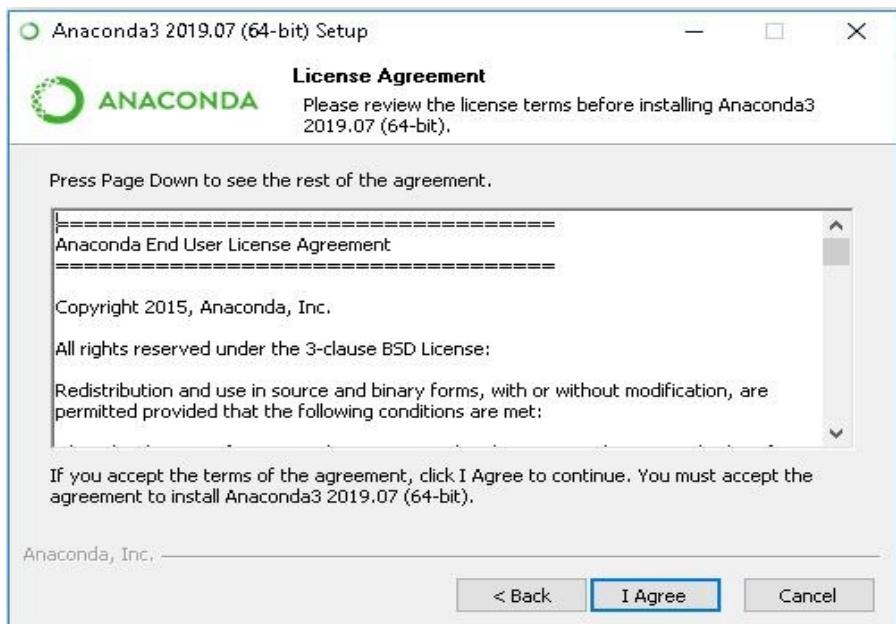
2. Setelah di download selanjutnya kita install dengan mengklik 2 kali (*double click*) pada file installer *Anaconda* sehingga muncul pop up seperti ini.



Gambar 4. 2 Tahap Awal Install Anaconda

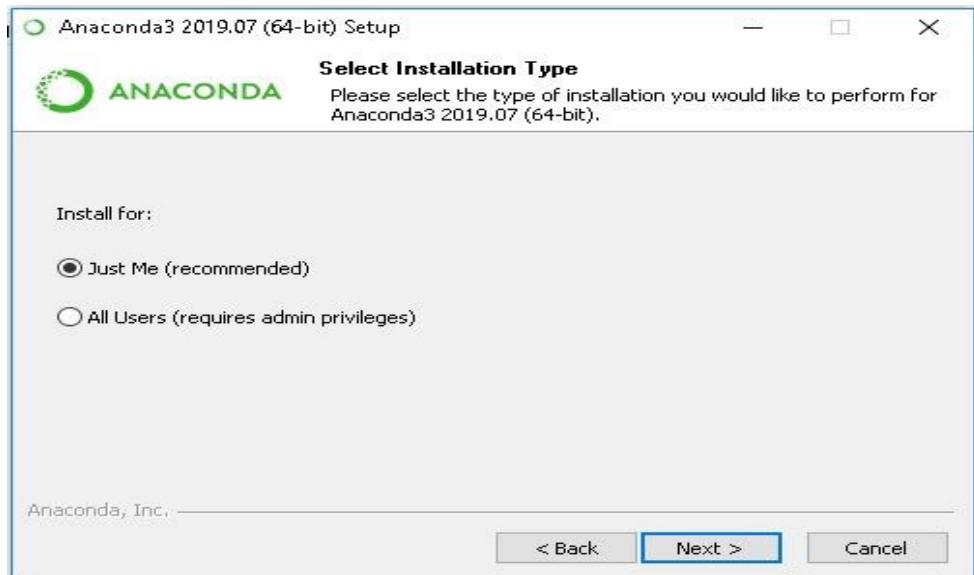
Pada gambar 4.2 diatas merupakan tampilan awal pada saat kita akan melakukan proses instalasi pada *anaconda*, dimana file nya sudah kita download sebelumnya.

3. Tekan Tombol Next untuk ke tahap selanjutnya. Anda akan masuk pada License Agreement. Pilih Tombol I Agree untuk melanjutkan.



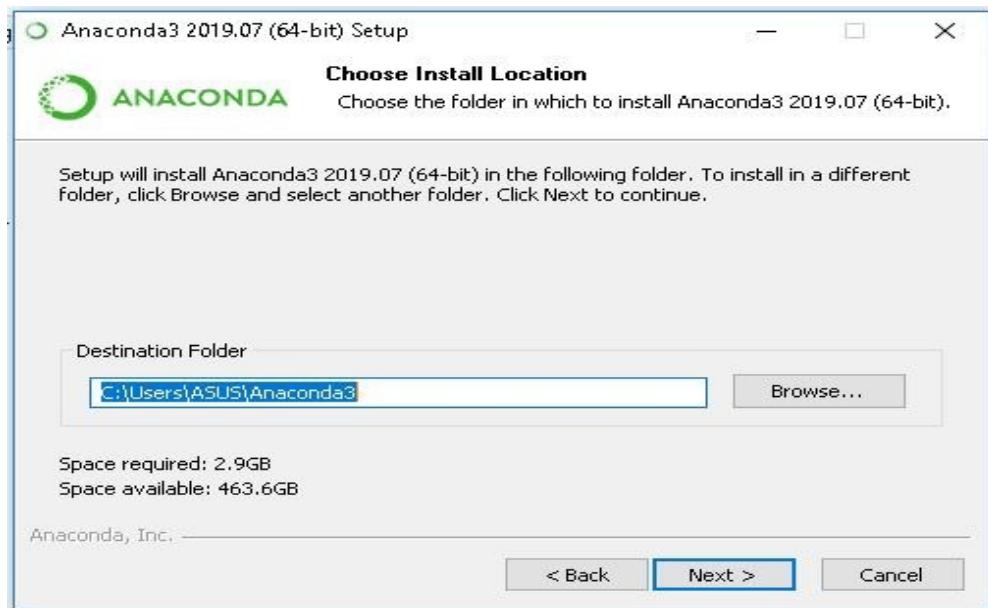
Gambar 4. 3 License Agreement

4. Kemudian pilih Just Me yang di rekomendasi => Klik next.



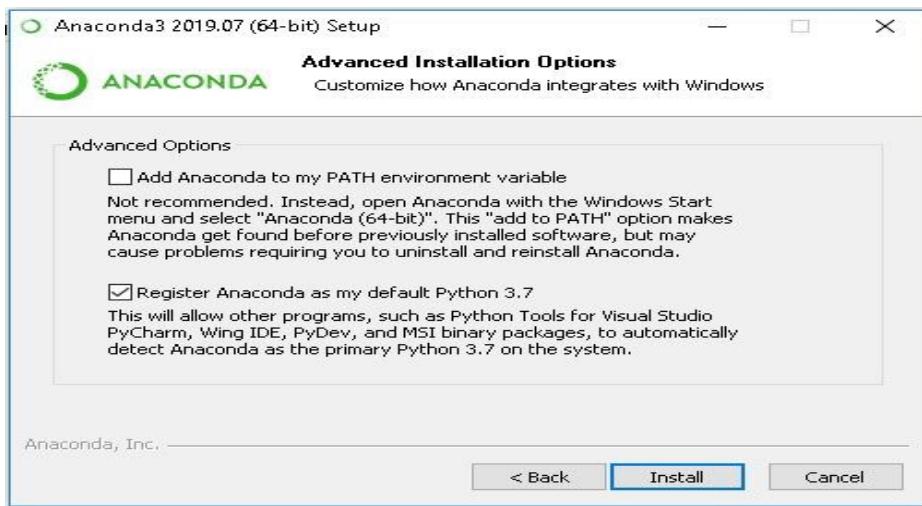
Gambar 4. 4 Select Installation Type

5. Kemudian pilih lokasi penyimpanan aplikasi. Klik browser kemudian arahkan sesuai keinginan kita, klik next.



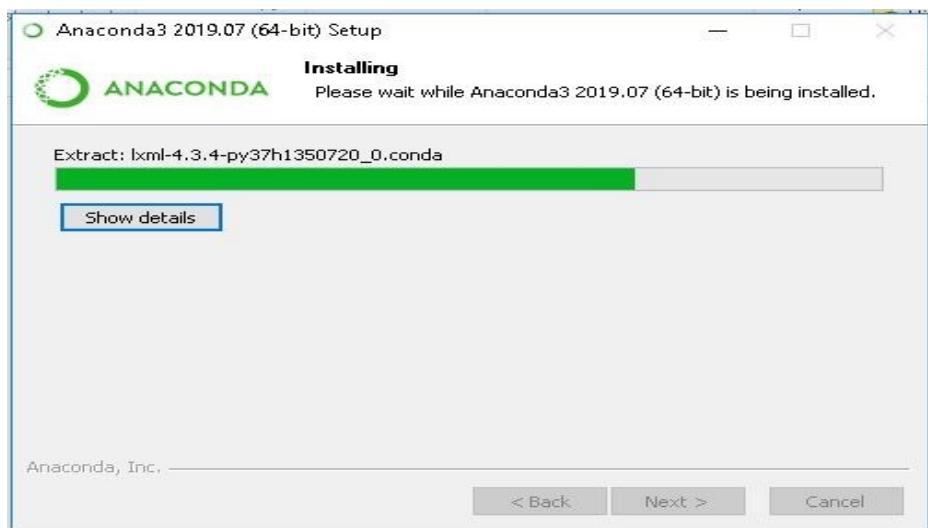
Gambar 4. 5 Choose Install Location

6. Kemudian dipilih add anaconda to PATH atau tidak. Pilih apakah akan mendaftarkan *Anaconda* sebagai default *Python* 3.6?. Kecuali kalian berencana menginstal dan menjalankan beberapa versi *Anaconda*, atau beberapa versi *Python*, biarkan default dan biarkan kotak ini dicentang. kemudian klik next.



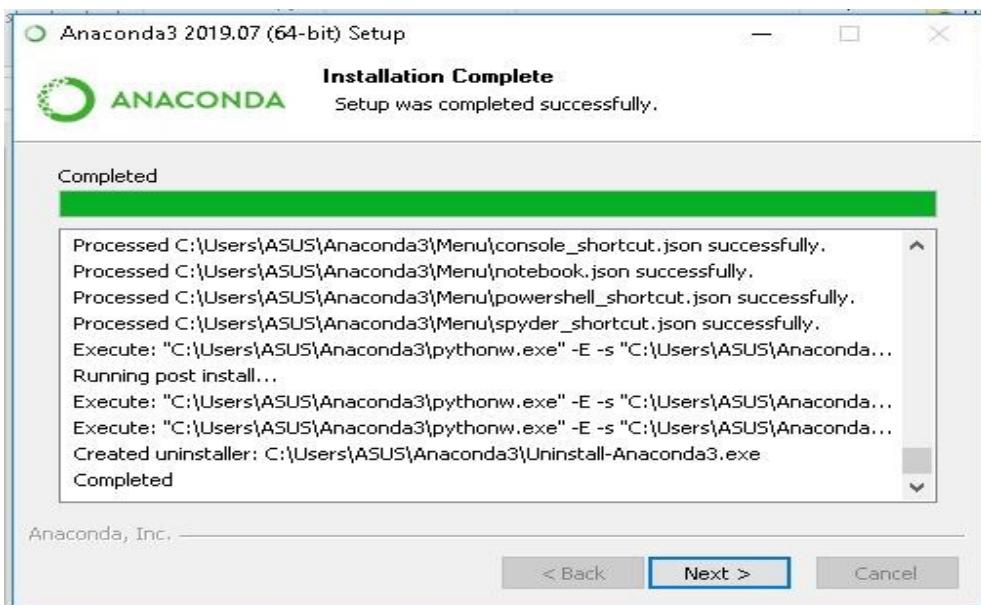
Gambar 4. 6 Advanced Installation Options

7. Klik tombol Install. Jika Kalian ingin melihat packages Anaconda yang sedang dipasang, klik Show Details.



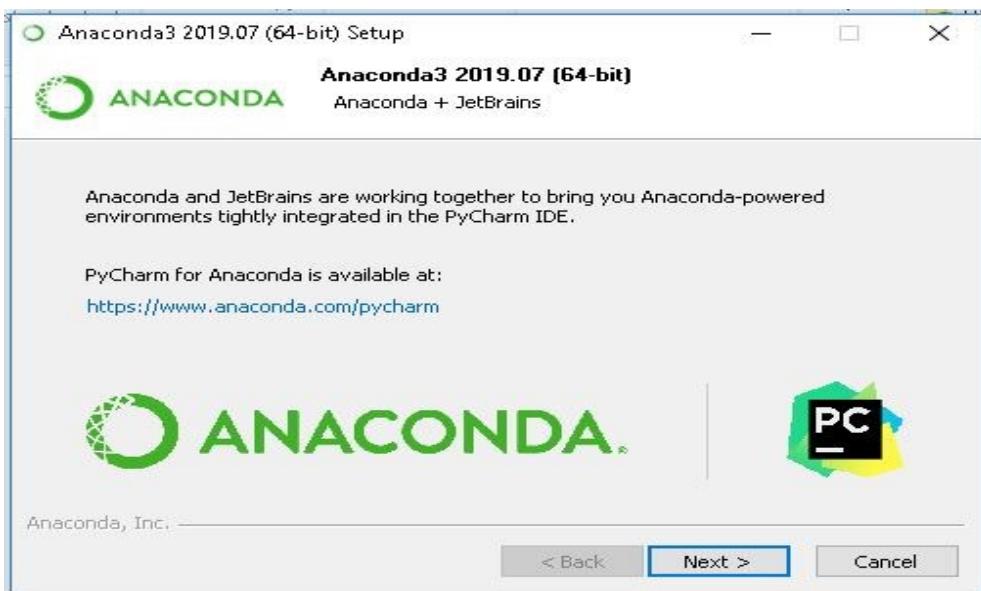
Gambar 4. 7 Proses Installing

8. Proses instalasi sedang berlangsung. Tunggu beberapa menit dan ketika complete tekan tombol Next.



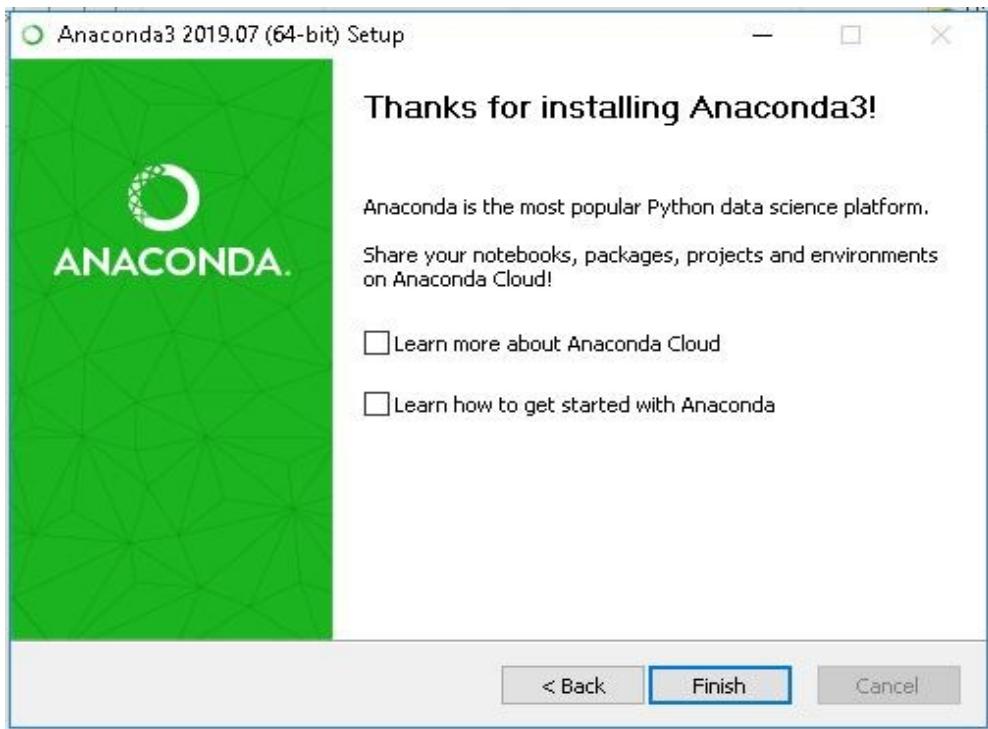
Gambar 4. 8 Installation Complete

9. Muncul halaman informasi tambahan untuk rekomendasi IDE menggunakan pyCharm. Tekan Next saja



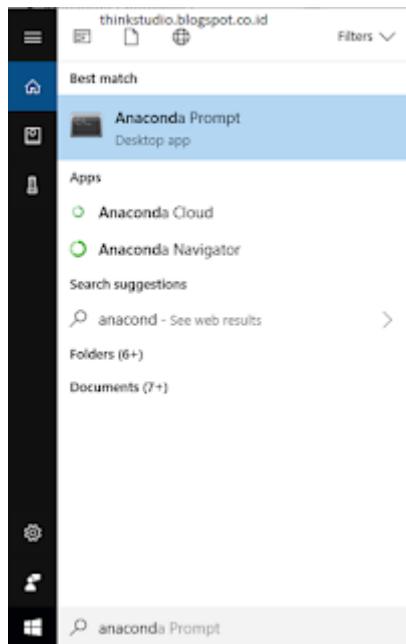
Gambar 4. 9 Anaconda3 2019.07(64-bit)

10. Halaman instalasi terakhir. Hilangkan centang dan tekan tombol Finish.



Gambar 4. 10 Proses Installation Selesai

11. Setelah instalasi kalian selesai, verifikasi dengan membuka *Anaconda Navigator*, program yang disertakan dengan *Anaconda. Search di windows kalian anaconda navigator*. Jika *Navigator* terbuka, Kalian telah berhasil menginstal *Anaconda*. Jika tidak, periksa bahwa Kalian telah menyelesaikan setiap langkah di atas.



Gambar 4. 11 Proses Verifikasi Anaconda

12. Untuk mengecek versi *anaconda* dan *Python* yang terinstall buka command prompt melalui start (seperti memanggil *Anaconda Navigator*) dan ketik “cmd”. Untuk mengecek versi *anaconda* ketik conda -V. untuk mengecek versi *Python* ketik *Python -V*

```
■ Anaconda Prompt (Anaconda3)

(base) C:\Users\ASUS>conda -V
conda 4.7.10

(base) C:\Users\ASUS>python -V
Python 3.7.3

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

*Gambar 4. 12 Cek versi Python dan Versi Anaconda*

#### **4.1.2 Instalasi LabelImg**

Berikut adalah langkah-langkah untuk melakukan instalasi LabelImg :

1. Untuk melakukan instalasi LabelImg maka pertama-tama melakukan download file .exe pada link berikut ini :  
<https://tzutalin.github.io/labelImg/> Sesuaikan dengan system operasi yang anda gunakan.
2. Anaconda Prompt dan buka direktori labelImg

```
Conda install pyqt = 5
pyrcc5 -o libs / resources.py resources.qrc
python labelImg.py
python labelImg.py [IMAGE_PATH] [FILE KELAS PRE-DEFINISI]
```

*Gambar 4. 13 Proses Instalasi LabelImg*

#### **3. Pemakaian**

- Langkah (PascalVOC)
  - Bangun dan luncurkan dengan menggunakan instruksi di atas.
  - Klik 'Ubah folder anotasi tersimpan standar' di Menu / File
  - Klik 'Buka Dir'
  - Klik 'Buat Kotak Panjang'
  - Klik dan lepaskan mouse kiri untuk memilih wilayah untuk memberi anotasi pada kotak segi empat
  - Anda dapat menggunakan mouse kanan untuk menarik kotak persegi panjang untuk menyalin atau memindahkannya

Anotasi akan disimpan ke folder yang Anda tentukan. Anda bisa merujuk ke hotkey di bawah ini untuk mempercepat alur kerja Anda.

- Langkah (YOLO)
  - Dalam data/predefined\_classes.txt menentukan daftar kelas yang akan digunakan untuk pelatihan Anda.
  - Bangun dan luncurkan dengan menggunakan instruksi di atas.
  - Tepat di bawah tombol "Simpan" di bilah alat, klik tombol "PascalVOC" untuk beralih ke format YOLO.
  - Anda dapat menggunakan Open / OpenDIR untuk memproses gambar tunggal atau ganda. Setelah selesai dengan satu gambar, klik simpan.

File txt dalam format YOLO akan disimpan di folder yang sama dengan gambar Anda dengan nama yang sama. File bernama "classes.txt" disimpan ke folder itu juga. "classes.txt" mendefinisikan daftar nama kelas yang dirujuk oleh label YOLO Anda.

catatan:

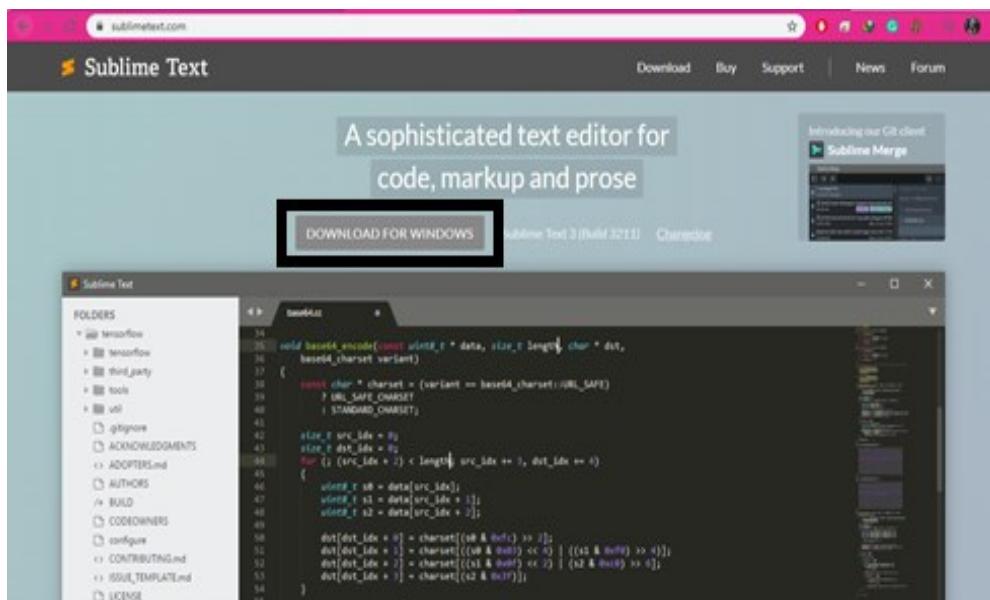
- Daftar label Anda tidak akan berubah di tengah pemrosesan daftar gambar. Saat Anda menyimpan gambar, classes.txt juga akan diperbarui, sedangkan anotasi sebelumnya tidak akan diperbarui.
- Anda seharusnya tidak menggunakan fungsi "kelas default" saat menyimpan ke format YOLO, itu tidak akan dirujuk.

- Saat menyimpan sebagai format YOLO, bendera "sulit" dibuang.

#### 4.1.3 Instalasi Sublime

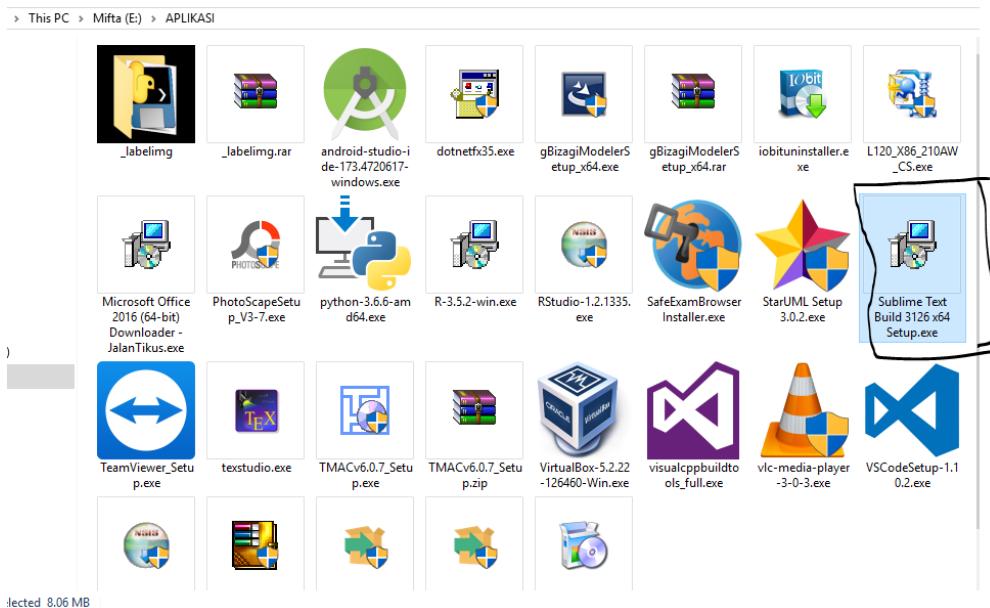
Berikut adalah langkah-langkah untuk melakukan instalasi Sublime :

1. Tahapan yang pertama untuk melakukan instalasi maka download dahulu file .exe pada link berikut :  
<https://www.sublimetext.com/>.



*Gambar 4. 14 Download file .exe*

2. Setelah download selesai maka double klik file .exe pada hasil download tadi.



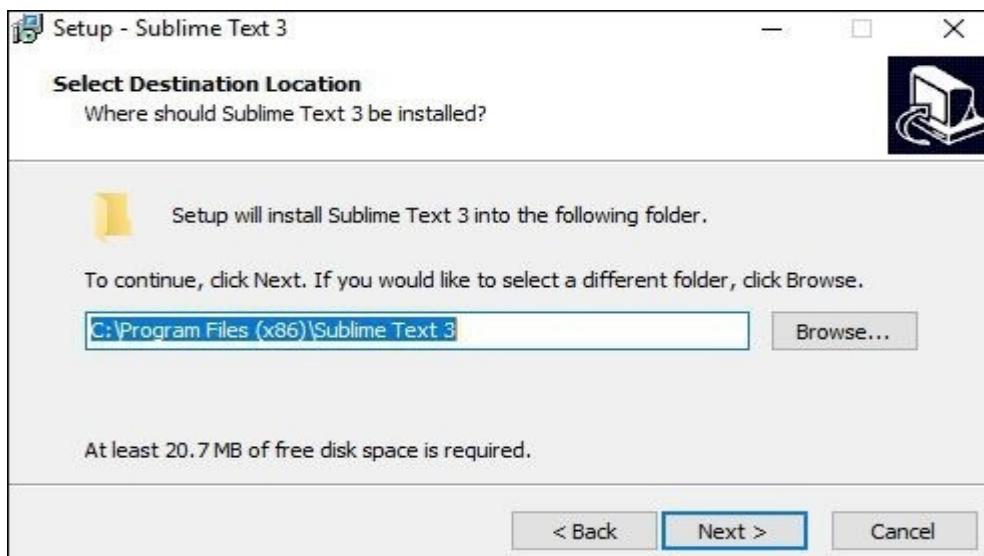
Gambar 4. 15 Double Klik File .exe

3. Jika sudah double klik file .exe maka akan muncul tampilan berikut kemudian klik next :



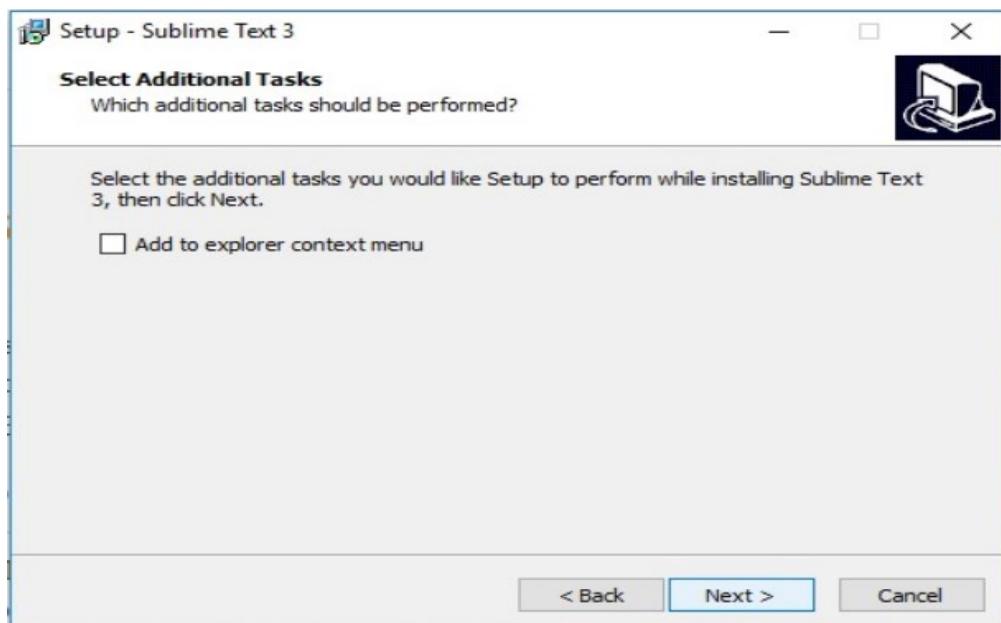
Gambar 4. 16 Proses Instalasi Sublime

4. Sekarang, pilih lokasi tujuan untuk menginstal Sublime Text3 dan klik Next .



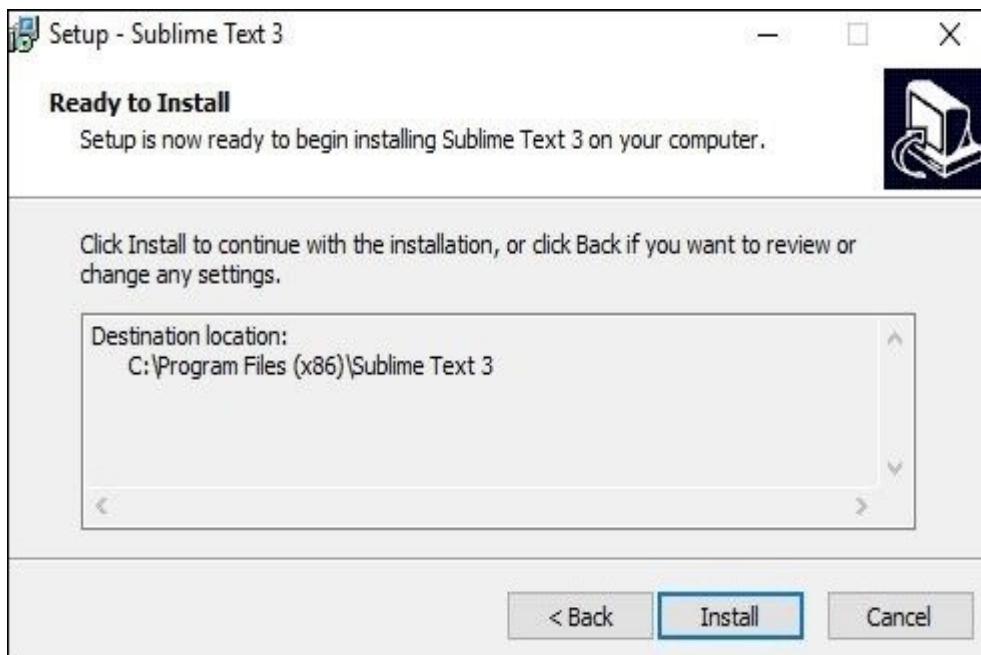
Gambar 4. 17 Select Destination Location

5. Selanjutnya klik next.



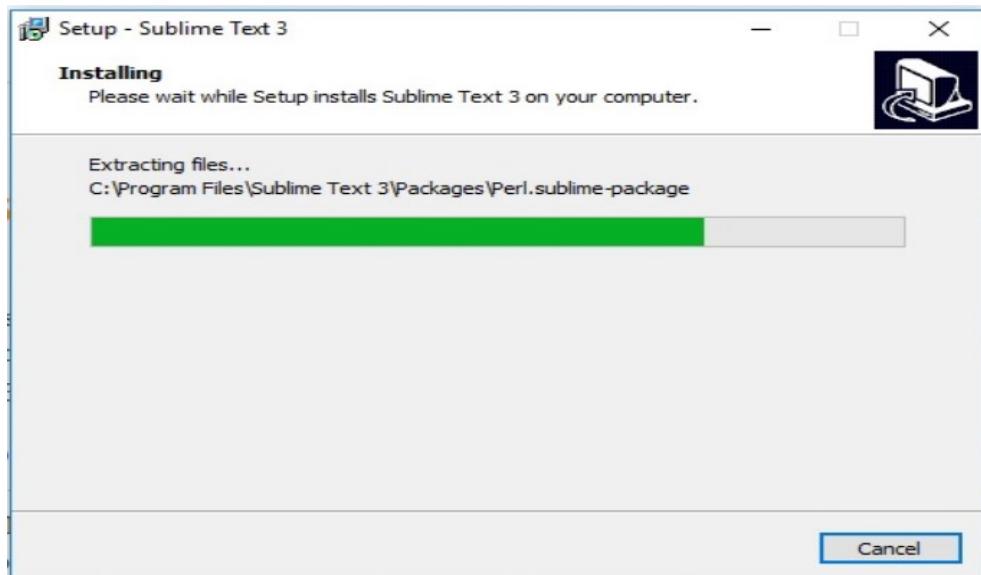
Gambar 4. 18 Select Additional Tasks

6. Verifikasi folder tujuan dan klik Instal .



Gambar 4. 19 Proses Instalasi Sublime

7. Tunggu sampai proses instalasi selesai, Setelah proses ini selesai maka sublime telah bisa digunakan



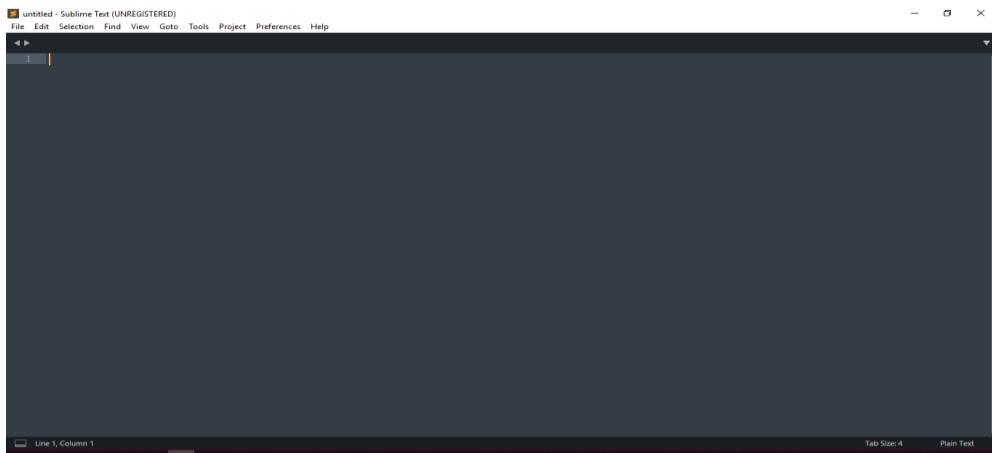
Gambar 4. 20 Proses Instalasi Sublime

8. Sekarang, klik Selesai untuk menyelesaikan instalasi.



*Gambar 4. 21 Proses Install Selesai*

9. Setelah instalasi berhasil, editor Anda akan muncul seperti yang ditunjukkan di bawah ini

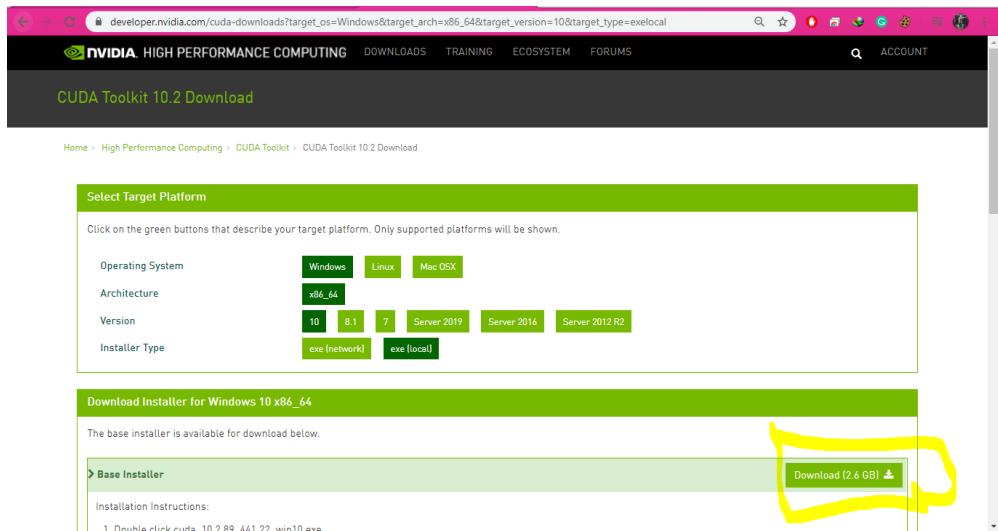


*Gambar 4. 22 Tampilan Sublime*

#### **4.1.4 Instalasi CUDA**

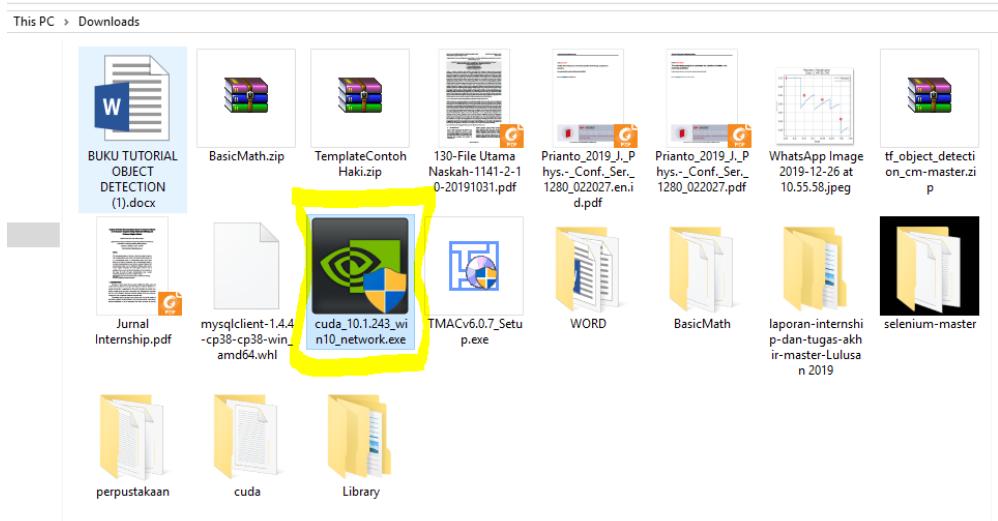
Berikut adalah langkah-langkah untuk melakukan instalasi CUDA :

1. Langkah pertama yaitu download terlebih dahulu file .exe pada link berikut : <https://developer.nvidia.com/cuda-downloads>  
Nanti kita akan diminta memilih versi yang akan kita download, pilih saja sesuai OS dan arsitektur komputer.



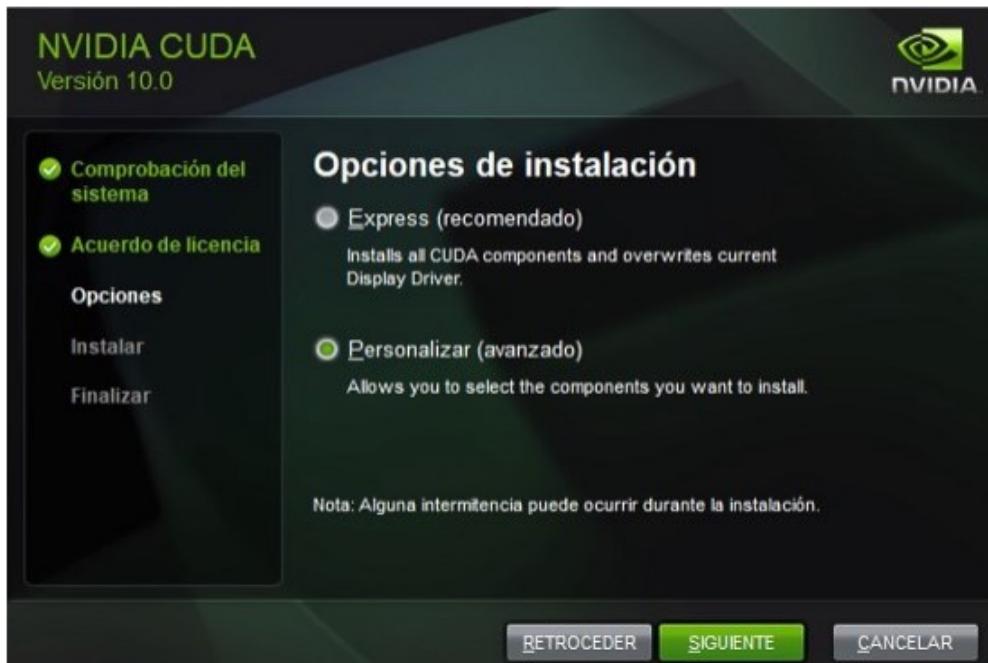
Gambar 4. 23 Download File .exe CUDA

2. Kemudian Proses instalasi dilakukan dengan double cara klik file yang telah di download tadi.



Gambar 4. 24 File Yang Telah di Download

3. Proses Instalasi CUDA hanya dengan klik next sampai prosesnya selesai, proses ini memakan waktu yang cukup lama. Tunggu hingga proses selesai.



Gambar 4. 25 Proses Instalasi CUDA

4. Setelah CUDA berhasil diinstall, kita perlu memastikan bahwa CUDA sudah ada di environment variable. Jika belum ada maka kita perlu menambahkannya.

System variables	
Variable	Value
ComSpec	C:\windows\system32\cmd.exe
CUDA_PATH_V10_1	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1
DriverData	C:\Windows\System32\Drivers\DriverData
img_path	C:\Users\Mifta\Documents\rotate_3d-master
NUMBER_OF_PROCESSORS	4
NVCUDASAMPLES_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.1
NVCUDASAMPLES10_1_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.1

New... Edit... Delete

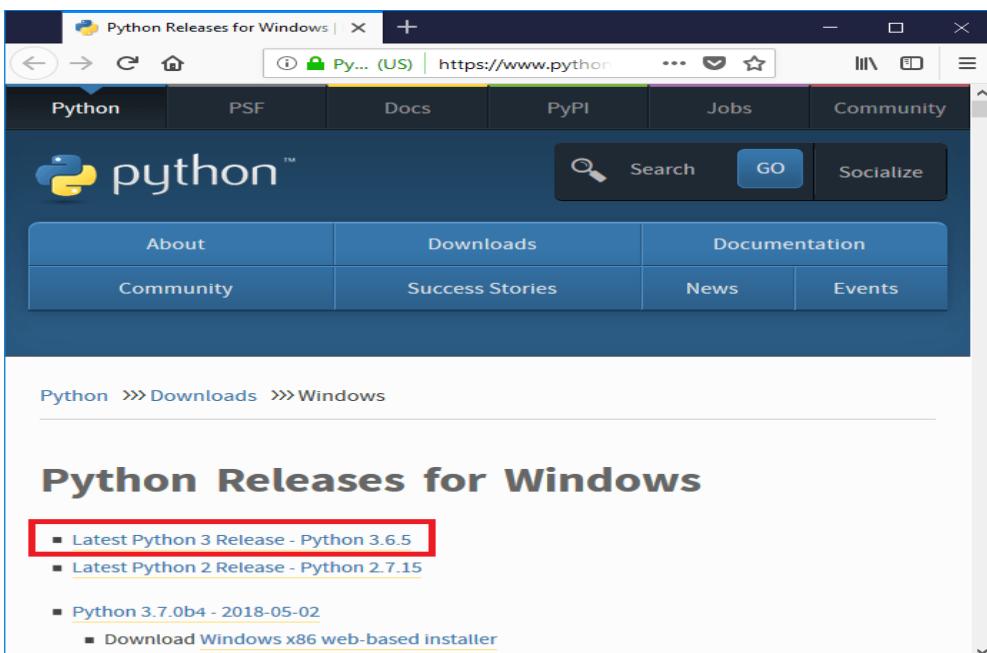
*Gambar 4. 26 Menambahkan CUDA pada Environment Variable*

## **4.2 Bahasa Pemrograman Yang Digunakan**

### **4.2.1 Instalasi *Python***

Berikut adalah langkah-langkah untuk melakukan instalasi Bahasa pemograman *Python*:

1. Unduh instalasi *Python* yang berbentuk executable di “<https://www.Python.org/downloads/windows/>”. Pilih versi terbaru yang paling stabil yaitu link paling atas.



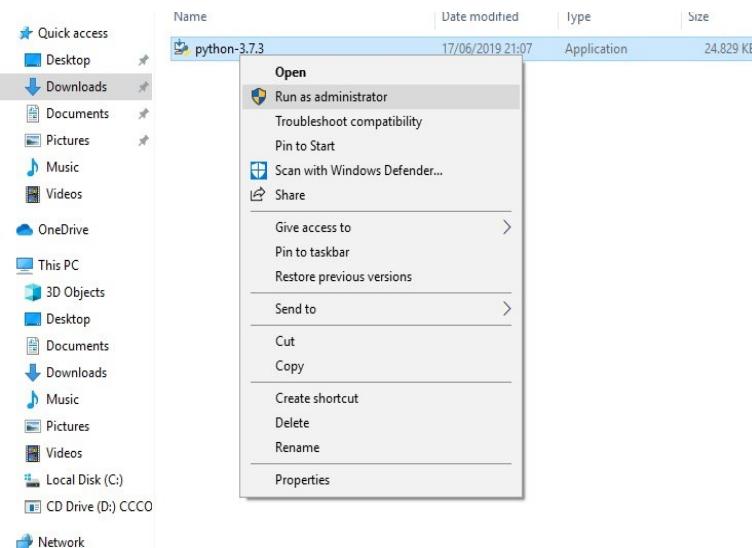
*Gambar 4. 27 Download File Python*

2. Kemudian sesuaikan versi arsitektur sistem operasi Windows yang digunakan. Pilih x86-64 untuk arsitektur sistem operasi 64bit atau pilih x86 untuk arsitektur sistem operasi 32bit.

Gzipped source tarball	Source release		ab25d24b1f8cc4990ade979f6dc37883	22994617	SIG
XZ compressed source tarball	Source release		9f49654a4d6f733ff3284ab9d227e9fd	17049912	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	bf319337bc68b52fc7d227da5b6f2f6	28093627	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	37d891988b6aeedd7f03a70171a8420d	26987706	SIG
Windows help file	Windows		be70202d483c0b7291a666ec66539784	8065193	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	04cc4f6f6a14ba74f6ae1a8b685ec471	7190516	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	9e96c934f5d16399f860812b4ac7002b	31776112	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	640736a3894022d30f7babff77391d6b	1320112	SIG
Windows x86 embeddable zip file	Windows		b0b099a4fa479fb37880c15f2b2f4f34	6429369	SIG
Windows x86 executable installer	Windows		2bb6ad2ecca6088171ef923bca483f02	30735232	SIG
Windows x86 web-based installer	Windows		596667cb91a9fb20e6f4f153f3a213a5	1294096	SIG

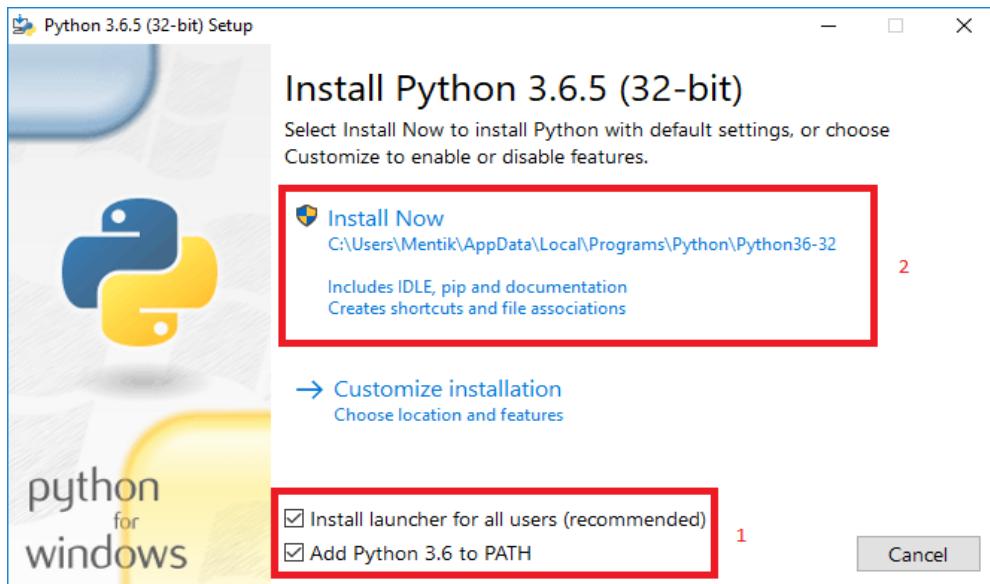
*Gambar 4. 28 Sesuaikan Versi Arsitektur Sistem Operasi Windows Yang Digunakan*

3. Setelah download selesai, kita akan mendapatkan file *Python-3.4.2.msi*. File *Python-3.4.2.msi* adalah file instalator *Python*. File ini akan melakukan instalasi ke sistem windows. Klik ganda untuk mengeksekusinya atau bisa juga dengan pilih run as administrator.



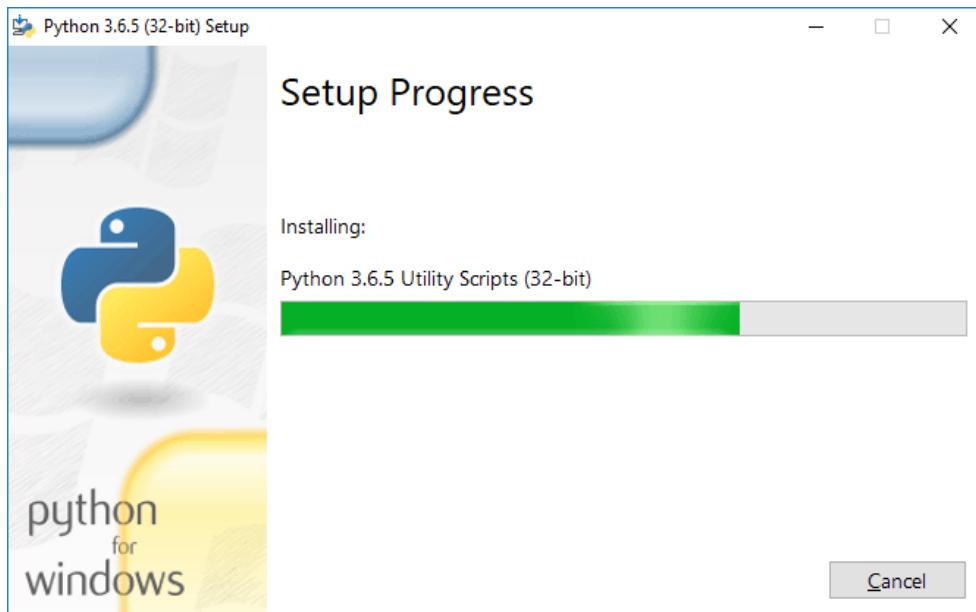
*Gambar 4. 29 Instalasi Python Pada Windows*

4. Buka file installer yang telah di download. Centang Install launcher for all user untuk mengaktifkan *Python* pada semua user Windows dan centang *Python* 3.6 to PATH untuk menambah path command *Python*. Kemudian klik Install Now. Klik Yes saat muncul notifikasi User Account Control.



Gambar 4. 30 Jendela pertama proses instalasi Python

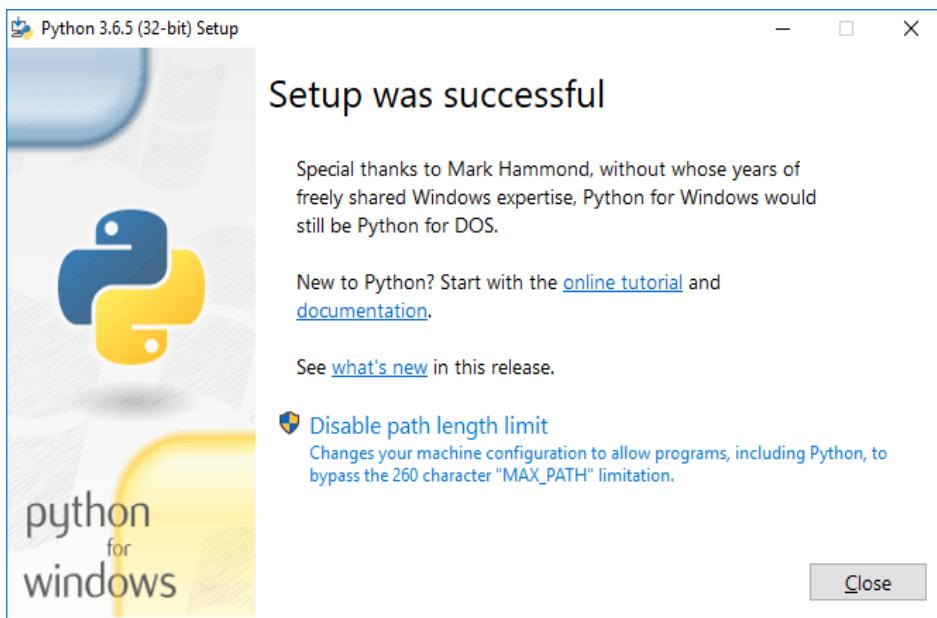
5. Centang pilihan “**Add Python 3.7 to PATH**” di bagian bawah jendela (1). Ini berguna agar *Python* interpreter bisa diakses dari lokasi mana saja. Setelah itu klik tombol “**Install Now**” (2). Proses instalasi akan berlangsung beberapa saat.



Gambar 4. 31 Proses instalasi Python sedang berlangsung

#### 6. Opsi “Disable path length limit”

Di sini, anda dapat mengaktifkan atau menonaktifkan batasan lintasan direktori *Python*. Untuk pengguna Linux batasan ini tidak mempunyai pengaruh yang besar, karena Linux menyimpan *Python* pada direktori yang pendek. Berbeda halnya dengan Windows, yang menyimpan direktori lintasan agak jauh dari direktori utama partisi. Disarankan untuk menonaktifkan batasan ini dengan klik “Disable path length limit”.



Gambar 4. 32 Proses instalasi Python sudah selesai

7. Install *Python* berhasil. Klik Close untuk menutup installer *Python*.

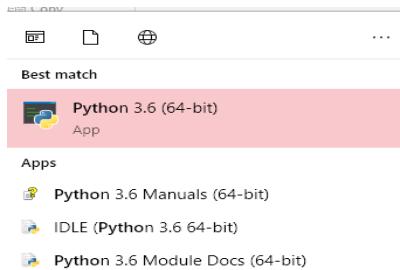


Gambar 4. 33 Proses instalasi Python sudah selesai

Selanjutnya periksa di bagian Menu Windows, lalu cari *Python 3.6*.

Anda akan mendapati 4 file dalam menu ini:

- *IDLE (Python 3.6)*  
*IDLE* adalah kependekan dari *Python's Integrated Development and Learning Environment* yang merupakan IDE standar *Python*.
- *Python 3.6*  
*Python 3.6* untuk membuka *Python* melalui command prompt.
- *Python 3.6 Manual*  
*Panduan manual Python*.
- *Python 3.6 Module Docs*  
*Panduan mengenai module pada Python*.



Gambar 4. 34 File Python Yang Dapat Digunakan

8. Untuk menandakan apakah *Python* sudah ter konfigurasi, caranya buka CMD lalu ketikkan perintah “*Python --version*”. Jika sudah ter konfigurasi maka akan muncul seperti gambar berikut ini:

```
Command Prompt
Microsoft Windows [Version 10.0.17134.915]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Mifta>python --version
Python 3.6.6

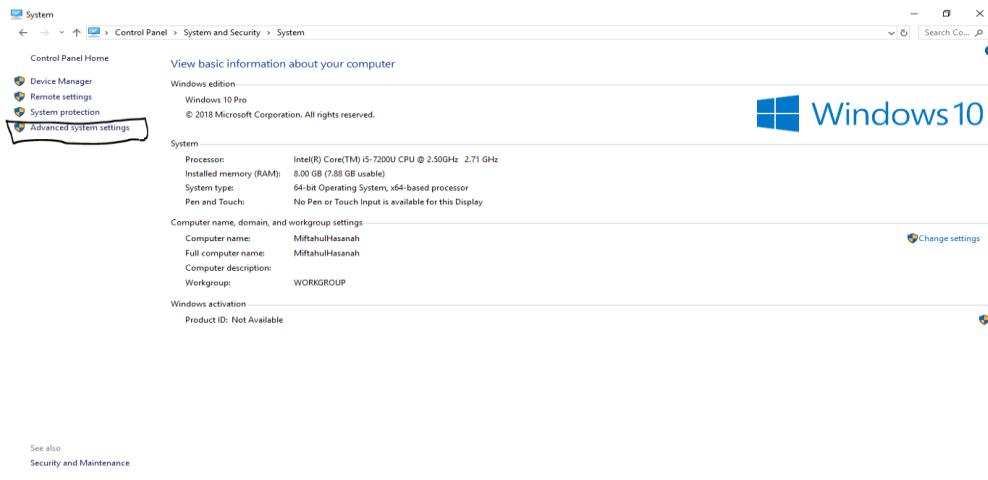
C:\Users\Mifta>
```

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:  
Microsoft Windows [Version 10.0.17134.915]  
(c) 2018 Microsoft Corporation. All rights reserved.  
  
C:\Users\Mifta>python --version  
Python 3.6.6  
  
C:\Users\Mifta>

Gambar 4. 35 Versi Python Yang Digunakan

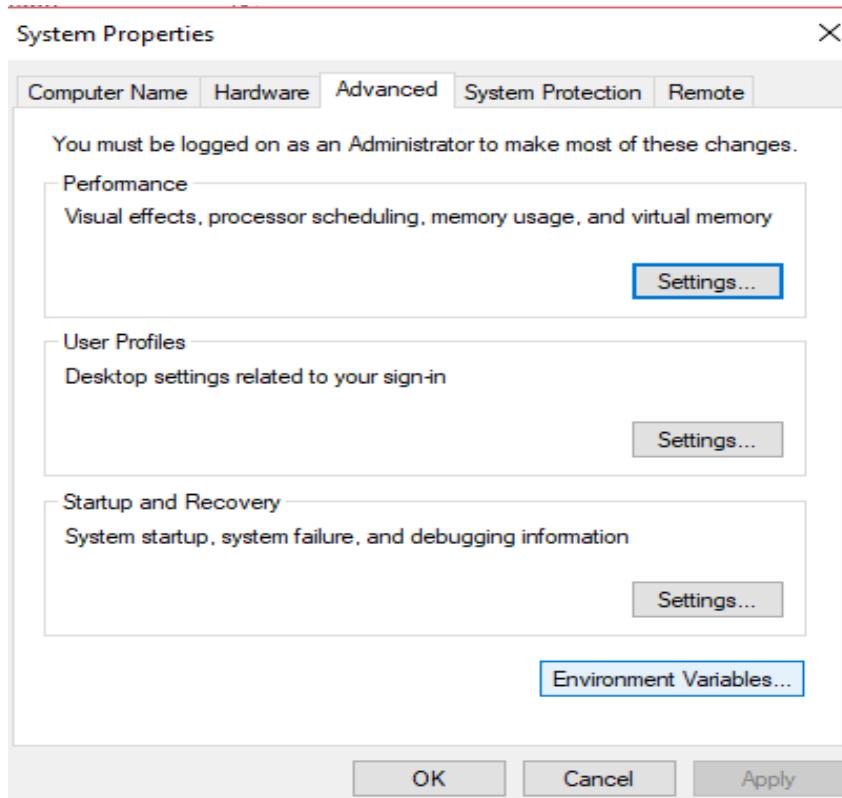
Jika tampil versi *Python* yang digunakan seperti pada gambar diatas, maka instalasi *Python* telah berhasil. Namun jika belum, di cek terlebih dahulu pada bagian *Environtment Variables* sistem operasi windows.

- Buka *Windows Explorer*, lalu pada bagian *This PC* lakukan klik kanan lalu pilih *Properties*.



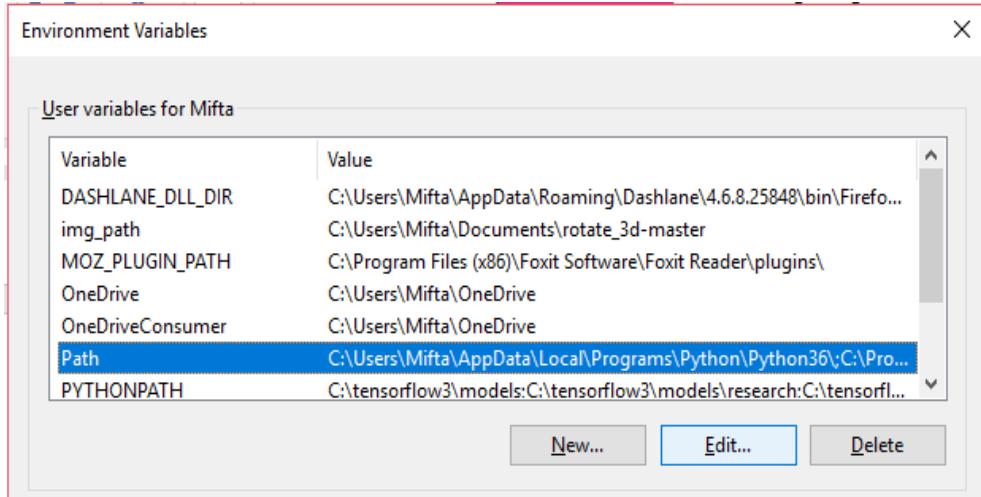
Gambar 4. 36 System Pada Pc

- Selanjutnya pada menu sebelah kiri pilih menu *Advanced System Settings*. maka akan tampil jendela *System Properties*.



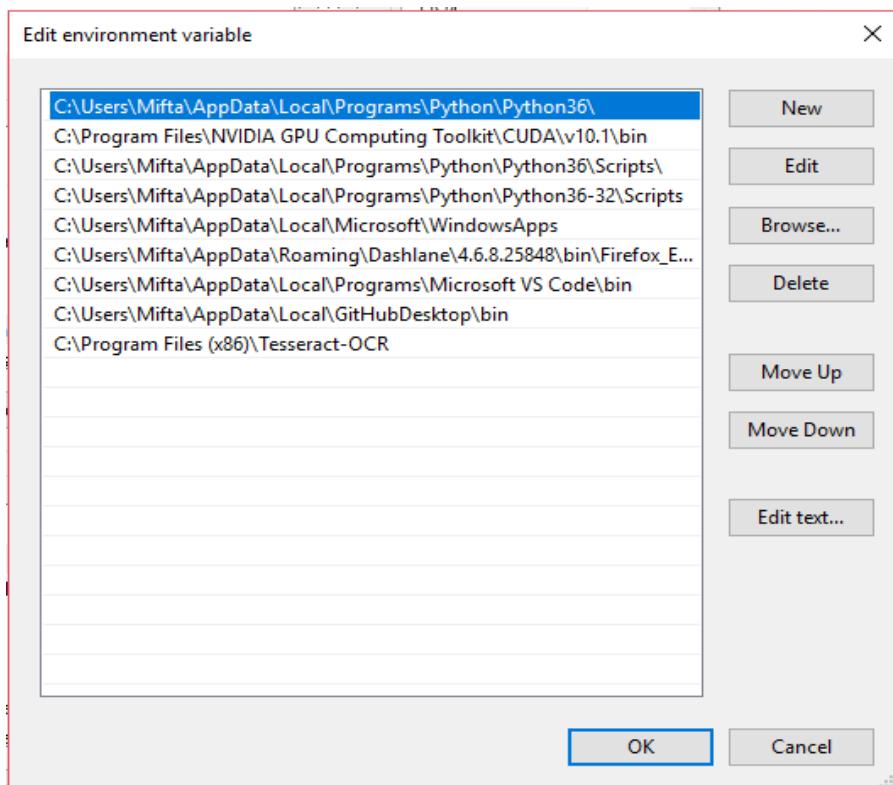
Gambar 4. 37 Konfigurasi Python pada Windows

- Pilih di bagian menu Advanced dan pilih tombol *Environment Variables for ...* di sini kita akan melakukan konfigurasi (*PATH environment*). Maka akan dialihkan ke:



Gambar 4. 38 Path Environment Variabel

- Untuk menambahkan *Python* klik edit di path, kemudian tambahkan direktori *Python* nya. Contohnya seperti pada gambar berikut ini :



Gambar 4. 39 Path Environment Variabel

## **PENJELASAN, INSTALASI FRAMEWORK & LIBRARY YANG DIGUNAKAN**

---

### **5.1 Membuat Environment Pada Anacoda**

Untuk melakukan instalasi framework dan library yang dibutuhkan maka kita harus membuat environment kemudian menginstal library dan framework tersebut ke dalam environment yang sudah dibuat. Hal ini bertujuan untuk memudahkan pengaturan environment sesuai kebutuhan program yang sedang dibuat. Apabila suatu saat nanti kita ingin menggunakan versi yang berbeda dari Tensor flow dan Keras yang telah terinstal tersebut maka hal yang perlu dilakukan hanyalah membuat environment yang baru sehingga tidak akan berdampak kepada environment sebelumnya yang telah diatur spesifik sesuai kebutuhan proyek yang lama. Berikut langkah-langkah untuk membuat environment pada anaconda:

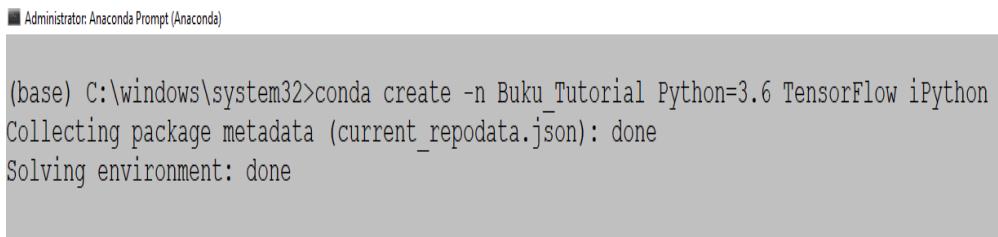
1. Buka *anaconda* command prompt, kemudian akan tampil *anaconda* prompt sebagai berikut :



```
Administrator: Anaconda Prompt (Anaconda3)
(base) C:\windows\system32>
```

*Gambar 5. 1 Tampilan Anaconda Prompt*

2. Untuk membuat environment baru dalam *Anaconda* Anda dapat menggunakan perintah sebagai berikut :  
“conda create -n Buku Tutorial Python=3.6 TensorFlow iPython”



```
Administrator: Anaconda Prompt (Anaconda)
(base) C:\windows\system32>conda create -n Buku Tutorial Python=3.6 TensorFlow iPython
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

*Gambar 5. 2 Membuat environment Dengan Nama “Buku\_Tutorial”*

3. Tunggu hingga proses pembuatan environment-nya selesai kemudian jalan kan perintah “y” untuk menginstall package yang disediakan *anaconda*

```
Administrator: Anaconda Prompt (Anaconda)
==> WARNING: A newer version of conda exists. <==
    current version: 4.7.12
    latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda


## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- ipython
- python=3.6
- tensorflow

The following packages will be downloaded:

      package          |       build
-----+-----
absl-py-0.9.0      |   py36_0
astor-0.7.1        |   py_0
backcall-0.1.0     |   py_0
ca-certificates-2019.11.28 | hecc5488_0
certifi-2019.11.28 | py36_0
colorama-0.4.3     |   py_0
decorator-4.4.1    |   py_0
gast-0.3.3         |   py_0
grpcio-1.23.0      | py36h3db2c7e_0
h5py-2.10.0         | nompi_py36h422b98e_101
                    |                                     949 KB  conda-forge
hdf5-1.10.5         | nompi_ha405e13_1104
                    |                                     35.1 MB  conda-forge
ipython-7.11.1      | py36h5cald4c_0
ipython_genutils-0.2.0 |   py_1
jedi-0.16.0         | py36_0
keras-applications-1.0.8 |   py_1
keras-preprocessing-1.1.0 |   py_0
libblas-3.8.0        |   14_mkl
libcblas-3.8.0       |   14_mkl
liblapack-3.8.0      |   14_mkl
libprotobuf-3.11.2    | hlalb453_0
```

Gambar 5. 3 Proses Membuat Environment

*Gambar 5. 4 Proses Membuat Environment*

Gambar 5.4 diatas merupakan proses dalam membuat environment, pada tahap ini kita akan men download dan meng ekstrak package yang disediakan oleh anaconda.

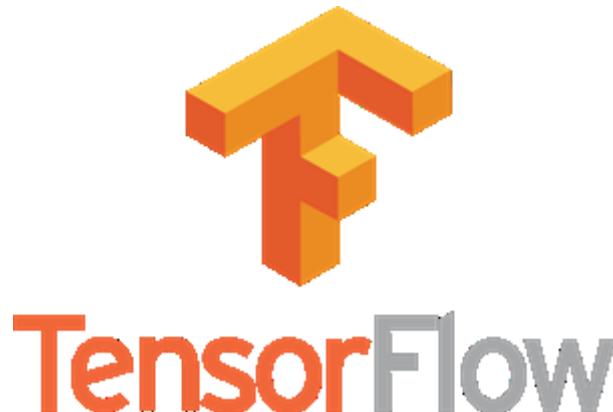
4. Jika pembuatan environment Berhasil maka akan seperti tampilan berikut :

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate Buku_Tutorial
#
# To deactivate an active environment, use
#
#     $ conda deactivate
```

Gambar 5. 5 Proses Instalasi Selesai

## 5.2 Library Tensorflow

### 5.2.1 Pengertian Tensor flow



Gambar 5. 6 Logo Tensorflow

*TensorFlow* merupakan open source framework yang dapat digunakan untuk mengembangkan, melatih, dan menggunakan model deteksi objek. Sistem ini sudah banyak diterapkan pada berbagai produk Google antara lain pencarian image, deteksi wajah dan plat nomor kendaraan pada Google Streetview, Google Assistant, Waymo atau Self Driving Car, dan lain-lain.

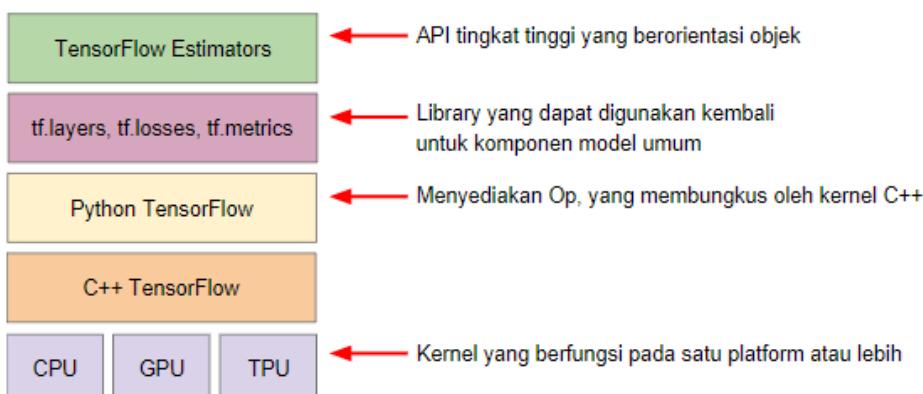
*TensorFlow* bekerja dengan komputasional untuk membuat model machine learning. *TensorFlow* menyediakan berbagai toolkit yang memungkinkan Anda membuat model pada tingkat abstraksi yang Anda suka. Anda dapat menggunakan API dengan tingkat yang lebih rendah untuk membuat model dengan menentukan serangkaian operasi matematis. Sebagai alternatif, Anda dapat menggunakan API dengan tingkat yang lebih tinggi (seperti tf.estimator) untuk menentukan arsitektur yang telah ditetapkan, seperti regresi linier atau jaringan neural.

Framework *TensorFlow* ini digunakan pada proses pembuatan sistem objek deteksi agar memudahkan implementasi Algoritma dan penggunaan Bahasa pemrograman kemudian terdapat GPU untuk mempercepat proses training. *TensorFlow* sebagai kerangka machine learning yang dapat digunakan untuk mengolah banyak data atau ingin mempelajari kecerdasan buatan (artificial intelligence) secara mendalam.

*TensorFlow* adalah pustaka perangkat lunak sumber terbuka untuk perhitungan numerik berkinerja tinggi. Arsitekturnya yang fleksibel memungkinkan penyebaran komputasi dengan mudah di berbagai platform (CPU, GPU, TPU), dan dari desktop ke cluster server hingga perangkat seluler dan tepi.

Awalnya dikembangkan oleh para peneliti dan insinyur dari tim Google Brain dalam organisasi AI Google, ia hadir dengan dukungan kuat untuk pembelajaran mesin dan pembelajaran mendalam dan inti perhitungan numerik yang fleksibel digunakan di banyak domain ilmiah lainnya.

Gambar berikut menunjukkan hierarki toolkit *TensorFlow* saat ini :



*Gambar 5. 7 Hierarki toolkit TensorFlow.*

Tabel berikut berisi ringkasan tujuan dari berbagai lapisan:

*Tabel 5. 1 Hierarki toolkit TensorFlow*

Toolkit	Deskripsi
Estimator ( <i>tf.estimator</i> )	API tingkat tinggi yang berorientasi objek.
<i>tf.layers/tf.losses/tf.metrics</i>	<i>Library</i> untuk komponen model umum.
<i>TensorFlow</i>	API dengan tingkat lebih rendah

*TensorFlow* terdiri dari dua komponen berikut:

- buffer protokol grafik
  - waktu proses yang menjalankan grafik (terdistribusi)
- komponen ini masing-masing bersifat analog terhadap kode Python dan penafsir Python. Sama seperti pemrograman Python diterapkan pada beberapa platform hardware untuk menjalankan kode Python, *TensorFlow* dapat menjalankan grafik pada beberapa platform hardware, termasuk CPU, GPU, dan TPU.

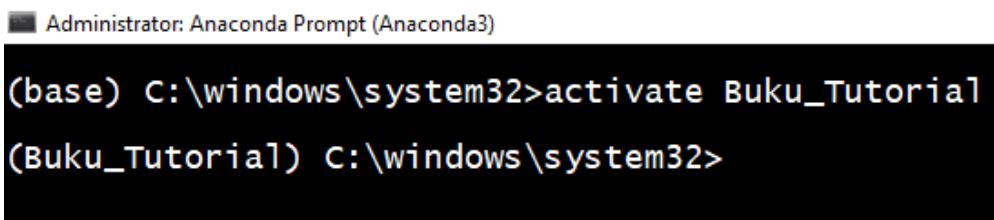
API mana yang harus Anda gunakan? Anda harus menggunakan tingkat abstraksi tertinggi yang dapat memecahkan masalah. Tingkat abstraksi yang lebih tinggi lebih mudah digunakan, tetapi juga (berdasarkan desain) kurang fleksibel. Sebaiknya mulai dengan API tingkat tertinggi terlebih dahulu dan pastikan semuanya berfungsi. Jika Anda memerlukan fleksibilitas tambahan untuk beberapa masalah pemodelan khusus, berpindahlah ke satu tingkat lebih rendah. Perhatikan bahwa setiap tingkat dibuat menggunakan

API di tingkat yang lebih rendah, sehingga menurunkan hierarki seharusnya cukup mudah.

### 5.2.2 Cara Instalasi Tensor flow

Tensorflow terdapat 2 tipe yaitu CPU only dan support GPU, GPU yang support disini hanyalah milik NVIDIA bukan AMD. Berikut adalah langkah untuk install *tensor flow* dengan *anaconda*.

- Buka Anaconda prompt (di sini saya menggunakan Anaconda3)
- Kemudian buat environment pada anaconda. Sebelumnya saya sudah buat environment dengan nama “Buku\_Tutorial”
- Aktifkan environment Buku\_Tutorial yang telah kita buat pada langkah 2 dengan perintah berikut:



```
Administrator: Anaconda Prompt (Anaconda3)
(base) C:\windows\system32>activate Buku_Tutorial
(Buku_Tutorial) C:\windows\system32>
```

Gambar 5. 8 Langkah Untuk Aktifkan Environment Buku\_Tutorial

- Masukkan perintah yang sesuai untuk menginstal *TensorFlow* di dalam conda environment Anda. Untuk menginstal versi *TensorFlow* support GPU, masukkan perintah berikut: (versi *TensorFlow* menyesuaikan dengan kebutuhan machine learning yang akan dibuat). Disini saya menggunakan Tensorflow versi 1.11.0

```
(Buku_Tutorial) C:\windows\system32>conda install tensorflow-gpu=1.11.0
Collecting package metadata (current_repodata.json): done
Solving environment: failed with initial frozen solve. Retrying with flexible solve.
Collecting package metadata (repodata.json): done
Solving environment: done
```

*Gambar 5. 9 Proses Instalasi TensorFlow support GPU*

```
Administrator: Anaconda Prompt (Anaconda) - conda install tensorflow-gpu=1.11.0

==> WARNING: A newer version of conda exists. <==
    current version: 4.7.12
    latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- tensorflow-gpu=1.11.0

The following packages will be downloaded:

  package          |      build
  -----|-----
  _tfflow_select-2.1.0 |      gpu      3 KB
  cudatoolkit-9.0   |      1        227.6 MB
  cudnn-7.6.5       |      cuda9.0_0 142.6 MB
  tensorflow-1.11.0 |      py36he025d50_0 3.0 MB
  tensorflow-1.11.0 |      gpu_py36h5dc63e2_0 5 KB
  tensorflow-base-1.11.0 |      gpu_py36h6e53903_0 73.1 MB
  tensorflow-gpu-1.11.0 |      h0d30ee6_0 4 KB
  -----
                                         Total:        446.3 MB
```

*Gambar 5. 10 Proses Download Packages pada Instalasi Tensorflow GPU*

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

```
Downloading and Extracting Packages
tensorflow-1.11.0           | 5 KB      | #####
########################################################################| 100%
tensorflow-base-1.11         | 73.1 MB   | #####
########################################################################| 100%
cudnn-7.6.5                  | 142.6 MB  | #####
########################################################################| 100%
tensorflow-gpu-1.11.        | 4 KB      | #####
########################################################################| 100%
_tflow_select-2.1.0          | 3 KB      | #####
########################################################################| 100%
cuda toolkit-9.0              | 227.6 MB  | #####
########################################################################| 100%
tensorboard-1.11.0            | 3.0 MB    | #####
########################################################################| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Gambar 5. 11 Proses Instalasi Tensorflow GPU berhasil

### 5.2.3 Kelebihan Tensorflow

Tensorflow memiliki kelebihan antara lain:

- Cepat
  - Performa merupakan faktor penting dalam mengembangkan dan menerapkan sistem machine learning. Karena itulah *TensorFlow* menggunakan XLA, pengkompilasi aljabar linear canggih yang membuat kode *TensorFlow* mampu berjalan secepat mungkin pada prosesor, CPU, GPU, TPU, dan *platform hardware* lain yang digunakan.

- Fleksibel

*Tensorflow* menyediakan API level tinggi yang memudahkan dalam mengembangkan dan melatih model, serta kontrol level rendah untuk mendapatkan fleksibilitas dan performa yang maksimal.

- Siap Produksi

Lingkup penggunaan *tensorflow* meliputi riset penyelidikan hingga produksi skala besar. Gunakan API *tensorflow* yang sama dan telah dipahami, baik untuk membuat jenis model baru maupun memproses jutaan permintaan dalam produksi.

### 5.3 Library Pandas

#### 5.3.1 Pengertian Pandas



Gambar 5. 12 Logo Pandas

Pandas kependekan dari Python Data Analysis Library. Nama Pandas tersebut adalah turunan dari kata Panel Data. Pandas merupakan sebuah *open source python package/library* dengan lisensi BSD yang menyediakan banyak perkakas untuk kebutuhan data analisis, manipulasi dan pembersihan data. Pandas mendukung pembacaan dan penulisan data dengan media berupa *excel spreadsheet*, CSV, dan SQL yang kemudian akan dijadikan sebagai objek python

dengan *rows* dan *columns* yang disebut *data frame* seperti halnya pada tabel statistik.[18]

Pandas (Python Data Analysis Library) adalah distribusi/modul opensource dari python untuk melakukan data manajemen dan data analisis. Pandas diciptakan oleh Wes McKinney pada sekitaran tahun 2008 dan diperbaharui oleh salah satu kolega nya Sien Chang pada tahun 2010. Pandas muncul dari kebutuhan komunitas pengguna *open source* khususnya python akan packages khusus yang berguna untuk melakukan analisis data, selain import dan export data, melakukan manipulasi data dan sebagainya. Melakukan import data menggunakan distribusi pandas dalam python merupakan hal yang tidak mudah, membutuhkan ketelitian mengenai struktur data. [19]

Pandas memadukan library NumPy yang memiliki kemampuan manipulasi data yang fleksibel dengan database relasional (seperti SQL). Sehingga memudahkan kita untuk melakukan reshape, slice dan dice, agregasi data, dan mengakses subset dari data. Menurut penulis buku Python for Data Analysis sekaligus pembuat pandas, Wes McKinney, nama pandas berdasarkan dari panel data, yaitu istilah ekonometrik untuk data multi-dimensi terstruktur, dan berdasarkan dari kata yang merupakan fungsional library itu sendiri yaitu Python data analysis. Untuk memulai, lakukan import terlebih dahulu library pandas as pd. Penggunaan as di sini, artinya kita menggantikan pemanggilan pandas dengan prefix pd untuk proses berikutnya.

Adapun beberapa fungsi dan fitur yang berguna pada Pandas seperti dikutip dari *Learning Pandas, Second Edition* oleh Michael Heydt(*Introducing pandas*) sebagai berikut:

- Fast and efficient Series and DataFrame objects for data manipulation with integrated indexing
- Intelligent data alignment using indexes and labels
- Integrated handling of missing data
- Facilities for converting messy data into orderly data (tidying)
- Built-in tools for reading and writing data between in-memory data structures and files, databases, and web services
- The ability to process data stored in many common formats such as CSV, Excel, HDF5, and JSON
- Flexible reshaping and pivoting of sets of data
- Smart label-based slicing, fancy indexing, and subsetting of large datasets
- Columns can be inserted and deleted from data structures for size mutability
- Aggregating or transforming data with a powerful data grouping facility to perform split-apply-combine on datasets
- High-performance merging and joining of datasets
- Hierarchical indexing facilitating working with high-dimensional data in a lower-dimensional data structure
- Extensive features for time series data, including date range generation and frequency conversion, moving window

statistics, moving window linear regressions, date shifting, and lagging

- Highly optimized for performance, with critical code paths written in *Cython* or C

### 5.3.2 Struktur Data Pandas

Pandas memiliki dua tipe struktur data untuk versi terbaru dan satu deprecated struktur data:

#### 1. Series

Series merupakan struktur data dasar dalam Pandas. Series bisa juga diibaratkan sebagai array satu dimensi seperti halnya yang ada pada numpy array, hanya bedanya mempunyai index dan kita dapat mengontrol index dari setiap elemen tersebut. Struktur data yang bisa ditampung berupa *integer*, *float*, dan juga *string*. Series juga mendukung operasi vektor. Secara definisi, Series tidak dapat mempunyai kolom ganda, untuk masalah ini bisa menggunakan struktur data *frame*.

#### 2. Data Frame

DataFrame adalah objek yang memiliki struktur data tabular, berorientasi pada kolom dengan label baris dan kolom. Data frame merupakan array dua dimensi dengan baris dan kolom. Struktur data ini merupakan cara paling standar untuk menyimpan data. Secara sederhana, data frame merupakan tabel/data tabular. Setiap kolom pada Data Frame merupakan objek dari Series, dan baris terdiri dari elemen yang ada pada Series. Baris berguna untuk

menyimpan informasi dan kolom untuk menyimpan nama dari informasi tersebut.

### 3. Panel (deprecated)

#### 5.3.3 Cara Instalasi Pandas

Berikut adalah langkah untuk install *numpy* dengan *anaconda*.

- Buka Anaconda prompt (di sini saya menggunakan Anaconda3)
- Kemudian buat environment pada anaconda. Sebelumnya saya sudah buat environment dengan nama “Buku\_Tutorial”
- Aktifkan environment Buku\_Tutorial yang telah kita buat pada langkah 2 dengan perintah berikut:

```
Administrator: Anaconda Prompt (Anaconda3)
(base) C:\windows\system32>activate Buku_Tutorial
(Buku_Tutorial) C:\windows\system32>
```

Gambar 5. 13 Langkah Untuk Aktifkan Environment Buku\_Tutorial

- Masukkan perintah yang sesuai untuk menginstal *library Pandas* di dalam conda environment Anda.

```
(Buku_Tutorial) C:\Users\Mifta>conda install pandas
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Gambar 5. 14 Proses Instalasi Pandas

```

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.12
  latest version: 4.8.1

Please update conda by running

  $ conda update -n base -c defaults conda


## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- pandas

The following packages will be downloaded:

  package          |      build
  -----|-----
  pandas-1.0.0    | py36he350917_0      9.4 MB  conda-forge
  python-dateutil-2.8.1 | py_0            220 KB  conda-forge
  pytz-2019.3     | py_0            237 KB  conda-forge
  -----
                                         Total:      9.8 MB

The following NEW packages will be INSTALLED:

  pandas           conda-forge/win-64::pandas-1.0.0-py36he350917_0
  python-dateutil  conda-forge/noarch::python-dateutil-2.8.1-py_0
  pytz             conda-forge/noarch::pytz-2019.3-py_0

```

*Gambar 5. 15 Proses Download Packages pada Instalasi Pandas*

```

Proceed ([y]/n)? y

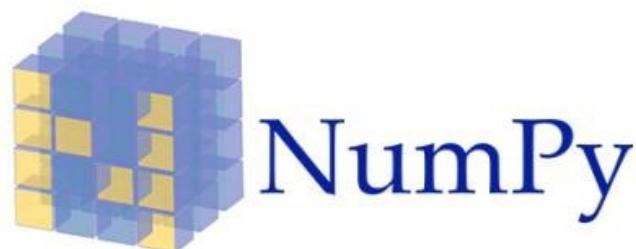
Downloading and Extracting Packages
pandas-1.0.0          | 9.4 MB      | ######:
######_##_##_##_##_##_| 100%
pytz-2019.3           | 237 KB      | ######:
######_##_##_##_##_##_| 100%
python-dateutil-2.8.1 | 220 KB      | ######:
######_##_##_##_##_##_| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done

```

*Gambar 5. 16 Proses Instalasi Pandas berhasil*

## 5.4 Library Numpy

### 5.4.1 Pengertian Numpy



*Gambar 5. 17 Logo Numpy*

NumPy adalah library Python yang digunakan para data scientist. Berguna untuk mengolah data numerical. NumPy (kependekan dari Numerical Python) adalah salah satu library teratas yang dilengkapi dengan sumber daya yang berguna untuk membantu para data scientist mengubah Python menjadi alat analisis dan pemodelan ilmiah yang kuat. Libary Open source terpopuler ini tersedia di bawah lisensi BSD. Ini adalah pustaka Python dasar untuk melakukan tugas dalam komputasi ilmiah. NumPy adalah bagian dari ekosistem berbasis Python yang lebih besar dari tool open source yang disebut SciPy.[20]

NumPy (Numerical Python) adalah library Python yang fokus pada scientific computing. NumPy memiliki kemampuan untuk membentuk objek N-dimensional array, yang mirip dengan list pada Python. Keunggulan NumPy array dibandingkan dengan list pada Python adalah konsumsi memory yang lebih kecil serta runtime yang lebih cepat. NumPy juga memudahkan kita pada Aljabar

Linear, terutama operasi pada Vector (1-d array) dan Matrix (2-d array).[21]

Perpustakaan memberdayakan Python dengan struktur data substansial untuk mudah melakukan perhitungan multi-dimensi (multi-dimensional arrays) dan perhitungan matrik. Selain penggunaannya dalam menyelesaikan persamaan aljabar linier (linear algebra equations) dan perhitungan matematis lainnya, NumPy juga digunakan sebagai wadah multi-dimensi serbaguna untuk berbagai jenis data generik.

Lebih hebatnya, NumPy terintegrasi dengan bahasa pemrograman lain seperti C / C ++ dan Fortran. Fleksibilitas perpustakaan NumPy memungkinkannya untuk dengan mudah dan cepat bergabung dengan berbagai database dan tools. Sebagai contoh, mari kita lihat bagaimana NumPy (disingkat np) dapat digunakan untuk mengalikan dua matriks.[20]

#### **5.4.2 Kelebihan dan Kekurangan Numpy**

Kelebihan menggunakan library NumPy dibandingkan *lists of lists* native yang disediakan Python:

1. Melakukan operasi komputasi terhadap data lebih mudah
2. Indexing dan slicing lebih cepat dan lebih mudah
3. Konversi data lebih cepat

Kekurangannya:

1. Semua item dalam NumPy array harus memiliki tipe data yang sama

2. Kolom dan Row hanya bisa direfer menggunakan angka, jadi kita harus bolak balik memastikan nama kolom XXX untuk kolom nomor Y, dll[22]

#### 5.4.3 Cara Instalasi Numpy

Masukkan perintah yang sesuai untuk menginstal *library numpy* di dalam *conda environment* Anda.

```
(Buku_Tutorial) C:\Users\Mifta>conda install numpy  
Collecting package metadata (current_repodata.json): done  
Solving environment: done
```

Gambar 5. 18 Proses Instalasi Numpy

```
==> WARNING: A newer version of conda exists. <==  
      current version: 4.7.12  
      latest version: 4.8.1
```

```
Please update conda by running
```

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

```
# All requested packages already installed.
```

Gambar 5. 19 Proses Instalasi Numpy berhasil

#### 5.5 Library OpenCV

##### 5.5.1 Pengertian OpenCV



Gambar 5. 20 Logo OpenCV

OpenCV (*Open Source Computer Vision*) merupakan sebuah library yang di khususkan untuk penglihatan komputer secara real time yang di kembangkan oleh pusat penelitian Intel di Nizhny Novgorod, Rusia. Pengunaan library ini tanpa di kenakan biaya, bila ingin menggunakannya kita tidak perlu meng-crack, karena telah dirilis dibawah lisensi BSD (Berkeley Software Distribution). Kata Open pada OpenCV dimaksudkan open source yakni gratis, tidak berbayar, bisa didownload oleh siapa saja. Sementara CV pada kata OpenCV adalah kependekan dari Computer Vision, maksudnya adalah komputer yang digunakan untuk mengolah image (citra/gambar) yang ditangkap oleh alat perekam (seperti kamera atau webcam) yang dikonversi dari analog ke digital lalu diolah di dalam komputer. Tujuan pengolahan image ini ada banyak, di antaranya untuk memperbaiki kualitas gambar atau untuk indentifikasi gambar.

OpenCV ini dapat di gunakan di berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan suport dengan Windows, Linux, Mac OS, iOS dan Android. OpenCV didesain untuk efisiensi komputasi dan dengan fokus yang kuat pada aplikasi real-time. Di seluruh dunia, library ini telah digunakan lebih dari 47 ribu pengguna dan jumlah downloadnya diperkirakan telah melebihi 7 juta kali.

Cara kerja yang di gunakan pada Computer Vision ini yaitu dengan mencoba meniru cara kerja system penglihatan/visual manusia, dimana objek di dapat dengan penglihatan mata dan selanjutnya citra yang di tangkap oleh mata diteruskan ke otak. Di

otak data citra yang di dapat akan di interpretasi atau di proses sehingga kita dapat mengerti objek apa yang kita lihat. Selanjutnya hasil dari proses penglihatan tadi dapat di gunakan untuk mengambil tindakan atau keputusan selanjutnya, misalkan tersenyum saat berpapasan dengan cewek yang kita takdir. Sehingga Computer vision ini termasuk dalam kecerdasan buatan atau istilah kerennya artificial intelligence. Tujuan dari ini adalah membuat bagaimana si komputer dapat melihat dan membedakan objek A dengan objek B dari apa yang di lihatnya.[23]

### **5.5.2 Sejarah OpenCV**

OpenCV pertama kali diluncurkan secara resmi pada tahun 1999 oleh Inter Research sebagai lanjutan dari bagian proyek bertajuk aplikasi intensif berbasis CPU, real-time ray tracing dan tembok penampil 3D. Para kontributor utama dalam proyek ini termasuk mereka yang berkecimpung dalam bidang optimasi di Intel Russia, dan juga Tim Pusatka Performansi Intel. Pada awalnya, tujuan utama dari proyek OpenCV ini dideskripsikan sebagai berikut,

- Penelitian penginderaan citra lanjutan tidak hanya melalui kode program terbuka, tetapi juga kode yang telah teroptimasi untuk infrastruktur penginderaan citra.
- Menyebarluarkan ilmu penginderaan citra dengan menyediakan infrastruktur bersama di mana para pengembang dapat menggunakannya secara bersama-sama, sehingga kode akan tampak lebih mudah dibaca dan ditransfer.

- Membuat aplikasi komersial berbasiskan penginderaan citra, di mana kode yang telah teroptimasi tersedia secara bebas dengan lisensi yang tersedia secara bebas yang tidak mensyaratkan program itu harus terbuka atau gratis.[24]

### 5.5.3 Fitur utama dari OpenCV

OpenCV mempunyai banyak fitur yang dapat dimanfaatkan, berikut ini adalah fitur utama dari OpenCV antara lain :

- Image and video I/O

Dengan antar muka ini kita dapat membaca data gambar dari file, atau dari umpan video langsung. Dan juga dapat menciptakan file gambar maupun video.

- Computer Vision secara umum dan pengolahan citra digital  
( untuk low dan mid level API)

Dengan antar muka ini kita dapat melakukan eksperimen uji coba dengan berbagai standar algoritma computer vision. Termasuk juga deteksi garis, tepi, pucuk, proyeksi elips, image pyramid untuk pemrosesan gambar multi skala, pencocokan template, dan berbagai transform ( Fourier, cosine diskrit, distance transform) dan lain lain.

- Modul computer vision high level

Di dalam OpenCV juga termasuk kemampuan “high level”, seperti kemampuan tambahan untuk deteksi wajah, pengenalan wajah, termasuk optical flow

- Metode untuk AI dan machine learning

Applikasi computer vision sering kali memerlukan machine learning atau metode AI lainnya, beberapa metode tersebut tersedia dalam paket OpenCV machine learning.

- Sampling gambar dan transformasi

Di dalam OpenCV sudah terdapat antar muka untuk substraksi subregion dari gambar, random sampling, rotating, dan lain lain.

- Metode untuk menciptakan dan menganalisa gambar biner
- Metode untuk memperhitungkan pemodelan 3D

Fungsi ini sangat bermanfaat untuk mapping dan localization, baik untuk stereo camera ataupun satu kamera dengan berbagai sudut pandang.[25]

Keistimewaan yang dimiliki open CV :

- ~ Manipulation data citra
- ~ Citra dan video I/O
- ~ Manipulasi Matriks dan Vektor beserta rutin-rutin aljabar linear
- ~ Data struktur dinamis
- ~ Pemroses Citra fundamental
- ~ Analisis struktur
- ~ Kalibrasi kamera
- ~ Analisis gerakan
- ~ Pengenalan obyek
- ~ Graphical User Interface
- ~ Pelabelan citra

#### **5.5.4 Kelebihan Dan Kekurangan OpenCV**

Kelebihan Opencv:

1. Opencv adalah program aplikasi yang *freeware*, sehingga kita dapat mendownload dan menggunakan program aplikasi ini dengan gratis.
2. memiliki *library* dokumen yang cukup banyak
3. Dapat bekerja secara cepat pada komputer yang menggunakan processor *Intel*.
4. Komputasi yang lebih ringan bila dibandingkan menggunakan Matlab dalam hal pengolahan citra digital.
5. dapat bekerja secara *real time*.

Selain memiliki beberapa keunggulan, ternyata Opencv memiliki beberapa kekurangan diantaranya adalah:

1. Kurang mendukung dengan program bahasa yang berbasis .NET seperti C# dan VB sehingga untuk menggunakan bahasa tersebut kita perlu menggunakan *wrappers*
2. Installasi yang lumayan ribet.

#### **5.5.5 Cara Instalasi OpenCV**

Masukkan perintah yang sesuai untuk menginstal *library* OpenCV di dalam *conda environment* Anda.

```
(Buku_Tutorial) C:\windows\system32>conda install opencv  
Collecting package metadata (current_repodata.json): done  
Solving environment: done
```

*Gambar 5. 21 Proses Instalasi OpenCV*

```

Administrator: Anaconda Prompt (Anaconda) - conda install opencv

==> WARNING: A newer version of conda exists. <==
    current version: 4.7.12
    latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- opencv

The following packages will be downloaded:

package          |      build
-----|-----
freetype-2.10.0 | h563cf7_1      476 KB  conda-forge
icu-64.2        | he025d50_1     14.1 MB  conda-forge
jpeg-9c         | hfa6e2cd_1001   314 KB  conda-forge
libclang-9.0.1  | default_hf44288c_0 20.8 MB  conda-forge
liblapacke-3.8.0| 14_mkl          3.5 MB  conda-forge
libopencv-4.2.0 | py36_2          45.0 MB  conda-forge
libpng-1.6.37   | h7602738_0     1.3 MB  conda-forge
libtiff-4.1.0   | h21b02b4_3     1.0 MB  conda-forge
libwebp-1.0.2   | hfa6e2cd_5     356 KB  conda-forge
lz4-c-1.8.3    | he025d50_1001  255 KB  conda-forge
opencv-4.2.0    | py36_2          19 KB   conda-forge
py-opencv-4.2.0| py36h5ca1d4c_2  21 KB   conda-forge
qt-5.12.5       | h7ef1ec2_0     104.4 MB conda-forge
sqlite-3.30.1   | hfa6e2cd_0     1.1 MB  conda-forge
xz-5.2.4        | h2fa13f4_1001  805 KB  conda-forge
zstd-1.4.4      | hd8a0e53_1     1.7 MB  conda-forge

```

*Gambar 5. 22 Proses Download Packages pada Instalasi OpenCV*

```
Administrator: Anaconda Prompt (Anaconda) - conda install opencv

Proceed ([y]/n)? y

Downloading and Extracting Packages
qt-5.12.5 | 104.4 MB | #####
##### | 100%
jpeg-9c | 314 KB | #####
##### | 100%
libpng-1.6.37 | 1.3 MB | #####
##### | 100%
libtiff-4.1.0 | 1.0 MB | #####
##### | 100%
xz-5.2.4 | 805 KB | #####
##### | 100%
opencv-4.2.0 | 19 KB | #####
##### | 100%
libclang-9.0.1 | 20.8 MB | #####
##### | 100%
sqlite-3.30.1 | 1.1 MB | #####
##### | 100%
freetype-2.10.0 | 476 KB | #####
##### | 100%
liblapacke-3.8.0 | 3.5 MB | #####
##### | 100%
libopencv-4.2.0 | 45.0 MB | #####
##### | 100%
py-opencv-4.2.0 | 21 KB | #####
##### | 100%
icu-64.2 | 14.1 MB | #####
##### | 100%
lz4-c-1.8.3 | 255 KB | #####
##### | 100%
libwebp-1.0.2 | 356 KB | #####
##### | 100%
zstd-1.4.4 | 1.7 MB | #####
##### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Gambar 5. 23 Proses Instalasi OpenCV berhasil

## 5.6 Library PyTorch

### 5.6.1 Pengertian PyTorch



Gambar 5. 24 Logo PyTorch

Berbasis sumber terbuka, PyTorch adalah perpustakaan pembelajaran mesin yang dikembangkan oleh kelompok riset AI dari Facebook. PyTorch memungkinkan setiap pengembang untuk melakukan perhitungan *tensor* dengan akselerasi GPU, membuat grafik komputasi dinamis, dan menghitung gradien secara otomatis.

Selain itu, PyTorch menawarkan API yang kaya untuk menyelesaikan masalah aplikasi yang berkaitan dengan jaringan saraf. Dengan alat dan perpustakaan yang luas, PyTorch menyediakan banyak sumber daya untuk mendukung pengembangan platform seperti AllenNLP, ELF, dan Glow.

Lebih lanjut, PyTorch sendiri merupakan perpustakaan pembelajaran mesin yang didasarkan oleh Torch. Diperkenalkan pada tahun 2017, PyTorch berhasil mendapatkan popularitas yang baik dan telah menarik banyak pengembang pembelajaran mesin untuk menggunakan hingga sekarang. [26]

### 5.6.2 Cara Instalasi PyTorch

Masukkan perintah yang sesuai untuk menginstal *library* Pytorch di dalam *conda environment* Anda.

Administrator: Anaconda Prompt (Anaconda) - conda install opencv - conda install pytorch

```
(Buku_Tutorial) C:\windows\system32>conda install pytorch
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Gambar 5. 25 Proses Instalasi PyTorch

```
Administrator: Anaconda Prompt (Anaconda) - conda install opencv - conda install pytorch
Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- pytorch

The following packages will be downloaded:

  package          |      build
  -----|-----
  _pytorch_select-1.1.0 |           cpu      2 KB
  cffi-1.13.2        | py36hb32ad35_0  220 KB  conda-forge
  ninja-1.10.0        | h1ad3211_0   122 KB  conda-forge
  pycparser-2.19       | py36_1        173 KB  conda-forge
  pytorch-1.3.1        | cpu_py36h9f948e0_0 50.8 MB
  -----
                           Total:    51.3 MB

The following NEW packages will be INSTALLED:

  _pytorch_select      pkgs/main/win-64::_pytorch_select-1.1.0-cpu
  cffi                  conda-forge/win-64::cffi-1.13.2-py36hb32ad35_0
  ninja                 conda-forge/win-64::ninja-1.10.0-h1ad3211_0
  pycparser             conda-forge/win-64::pycparser-2.19-py36_1
  pytorch                pkgs/main/win-64::pytorch-1.3.1-cpu_py36h9f948e0_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
  _pytorch_select-1.1.0 | 2 KB      | #####|#####
  pycparser-2.19         | 173 KB     | #####|#####
  ############################################################################ | 100%
```

Gambar 5. 26 Proses Download dan Install Packages

```
ninja-1.10.0           | 122 KB  | #####| 100%
cffi-1.13.2           | 220 KB  | #####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

Gambar 5. 27 Proses Instalasi Pytorch Berhasil

## 5.7 Library Theano

### 5.7.1 Pengertian Theano



Gambar 5. 28 Logo Theano

Theano merupakan sebuah Python library yang diperuntukan untuk memodelkan sebuah permasalahan matematika. Theano akan mentransform code yang kita tuliskan kedalam bentuk *optimized C* sehingga pada kasus tertentu performa Theano dapat lebih cepat dibandingkan '*hand-crafted C*' implementasi . Selain itu, Theano juga dapat dijalankan menggunakan GPU, yang tentunya lebih cepat dibanding CPU dalam proses komputasi dan training ANN dengan jumlah data yang banyak.[27]

### 5.7.2 Cara Instalasi Theano

Masukkan perintah yang sesuai untuk menginstal *library Theano* di dalam *conda environment* Anda.

```
Administrator: Anaconda Prompt (Anaconda) - conda install opencv - conda install pytorch - conda install theano

(Buku_Tutorial) C:\windows\system32>conda install theano
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

### Gambar 5. 29 Proses Instalasi Theano

```
Select Administrator: Anaconda Prompt (Anaconda) - conda install opencv - conda install pytorch - conda install theano

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
  - theano

The following packages will be downloaded:

  package          |      build
  -----|-----
libgpuarray-0.7.6 | hfa6e2cd_1003    314 KB  conda-forge
mako-1.1.0        | py_0                  57 KB   conda-forge
markupsafe-1.1.1  | py36hfa6e2cd_0    28 KB   conda-forge
pygpu-0.7.6       | py36hc8d92b1_1000 594 KB  conda-forge
theano-1.0.4       | py36h6538335_1001 3.7 MB  conda-forge
vs2015_win-64-14.0.25420 | h55c1224_11      7 KB

Total:           4.7 MB

The following NEW packages will be INSTALLED:

  libgpuarray      conda-forge/win-64::libgpuarray-0.7.6-hfa6e2cd_1003
  mako             conda-forge/noarch::mako-1.1.0-py_0
  markupsafe       conda-forge/win-64::markupsafe-1.1.1-py36hfa6e2cd_0
  pygpu            conda-forge/win-64::pygpu-0.7.6-py36hc8d92b1_1000
  theano           conda-forge/win-64::theano-1.0.4-py36h6538335_1001
  vs2015_win-64   pkgs/main/win-64::vs2015_win-64-14.0.25420-h55c1224_11

Proceed ([y]/n)? y

Downloading and Extracting Packages
mako-1.1.0          | 57 KB      | #####| 100%
pygpu-0.7.6         | 594 KB     | #####| 100%
theano-1.0.4         | 3.7 MB     | #####| 100%
vs2015_win-64-14.0.2 | 7 KB       | #####| 100%
libgpuarray-0.7.6   | 314 KB     | #####| 100%
markupsafe-1.1.1    | 28 KB      | #####| 100%
```

### Gambar 5. 30 Proses Download dan Install Packages

```
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

### Gambar 5. 31 Proses Instalasi Theano Berhasil

## 5.8 Library Keras

### 5.8.1 Pengertian Keras



*Gambar 5. 32 Logo Keras*

Keras merupakan interface library yang dibangun untuk mensederhanakan implementasi algoritma-algoritma Deep Learning di atas TensorFlow. TensorFlow sendiri merupakan platform High Performance computing berbasis alur graph. Oh ya, bicara tentang TensorFlow banyak teman-teman yang salah paham, dan mengira TensorFlow merupakan library yang berisi kumpulan algoritma Deep Learning. Yang sebenarnya adalah, banyak kontributor TensorFlow yang mengimplementasikan algoritma Deep Learning di atas TensorFlow. [28]

Keras berisi banyak implementasi blok bangunan jaringan saraf yang biasa digunakan seperti lapisan, tujuan, fungsi aktivasi, pengoptimal, dan sejumlah alat untuk membuat bekerja dengan data gambar dan teks lebih mudah untuk menyederhanakan pengkodean yang diperlukan untuk menulis kode Jaringan Neural Network. Kode ini dihosting di GitHub, dan forum dukungan

komunitas menyertakan halaman masalah GitHub, dan saluran Slack.

Selain jaringan saraf standar, Keras memiliki dukungan untuk jaringan saraf konvolusional dan berulang. Ini mendukung lapisan utilitas umum lainnya seperti dropout, normalisasi batch, dan pengumpulan. Keras memungkinkan pengguna untuk menghasilkan model yang dalam pada telepon pintar (iOS dan Android), di web, atau di Java Virtual Machine. Hal ini juga memungkinkan penggunaan pelatihan terdistribusi model pembelajaran mendalam tentang kelompok Unit Pemrosesan Grafis (GPU) dan unit pemroses Tensor (TPU) terutama dalam hubungannya dengan CUDA.

Beberapa fitur utama dari Keras:

- Mampu menjalankan *source code* yang sama menggunakan CPU atau GPU dengan lancar
- API yang *user-friendly* sehingga mempermudah penggunanya dalam proses prototipe model deep learning
- Dukungan *built-in* untuk CNN atau Convolutional Neural Networks (Computer Vision), RNN atau Recurrent Neural Networks (untuk *sequence processing*), dan kombinasi keduanya
- Dapat digunakan untuk hampir semua jenis dari model deep learning

### 5.8.1 Cara Instalasi Keras

Masukkan perintah yang sesuai untuk menginstal *library Keras* di dalam *conda environment* Anda.

```
(Buku_Tutorial) C:\Windows\System32>conda install keras
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

Gambar 5. 33 proses Instalasi Keras

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

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- keras

The following packages will be downloaded:

package          |      build
-----|-----
keras-2.3.1     | py36h21ff451_0      589 KB  conda-forge
pyyaml-5.3       | py36hfa6e2cd_0      154 KB  conda-forge
yaml-0.2.2       | hfa6e2cd_1           63 KB   conda-forge
-----|-----
                                         Total:    806 KB

The following NEW packages will be INSTALLED:

keras            conda-forge/win-64::keras-2.3.1-py36h21ff451_0
pyyaml           conda-forge/win-64::pyyaml-5.3-py36hfa6e2cd_0
yaml              conda-forge/win-64::yaml-0.2.2-hfa6e2cd_1

Proceed ([y]/n)? y

Downloading and Extracting Packages
keras-2.3.1      | 589 KB      | #####|#####
pyyaml-5.3        | 154 KB      | #####|#####
yaml-0.2.2        | 63 KB       | #####|#####
```

Gambar 5. 34 Proses Download dan Install Packages

```
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done
```

Gambar 5. 35 Proses Instalasi Keras Berhasil

## 5.9 Library Sklearn

### 5.9.1 Pengertian Sklearn



Gambar 5. 36 Logo Sklearn

Scikit-learn adalah *library* untuk *machine learning* bagi para pengguna python. Scikit-learn merupakan *free software*, dan memungkinkan kita melakukan beragam pekerjaan dalam *Data Science*, seperti regresi (*regression*), klasifikasi (*classification*), pengelompokan/penggugusan (*clustering*), data *preprocessing*, *dimensionality reduction*, dan *model selection* (pembandingan, validasi, dan pemilihan parameter maupun model). Salah satu library yg memudahkan untuk membuat machine learning adalah scikit. Menggunakan library ini memudahkan pemrograman karena

lebih sedikit baris kode dibanding library lainnya seperti tensorflow atau numpy.

Scikit-Learn bisa digunakan untuk :

- Classification
- Regression
- Clustering
- Dimensionality Reduction
- Model Selection
- Preprocessing

### 5.9.2 Cara Instalasi Sklearn

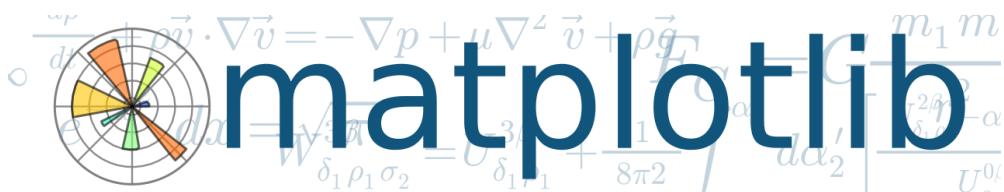
Masukkan perintah yang sesuai untuk menginstal *library* *sklearn* di dalam *conda environment* Anda.

```
(Buku_Tutorial) C:\Windows\System32>pip install sklearn
WARNING: pip is being invoked by an old script wrapper. This will fail in a future version of
pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip directly.
Processing c:/users/mifta/appdata/local/pip/cache/wheels/76\03\bb\589d421d27431bcd2c6da284d5f2
286c8e3b2ea3cf1594c074\sklearn-0.0-py2.py3-none-any.whl
Collecting scikit-learn
  Downloading scikit_learn-0.22.1-cp36-cp36m-win_amd64.whl (6.3 MB)
    [██████████] | 6.3 MB 252 kB/s
Collecting joblib>=0.11
  Downloading joblib-0.14.1-py2.py3-none-any.whl (294 kB)
    [██████████] | 294 kB 3.2 MB/s
Requirement already satisfied: numpy>=1.11.0 in e:/anaconda/envs/buku_tutorial/lib/site-packages
  (from scikit-learn->sklearn) (1.18.1)
Requirement already satisfied: scipy>=0.17.0 in e:/anaconda/envs/buku_tutorial/lib/site-packages
  (from scikit-learn->sklearn) (1.4.1)
Installing collected packages: joblib, scikit-learn, sklearn
Successfully installed joblib-0.14.1 scikit-learn-0.22.1 sklearn-0.0
```

Gambar 5. 37 Proses Instalasi Sklearn Berhasil

## 5.10 Library Matplotlib

### 5.10.1 Pengertian Matplotlib



*Gambar 5. 38 Logo Matplotlib*

Matplotlib adalah library plotting 2D Python yang menghasilkan gambar publikasi bermutu di dalam berbagai format hardcopy dan lingkungan interaktif sepanjang platform. matplotlib dapat digunakan di dalam script Python, shell Python dan ipython (ala MATLAB or Mathematica), server aplikasi web, dan enam GUI toolkit. Matplotlib mencoba untuk membuat hal mudah menjadi lebih mudah dan hal sulit menjadi mungkin. Kita dapat membuat plot, histogram, power spectra, grafik batang, grafik error, scatterplot, dll, hanya dengan beberapa baris kode. [29]

## 5.10.2 Cara Instalasi Matplotlib

```
(Buku_Tutorial) C:\Windows\System32>conda install matplotlib
Collecting package metadata (current_repodata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.12
  latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda

## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- matplotlib

The following packages will be downloaded:

  package          |      build
  -----|-----
  cycler-0.10.0    |      py_2
  kiwisolver-1.1.0 |      py36he980bc4_0
  matplotlib-3.1.3 |      py36_0
  matplotlib-base-3.1.3 |      py36h2981e6d_0
  pyparsing-2.4.6   |      py_0
  pyqt-5.12.3       |      py36h6530335_1
  tornado-6.0.3     |      py36hfa6e2cd_0
  -----
                                         Total:    12.0 MB
```

Gambar 5. 39 Proses Instalasi Matplotlib

```
The following NEW packages will be INSTALLED:

cycler          conda-forge/noarch::cycler-0.10.0-py_2
kiwisolver      conda-forge/win-64::kiwisolver-1.1.0-py36he980bc4_0
matplotlib      conda-forge/win-64::matplotlib-3.1.3-py36_0
matplotlib-base conda-forge/win-64::matplotlib-base-3.1.3-py36h2981e6d_0
pyparsing        conda-forge/noarch::pyparsing-2.4.6-py_0
pyqt             conda-forge/win-64::pyqt-5.12.3-py36h6538335_1
tornado          conda-forge/win-64::tornado-6.0.3-py36hfa6e2cd_0
```

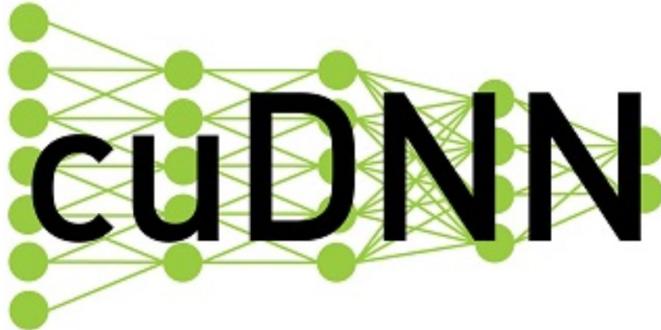
```
Proceed ([y]/n)? y
```

```
Downloading and Extracting Packages
matplotlib-base-3.1. | 6.6 MB  | #####|#####|#####|#####|#####|#####|#####|#####| 100%
kiwisolver-1.1.0    | 61 KB   | #####|#####|#####|#####|#####|#####|#####|#####| 100%
pyqt-5.12.3         | 4.7 MB  | #####|#####|#####|#####|#####|#####|#####|#####| 100%
tornado-6.0.3       | 639 KB  | #####|#####|#####|#####|#####|#####|#####|#####| 100%
cycler-0.10.0       | 9 KB    | #####|#####|#####|#####|#####|#####|#####|#####| 100%
pyparsing-2.4.6     | 59 KB   | #####|#####|#####|#####|#####|#####|#####|#####| 100%
matplotlib-3.1.3    | 7 KB    | #####|#####|#####|#####|#####|#####|#####|#####| 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

*Gambar 5. 40 proses Intalasi Berhasil*

## 5.11 Library Cudnn

### 5.11.1 Pengertian Cudnn



*Gambar 5. 41 Logo Cudnn*

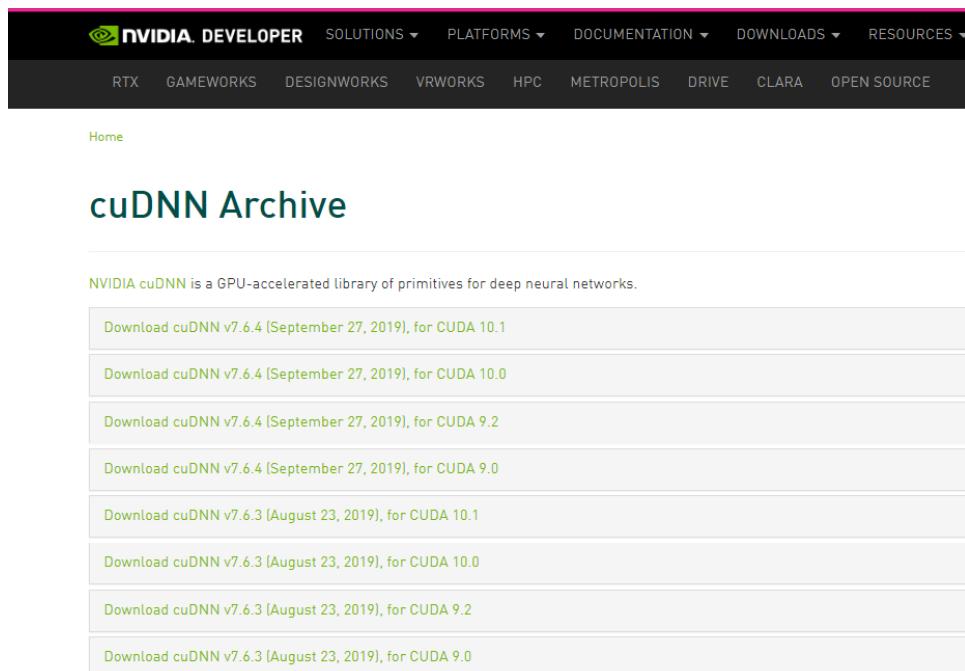
CuDNN adalah perpustakaan yang membantu mempercepat kerangka kerja pembelajaran yang dalam, seperti TensorFlow atau Theano. Pustaka Deep Natural Network NVDIA CUDA (cuDNN) adalah pustaka primitif yang diperuntukkan bagi DNN untuk GPU. CuDNN menyediakan implementasi yang sangat disesuaikan untuk rutin standar seperti konvolusi maju dan kata kunci, pooling,

normalisasi, dan lapisan aktivasi. CuDNN adalah bagian dari SDK pembelajaran NVIDIA.

### 5.11.2 Cara Instalasi Cudnn

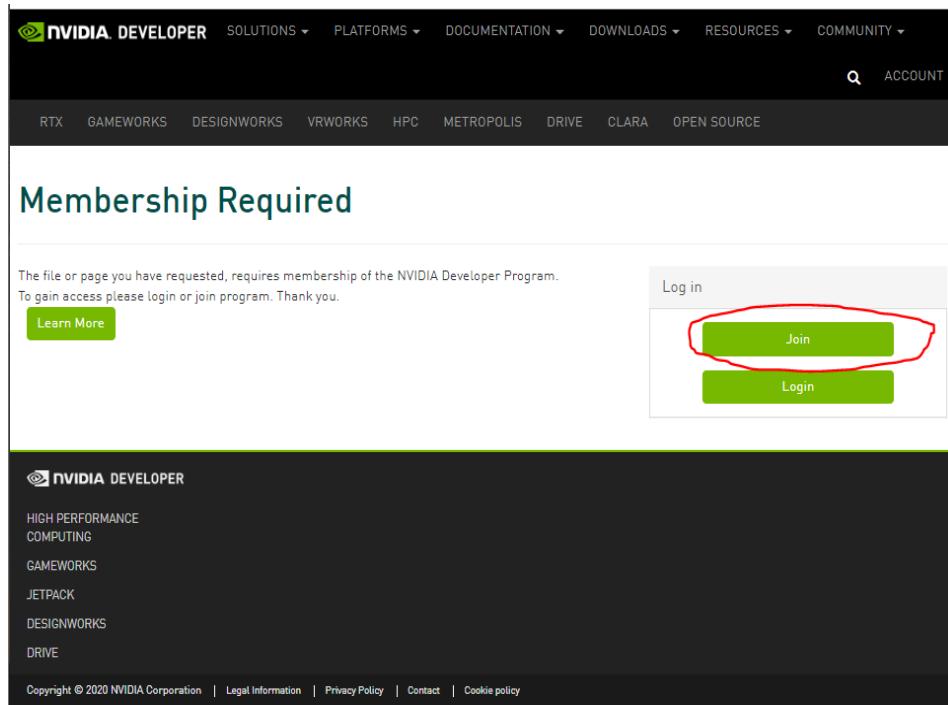
Untuk instalasi cudnn menyesuaikan dengan versi CUDA yang digunakan. Untuk mendapatkan cudnn maka kunjungi link berikut <https://developer.nvidia.com/rdp/cudnn-archive> kemudian sesuaikan versi cudnn yang akan di downlod dengan CUDA yang di install di computer.

1. Berikut tampilan website untuk mendownload cudnn



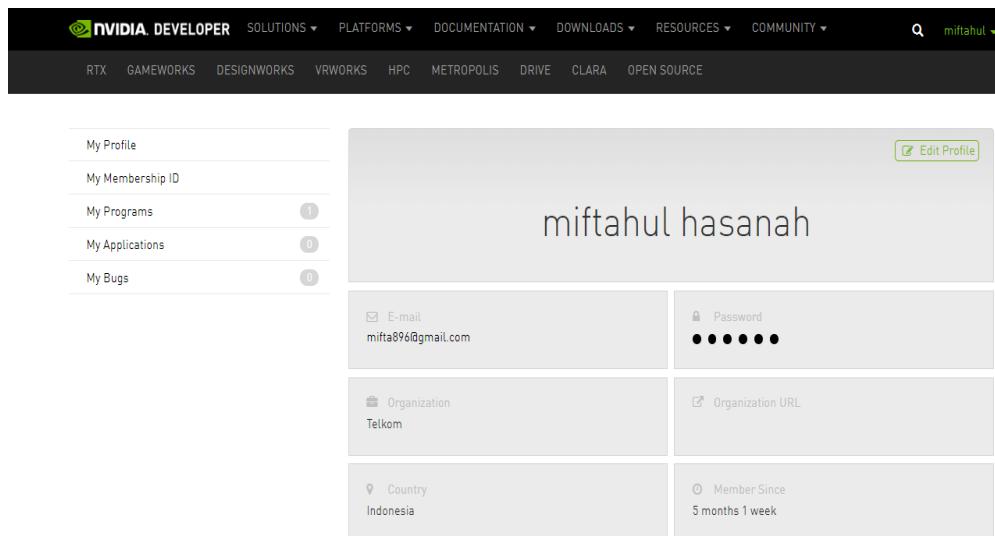
*Gambar 5. 42 Download Cudnn*

2. Saat ingin mendownload cudnn anda harus join dan memiliki akun membership agar bisa login dan mendownload file.



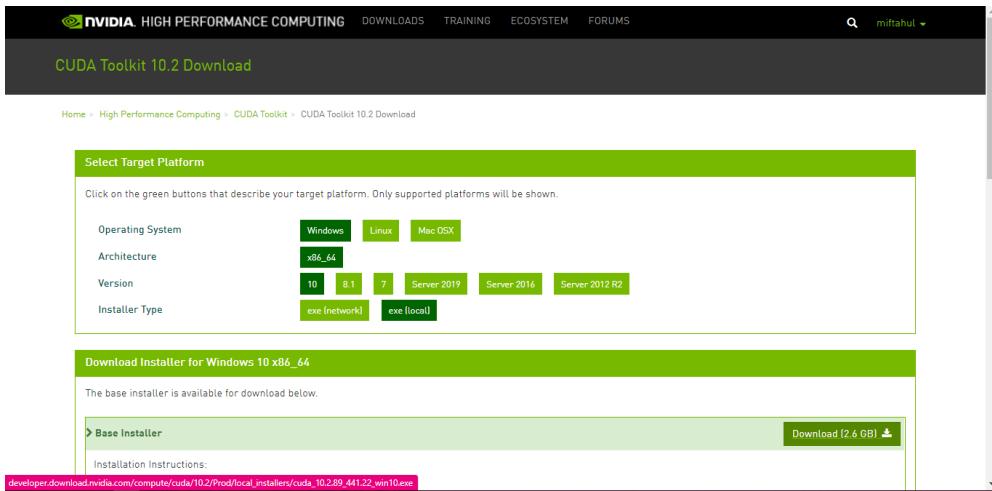
Gambar 5. 43 Daftar Membership

3. Login sebagai email yang didaftarkan.



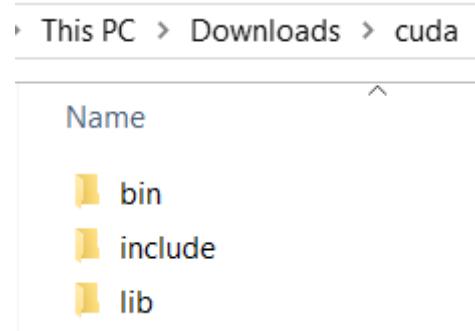
Gambar 5. 44 Login Email

#### 4. Download Cuda



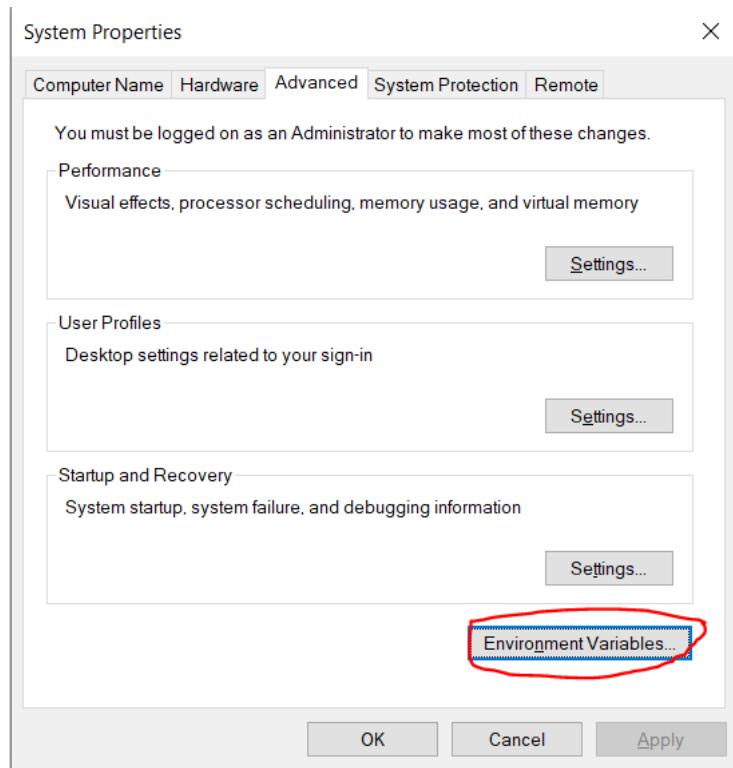
Gambar 5. 45 Download Cuda

5. Berikut file yang terdapat pada folder cudnn yang di download :



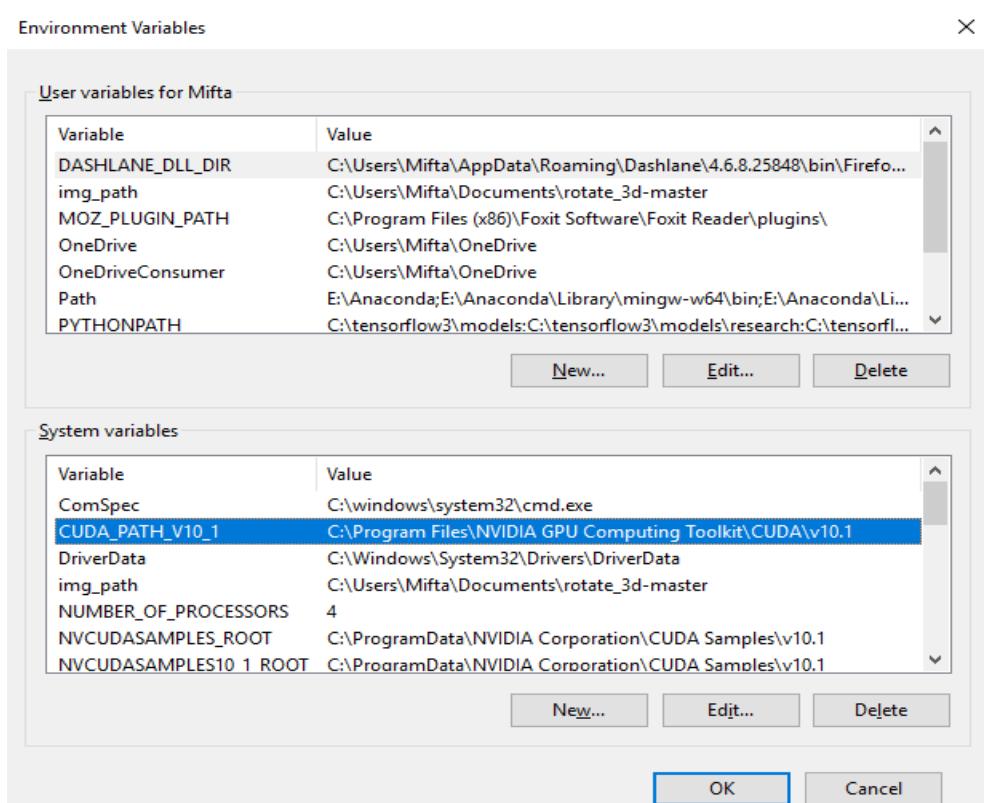
Gambar 5. 46 File cudnn

6. Selanjutnya buka path instalasi CUDA dengan melakukan klik environment variabels :



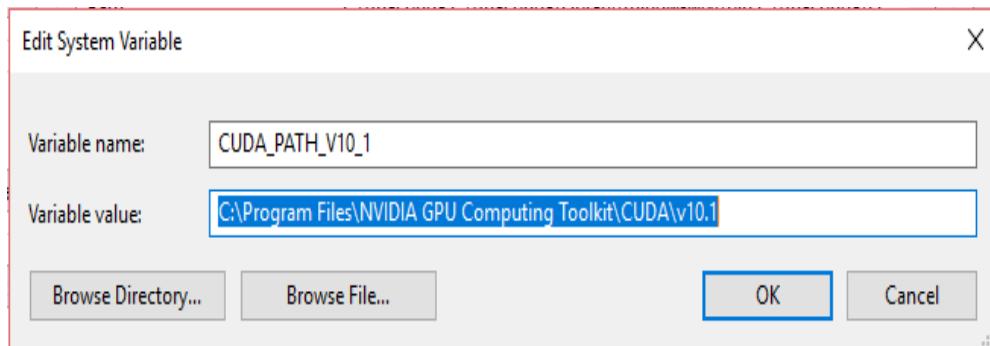
Gambar 5. 47 Membuka Environment

7. Lakukan double klik pada path cuda untuk mendapatkan lokasi penyimpanan file :



Gambar 5. 48 Membuka Path CUDA

8. Copy direktori berikut :



Gambar 5. 49 Copy Link

9. Berikut direktori CUDA yang sesuai dengan path :

System (C:) > Program Files > NVIDIA GPU Computing Toolkit > CUDA > v10.1			
Name	Date modified	Type	Size
bin	8/28/2019 10:25 AM	File folder	
doc	8/28/2019 10:24 AM	File folder	
extras	8/28/2019 10:25 AM	File folder	
include	8/28/2019 10:24 AM	File folder	
lib	8/28/2019 10:24 AM	File folder	
libnvvp	8/28/2019 10:24 AM	File folder	
nvml	8/28/2019 10:24 AM	File folder	
nvvm	8/28/2019 10:24 AM	File folder	
src	8/28/2019 10:24 AM	File folder	
tools	8/28/2019 10:24 AM	File folder	
CUDA_Toolkit_Release_Notes.txt	7/29/2019 12:27 PM	Text Document	55 KB
EULA.txt	7/29/2019 12:27 PM	Text Document	59 KB
version.txt	7/29/2019 12:28 PM	Text Document	1 KB

*Gambar 5. 50 Replace File*

Selanjutnya replace file bin, include, lib dengan file bin, include, lib dari hasil download cudnn tadi, jika sudah maka cudnn sudah terpasang pada CUDA.

## 5.12 Library Pillow

### 5.12.1 Pengertian Pillow

Pustaka Pencitraan Python (PIL) dalam versi yang lebih baru yang dikenal sebagai Pillow adalah pustaka gratis untuk bahasa pemrograman Python yang menambahkan dukungan untuk membuka, memanipulasi, dan menyimpan banyak format file gambar yang berbeda. Ini tersedia untuk Windows, Mac OS X dan Linux. Versi terbaru dari PIL adalah 1.1.7, dirilis pada September 2009 dan mendukung Python 1.5.2-2.7, dengan dukungan Python 3 akan dirilis "nanti".

Pengembangan tampaknya dihentikan dengan komitmen terakhir ke repositori PIL yang akan datang pada 2011. Akibatnya, proyek penerus yang disebut Pillow telah memotong repositori PIL dan menambahkan dukungan Python 3.x. Garpu ini telah diadopsi sebagai pengganti PIL asli dalam distribusi Linux termasuk Debian dan Ubuntu (sejak 13.04). Beberapa format file yang didukung adalah PPM, PNG, JPEG, GIF, TIFF, dan BMP. Dimungkinkan juga untuk membuat decoder file baru untuk memperluas pustaka format file yang dapat diakses.

Pillow menawarkan beberapa prosedur standar untuk manipulasi gambar. Ini termasuk:

- manipulasi per-pixel,
- masking dan penanganan transparansi,
- pemfilteran gambar, seperti mengaburkan, membentuk, menghaluskan, atau menemukan tepi,
- peningkatan gambar, seperti mempertajam, menyesuaikan kecerahan, kontras atau warna,
- Menambahkan teks ke gambar dan banyak lagi.

### **5.12.2 Instalasi Pillow**

Untuk melakukan instalasi pillow, buka anaconda command prompt kemudian aktivasi environment yang telah dibuat, dan jalankan perintah berikut “conda install pillow” (versi pillow menyesuaikan dengan kebutuhan machine learning yang akan dibuat). Berikut tampilan apabila berhasil menginstal pillow:

```
(Buku_Tutorial) C:\Windows\System32>conda install pillow
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

*Gambar 5. 51 Proses Instalasi Pillow*

```
Administrator: Anaconda Prompt (Anaconda) - conda install pillow
==> WARNING: A newer version of conda exists. <==
    current version: 4.7.12
    latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda


## Package Plan ##

environment location: E:\Anaconda\envs\Buku_Tutorial

added / updated specs:
- pillow

The following packages will be downloaded:

  package          |      build
  -----|-----
  olefile-0.46    |      py_0      31 KB  conda-forge
  pillow-7.0.0     |  py36h9ealdd6_0   711 KB  conda-forge
  tk-8.6.10        |      hfa6e2cd_0   3.2 MB  conda-forge
  -----
                           Total:       3.9 MB

The following NEW packages will be INSTALLED:

  olefile           conda-forge/noarch::olefile-0.46-py_0
  pillow            conda-forge/win-64::pillow-7.0.0-py36h9ealdd6_0
  tk                conda-forge/win-64::tk-8.6.10-hfa6e2cd_0

Proceed ([y]/n)? y

Downloading and Extracting Packages
tk-8.6.10          | 3.2 MB      | #####| 100%
#####| 100%
pillow-7.0.0        | 711 KB      | #####| 100%
#####| 100%
olefile-0.46        | 31 KB       | #####| 100%
#####| 100%
```

*Gambar 5. 52 Download Dan Install Packages Pillow*

```
.....
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

*Gambar 5. 53 Proses Instalasi Pillow Berhasil*

## **5.13 Library protobuf**

### **5.13.1 Pengertian Protobuf**



*Gambar 5. 54 Logo Protobuf*

Protokol buffer adalah solusi fleksibel, efisien, otomatis untuk menyelesaikan masalah ini dengan tepat. Dengan buffer protokol, Anda menulis .protodeskripsi tentang struktur data yang ingin Anda simpan. Dari itu, kompiler buffer protokol menciptakan kelas yang mengimplementasikan pengkodean otomatis dan penguraian data buffer protokol dengan format biner yang efisien. Kelas yang dihasilkan menyediakan getter dan setter untuk bidang yang membentuk buffer protokol dan mengurus detail membaca dan menulis buffer protokol sebagai satu unit. Yang penting, format buffer protokol mendukung gagasan untuk memperluas format dari waktu ke waktu sedemikian rupa sehingga kode masih dapat membaca data yang dikodekan dengan format lama.

### **5.13.2 Instalasi Pillow**

Untuk melakukan instalasi protobuf, buka anaconda command prompt kemudian aktivasi environment yang telah dibuat, dan jalankan perintah berikut “conda install protobuf” (versi protobuf menyesuaikan dengan kebutuhan machine learning yang akan dibuat). Berikut tampilan apabila berhasil menginstal protobuf :

```
(Buku_Tutorial) C:\Windows\System32>conda install protobuf
Collecting package metadata (current_repotdata.json): done
Solving environment: done

==> WARNING: A newer version of conda exists. <==
  current version: 4.7.12
  latest version: 4.8.1

Please update conda by running

$ conda update -n base -c defaults conda

# All requested packages already installed.
```

*Gambar 5. 55 Library prorobuf Berhasil DiInstall*

## **5.14 Library Tensorboard**

### **5.14.1 Pengertian Tensorboard**

TensorBoard, alat visualisasi TensorFlow, sering digunakan oleh para peneliti dan insinyur untuk memvisualisasikan dan memahami eksperimen ML mereka. Ini memungkinkan pelacakan metrik eksperimen, memvisualisasikan model, membuat profil program ML, memvisualisasikan eksperimen penyetelan hyperparameter, dan banyak lagi.

Sementara TensorBoard memudahkan untuk memvisualisasikan eksperimen Anda sendiri, pembelajaran mesin sering melibatkan kolaborasi. Anda mungkin ingin membagikan penelitian Anda tentang efek hyperparameter, menjelaskan prosedur pelatihan yang rumit, atau mendapatkan bantuan pemecahan masalah perilaku model aneh.

Kami telah melihat orang berbagi tangkapan layar TensorBoards mereka untuk mencapai ini. Namun, tangkapan layar tidak

interaktif dan gagal menangkap semua detail. Di Google, peneliti dan insinyur sering mengomunikasikan wawasan mereka tentang perilaku model dengan mengirimkan visualisasi TensorBoard mereka ke rekan tim. Tujuan kami adalah untuk memberikan kemampuan ini kepada komunitas yang lebih luas.

Itu sebabnya kami meluncurkan TensorBoard.dev: layanan terkelola (saat ini dalam pratinjau) yang memungkinkan Anda untuk meng-host, melacak, dan berbagi eksperimen ML dengan mudah secara gratis. Cukup unggah log TensorBoard Anda dan terima tautan yang dapat dilihat oleh semua orang, tanpa instalasi atau pengaturan.

#### **5.14.2 Instalasi Tensor board**

Untuk melakukan instalasi tensorboard, buka anaconda command prompt kemudian aktivasi environment yang telah dibuat, dan jalankan perintah berikut “pip install tensorboard” (versi tensorboard menyesuaikan dengan kebutuhan machine learning yang akan dibuat). Berikut tampilan apabila berhasil menginstal tensorboard :

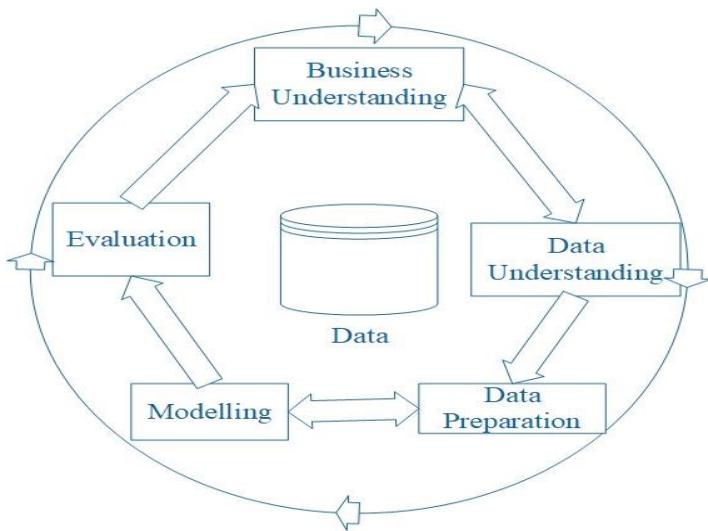
```
(Buku_Tutorial) C:\Windows\System32>pip install tensorboard
WARNING: pip is being invoked by an old script wrapper. This will fail in a future
version of pip.
Please see https://github.com/pypa/pip/issues/5599 for advice on fixing the underlying
issue.
To avoid this problem you can invoke Python with '-m pip' instead of running pip di-
rectly.
Requirement already satisfied: tensorboard in e:\anaconda\envs\buku_tutorial\lib\site-
packages (2.1.0)
Requirement already satisfied: grpcio>=1.24.3 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (1.26.0)
Requirement already satisfied: werkzeug>=0.11.15 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (0.16.1)
Requirement already satisfied: markdown>=2.6.8 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (3.1.1)
Requirement already satisfied: wheel>=0.26; python_version >= "3" in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (0.34.1)
Requirement already satisfied: six>=1.10.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (1.14.0)
Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (0.4.1)
Requirement already satisfied: requests<3,>=2.21.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (2.22.0)
Requirement already satisfied: numpy>=1.12.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (1.18.1)
Requirement already satisfied: setuptools>=41.0.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (45.1.0.post20200119)
Requirement already satisfied: protobuf>=3.6.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (3.11.2)
Requirement already satisfied: google-auth<2,>=1.6.3 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (1.11.0)
Requirement already satisfied: absl-py>=0.4 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from tensorflow) (0.9.0)
Requirement already satisfied: requests-oauthlib>=0.7.0 in e:\anaconda\envs\buku_tutorial\lib\site-
packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorflow) (1.3.
0)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in e:\anacon-
da\envs\buku_tutorial\lib\site-packages (from requests<3,>=2.21.0->tensorflow) (1.
25.8)
```

Gambar 5. 56 Library Tensorboard Berhasil Di install

## 5.15 Metodologi CRISP-DM

### 5.15.1 Pengertian CRISP-DM

Proses data *mining* digunakan dengan tujuan untuk dapat memilih teknik data *mining* apa yang cocok untuk diterapkan, dan digunakan juga sebagai acuan untuk keseluruhan proses penambangan data dari objek bisnis. Saat ini terdapat beberapa metode yang bisa dipakai untuk melakukan data mining, diantaranya model *CRISPDM*, *generic* model dan lain sebagainya. Proses data *mining* yang akan digunakan pada buku ini adalah model referensi *CRISP-DM*, seperti terlihat pada gambar berikut ini.



*Gambar 5. 57 Proses Cross-Industry Process for Data Mining (CRISP-DM)*

Metodologi CRISP-DM adalah standar *data mining* yang disusun oleh tiga tambang penggagas data pasar, yaitu Daimler Chrysler (Daimler-Benz), SPSS (ISL), NCR. Kemudian berkembang dalam berbagai lokakarya (antara 1997-1999). Lebih dari 300 organisasi memberikan kontribusi untuk proses pemodelan ini dan akhirnya CRISP-DM 1.0 diterbitkan pada tahun 1999. Pada gambar 5.57 merupakan model CRISP-DM yang berisikan daur hidup kegiatan *data mining* yang terdiri atas lima fase. Gambar 5.5.7 merupakan keseluruhan tahapan pada model CRISP-DM. salah satu keuntungan dari model ini adalah, tahapan fase dari model bukanlah sebuah tahapan yang kaku. Perpindahan maju dan mundur antara tiap fase yang berbeda bisa selalu dilakukan. Hal ini sesuai dengan sifat alami dari *data mining* itu sendiri, dimana proses *data mining* tidak selesai saat sebuah hasil di temukan, sebab proses *data mining* merupakan sebuah proses pembelajaran terus-menerus.

### **5.1.5.2 Tahapan-Tahapan Cross-Industry Process For Data Mining**

Adapun fase-fase dalam *Cross-Industry Process For Data Mining (CRISP-DM)* adalah sebagai berikut:

- 1. Pemahaman Bisnis (Business Understanding)**

Fase ini merupakan fase awal dari model *CRISP-DM*. Pada tahapan pertama peneliti mencoba untuk memahami permasalahan yang ada dalam pengenalan karakter plat nomor kendaraan Indonesia. Sehingga dapat menentukan tujuan dan pola yang akan didapatkan dengan *data mining*. Adapun permasalahan dari penelitian ini adalah karena pada saat sekarang ini ditempat parkir pada saat kendaraan akan parkir itu masih dicatat secara manual oleh petugas. Kemudian sistem ini akan dikembangkan menjadi *smart parking*, yaitu pencatatan plat nomor kendaraan sudah dilakukan secara otomatis pada saat masuk ke tempat parkir. Untuk menyelesaikan permasalahan tersebut digunakan menggunakan metode *Convolutional Neural Network (CNN)* sehingga kita dapat mengetahui hasil dari pengujian plat berdasarkan karakter pada plat nomor kendaraan baik angka maupun huruf.

- 2. Pemahaman Data (Data Understanding)**

Fase ini dikenal sebagai fase pengumpulan data. Data yang telah dikumpulkan dalam penelitian ini data gambar tentang pengenalan karakter pada plat nomor kendaraan Indonesia. Data yang diolah adalah data gambar plat nomor kendaraan baik plat motor maupun plat mobil. Data yang digunakan dalam penelitian berjumlah 1615 gambar data plat nomor kendaraan Indonesia, dimana data tersebut diambil dari periode bulan september sampai bulan november, jika data yang diperoleh semakin menyelesaikan permasalahan tersebut kita menggunakan metode *Convolutional Neural Network (CNN)* untuk pengenalan karakter pada plat nomor kendaraan Indonesia.

### 3. Pengolahan Data (Data Preparation)

Meliputi semua aktivitas untuk membuat *dataset* final. Pada penelitian mengenai pengenalan karakter pada plat nomor kendaraan Indonesia dengan menggunakan metode *convolutional neural network*, dijelaskan bahwa tahapan dalam proses pengolahan data mencakup semua kegiatan yang membangun *dataset* akhir (data yang akan dimasukkan ke dalam pemodelan) dari data mentah awal. Data preparation mencakup semua kegiatan untuk membangun data set yang akan diolah dalam proses pemodelan dengan menggunakan metode *Convolutional Neural Network (CNN)*. Mulai dari pemberian label pada gambar, hingga proses *training* data dengan menggunakan *tensor flow*.

### 4. Pemodelan (Modelling)

Model adalah aplikasi dari algoritme untuk mencari, mengidentifikasi, dan menampilkan pola. Pada penelitian ini menggunakan metode Convolutional Neural Network sehingga kita dapat melakukan proses pengenalan plat nomor kendaraan Indonesia. Dari data yang telah dipilih pada fase data preparation yaitu data set training dan data set testing.

## 5. Evaluasi

Pada tahap ini, model sudah terbentuk dan diharapkan memiliki kualitas baik jika dari sudut pandang analisa data, Pada tahap ini akan dilakukan evaluasi terhadap keefektifan dan kualitas model sebelum digunakan dan menentukan apakah model dapat mencapai tujuan yang ditetapkan pada fase awal( Business Understanding).

## **BAB 6**

---

# **TUTORIAL OBJEK DETEKSI PENGENALAN KARAKTER PADA PLAT NOMOR KENDARAAN INDONESIA**

---

Sebelum melakukan tutorial penulis akan menjelaskan sedikit pemaparan latar belakang mengapa system objek deteksi plat nomor ini dibuat dan hasil yang diperoleh dari sistem yang sudah berhasil dibuat dan diuji.

Pada saat ini jumlah kendaraan di Indonesia setiap tahunnya terus mengalami peningkatan. Berdasarkan data milik Korps Lalu Lintas Kepolisian Negara Indonesia yang dikutip oleh situs surat kabar Kompas, pada tahun 2013 jumlah kendaraan di Indonesia mencapai 104.211.000 unit, atau meningkat sebesar 11% dibandingkan dengan tahun sebelumnya. Setiap kendaraan baik motor maupun mobil memiliki plat nomor. Plat nomor adalah salah satu bentuk identitas dari setiap kendaraan baik motor maupun mobil. Plat nomor kendaraan terdiri dari kombinasi huruf dan

angka, yang mana setiap huruf dan angka tersebut mengandung informasi tentang kode propinsi dan kode daerah dimana kendaraan tersebut terdaftar. Plat nomor dari tiap negara memiliki model yang berbeda, baik dari warna maupun bentuk susunan karakternya, sehingga pengenalan plat nomor yang dilakukan di negara lain belum tentu bisa mengenali plat nomor yang ada di Indonesia.

Meningkatnya jumlah kendaraan bermotor dan semakin padatnya arus lalu lintas di suatu daerah membuat sistem pengenalan citra *digital* plat nomor kendaraan oleh komputer menjadi penting. Sistem pengenalan citra digital plat nomor kendaraan diperlukan untuk mendukung ketertiban lalu lintas, mengetahui dengan cepat pemilik kendaraan, dan mempermudah pengaturan area parkir. Sistem pengenalan citra *digital* plat nomor kendaraan dapat dilakukan menggunakan teknik pengenalan pola. Penerapan yang menggunakan pengenalan plat nomor salah satunya sebagai input parkir, dimana awalnya pencatatan nomor plat kendaraan menggunakan cara manual sekarang bisa menggunakan gambar sebagai alat input. Teknologi ini secara tidak langsung mengurangi peran manusia dalam sistem parkir. Penelitian mengenai pengenalan karakter pada plat nomor kendaraan terus mengalami perkembangan. Dimulai dengan penelitian untuk deteksi lokasi plat nomor

Sampai dengan pengenalan karakter pada plat nomor. Pada saat sekarang ini perkembangan teknologi bukan hanya pada bidang teknologi saja tetapi juga pada bidang komputerisasi. Pada saat sekarang ini pada bidang komputerisasi salah satunya adalah

computer vision mengalami perkembangan yang sangat pesat. Dalam computer vision terdapat beberapa permasalahan dan salah satunya adalah pengenalan plat nomor kendaraan / Recognition Plate. Recognition Plate dengan deep learning atau jaringan saraf tiruan ini masih berkembang sebagai teknologi untuk menduplikasi kemampuan manusia dalam memahami informasi dari sebuah gambar agar komputer dapat mengenali objek pada gambar selayaknya manusia. Deep learning memiliki kemampuan yang sangat baik dalam visi computer dengan kapabilitas nya yang signifikan dalam memodelkan berbagai data kompleks seperti data gambar. Salah satu metode yaitu metode Deep Learning pada saat ini memiliki hasil paling bagus dalam pengenalan citra adalah Convolutional Neural Network (CNN). Hal itu disebabkan karena metode CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia sehingga memiliki kemampuan mengolah informasi citra. Namun CNN, seperti metode Deep Learning lainnya, memiliki kelemahan yaitu proses pelatihan model yang lama. Salah satu sub tipe jaringan saraf tiruan yang menangani permasalahan computer vision adalah Convolutional Neural Network (CNN). CNN dirancang khusus untuk deep learning yaitu pada bidang image processing. CNN digunakan karena dapat menangani masalah yang kompleks tersebut dan memiliki kinerja yang baik dari hasil penelitian terkait. CNN merupakan variasi dari jaringan saraf tiruan yang memiliki bobot dan beberapa lapisan tersembunyi yang disusun menjadi arsitektur.

CNN memiliki kemampuan klasifikasi yang diperuntukkan untuk pengolahan data gambar. Convolutional Neural Network dapat kita gunakan untuk proses klasifikasi data yang sudah terlabel dengan menggunakan metode supervised learning, dimana cara kerja dari supervised learning adalah sudah terdapat data yang akan dilatih dan terdapat variabel yang sudah kita targetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada. CNN sering digunakan untuk mengenali benda atau pemandangan, dan melakukan deteksi dan segmentasi objek.

Berdasarkan uraian diatas, maka penulis ingin membuat sistem untuk mendeteksi pengenalan karakter pada plat nomor kendaraan Indonesia. Adapun metode yang digunakan pada sistem tahap pengenalan karakter plat nomor kendaraan ini adalah metode Convolutional Neural Network (CNN). Dimulai dengan penelitian untuk deteksi lokasi plat nomor sampai dengan pengenalan karakter pada plat nomor kendaraan.

### **6.1 Mempersiapkan Tools**

Pada tutorial ini saya menggunakan TensorFlow-GPU sebagai ganti TensorFlow biasa untuk mengurangi waktu pelatihan dengan faktor sekitar 8 (3 jam untuk melatih). Versi CPU-only dari TensorFlow juga dapat digunakan untuk tutorial ini, tetapi akan memakan waktu lebih lama.

Untuk membuat system objek deteksi maka diperlukan beberapa tools yang dapat mendukung dalam pembuatannya dan tools ini

sudah dibahas pada bab sebelumnya, berikut tools yang harus dipersiapkan dalam pembuatan system object deteksi :

- Anaconda
- LabelImg
- CUDA
- CUDNN

Jika Anda menggunakan versi TensorFlow yang lebih lama, pastikan Anda menggunakan versi CUDA dan cuDNN yang kompatibel dengan versi TensorFlow yang Anda gunakan. Versi Anaconda saat ini menggunakan Python 3.7, yang tidak secara resmi didukung oleh TensorFlow.

## **6.2 Mengatur Direktori Tensor Flow dan Lingkungan Virtual Anaconda**

API Deteksi Objek TensorFlow membutuhkan penggunaan struktur direktori khusus yang disediakan dalam repositori GitHubnya. Ini juga memerlukan beberapa paket Python tambahan, tambahan spesifik untuk variabel PATH dan PYTHONPATH, dan beberapa perintah pengaturan tambahan untuk mengatur semuanya agar berjalan atau melatih model deteksi objek. Bagian tutorial ini membahas pengaturan lengkap yang diperlukan. Ini cukup teliti, tetapi ikuti instruksi dengan seksama, karena pengaturan yang tidak tepat dapat menyebabkan kesalahan berat di jalan.

### **6.2.1 Unduh repository API Deteksi Objek TensorFlow dari GitHub**

Buat folder langsung di C: dan beri nama "tensorflow3". Direktori kerja ini akan berisi kerangka kerja

pendeksi objek TensorFlow penuh, serta gambar pelatihan, data pelatihan, classifier terlatih, file konfigurasi, dan segala sesuatu yang diperlukan untuk classifier deteksi objek. Unduh repositori deteksi objek TensorFlow lengkap yang terdapat di <https://github.com/tensorflow/models> dengan mengeklik tombol “Kloning atau Unduh” dan mengunduh file zip. Buka file zip yang diunduh dan ekstrak folder "model-master" langsung ke direktori C: \ tensorflow3 yang baru saja Anda buat. Ganti nama "model-master" menjadi "model".

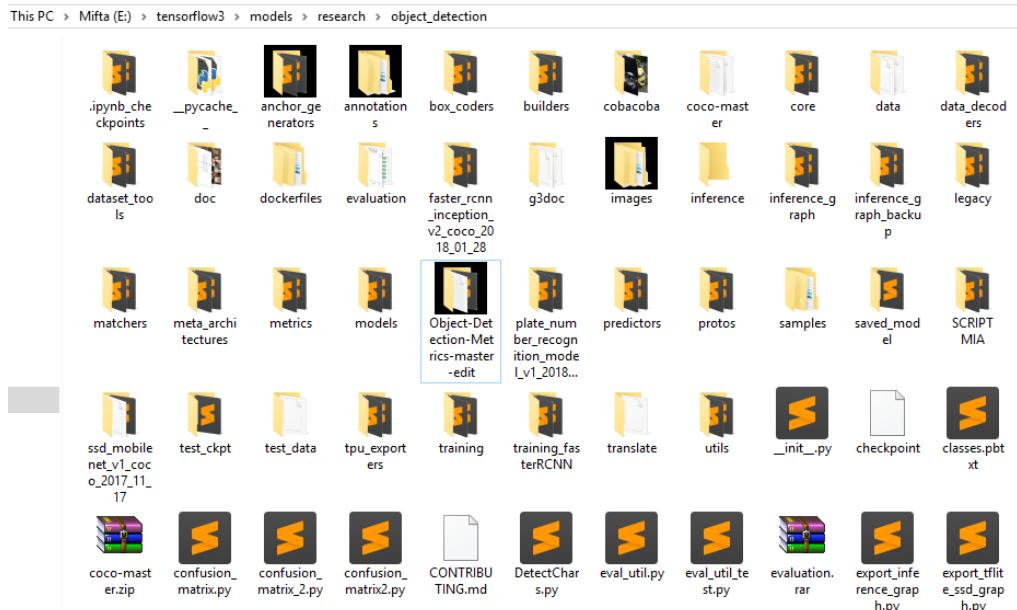
### **6.2.2 Unduh model Faster-RCNN-Inception**

TensorFlow menyediakan beberapa model deteksi objek (pengklasifikasi pra-terlatih dengan arsitektur jaringan saraf. Tutorial ini akan menggunakan model Faster-RCNN-Inception-V2. Unduh modelnya di sini. Buka file fast\_rcnn\_inception\_v2\_coco\_2018\_01\_28.tar.gz yang diunduh dengan pengarsipan file seperti WinZip atau 7-Zip dan ekstrak folder fast\_rcnn\_inception\_v2\_coco\_2018\_01\_28 ke folder E:\tensorflow3\models\research\object\_detection. (Catatan: Tanggal dan versi model kemungkinan akan berubah di masa mendatang, tetapi masih harus bekerja dengan tutorial ini.).

### **6.2.3 Unduh repositori tutorial ini dari GitHub**

Unduh repositori lengkap yang terdapat di halaman ini [“https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10/tree/b549e98d96f2575a3e408c7234846653aec5b090”](https://github.com/EdjeElectronics/TensorFlow-Object-Detection-API-Tutorial-Train-Multiple-Objects-Windows-10/tree/b549e98d96f2575a3e408c7234846653aec5b090) (gulir ke atas dan klik Klon atau Unduh) dan ekstrak semua konten langsung

ke direktori E:\tensorflow3\models\research\object\_detection. (Anda dapat menimpa file "README.md" yang ada.) Ini menetapkan struktur direktori tertentu yang akan digunakan untuk siswa tutorial. Pada titik ini, inilah tampilan folder \ object\_detection Anda:



Gambar 6. 1 Repository Object Detection

Pada gambar 6.1 merupakan repository object detection. Reposisori ini berisi gambar, data anotasi, file .csv, dan TFRecords yang diperlukan untuk melatih detektor plate. Anda dapat menggunakan gambar dan data ini untuk berlatih membuat detector plate Anda sendiri. Jika Anda ingin melatih detektor objek Anda sendiri, hapus file-file berikut (jangan hapus folder):

- Semua file dalam \object\_detection\images\train and \object\_detection\images\test
- File "test\_labels.csv" dan "train\_labels.csv" di \object\_detection\images

- Semua file dalam \object\_detection\training
- Semua file di \object\_detection\inference\_graph

Sekarang, Anda siap untuk memulai dari awal dalam melatih detektor objek Anda sendiri. Tutorial ini akan mengasumsikan bahwa semua file yang tercantum di atas telah dihapus, dan selanjutnya akan menjelaskan cara membuat file untuk dataset pelatihan Anda sendiri.

#### **6.2.4 Pengaturan Anaconda Virtual Environment**

Selanjutnya melakukan pengaturan path di Anaconda untuk tensorflow-gpu. Dari menu Start di Windows, cari utilitas Anaconda Prompt, klik kanan padanya, dan klik "Run as Administrator". Jika Windows bertanya apakah Anda ingin mengizinkannya untuk melakukan perubahan pada komputer Anda, klik ya.

- Pada anaconda command prompt, buat environment dengan nama "tensorflow1" dengan menggunakan perintah berikut :  
“Conda create -n tensorflow1 pip python=3.6”
- Kemudian, aktifkan environment dan perbarui pip dengan menjalankan perintah berikut:  
“Activate tensorflow3”  
“Python -m pip install --upgrade pip”
- Instal tensorflow-gpu di lingkungan ini dengan perintah:  
“Pip install --ignore-installed --upgrade tensorflow-gpu”
- Instal paket-paket lain yang diperlukan dengan mengeluarkan perintah berikut:
  - conda install -c anaconda protobuf
  - pip install pillow

- pip install lxml
- pip install Cython
- pip install contextlib2
- pip install jupyter
- pip install matplotlib
- pip install pandas
- pip install opencv-python

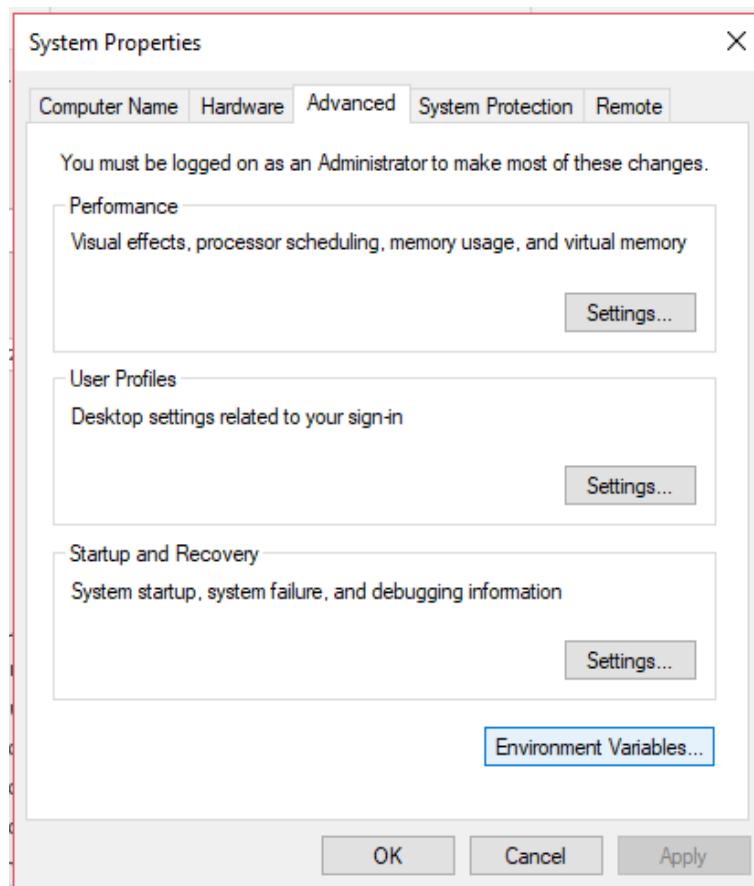
Catatan: Package 'pandas' dan 'opencv-python' tidak diperlukan oleh TensorFlow, tetapi mereka digunakan dalam skrip Python untuk menghasilkan TFRecords dan untuk bekerja dengan gambar.

#### **6.2.5 Konfigurasikan PYTHONPATH environment variable**

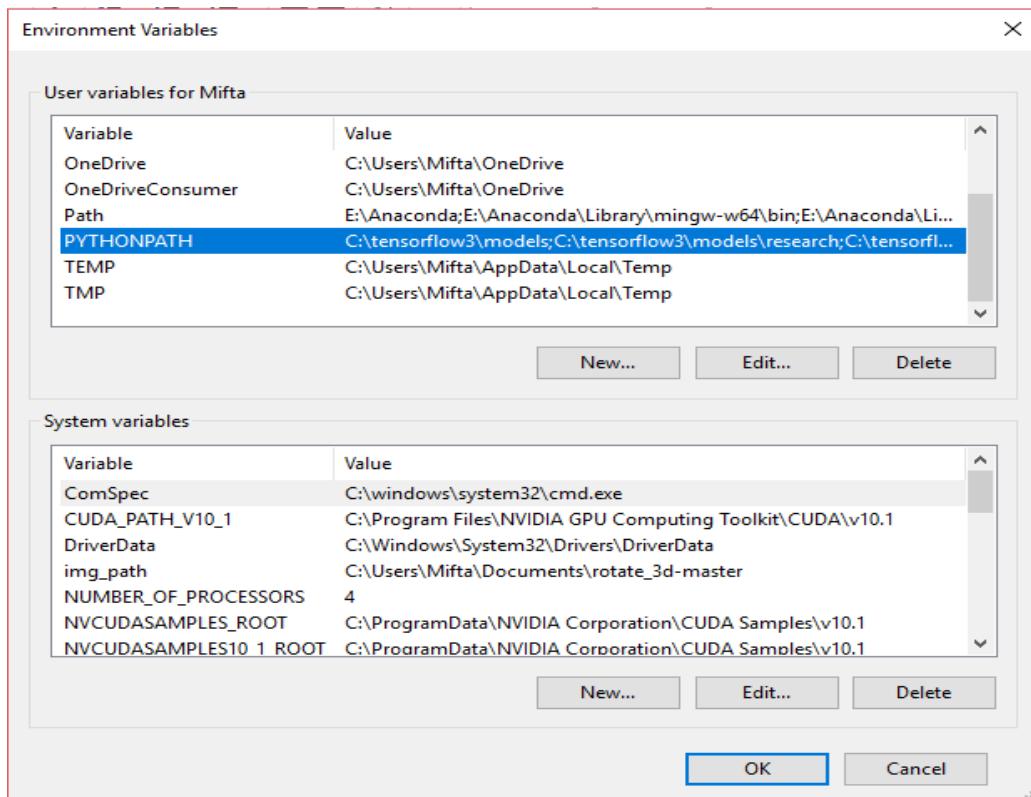
Variabel PYTHONPATH harus dibuat yang menunjuk ke direktori \ models, \ models \ research, dan \ models \ research \ slim. Lakukan ini dengan mengeluarkan perintah berikut (dari direktori mana saja):

```
(tensorflow3) E:\> set PYTHONPATH=C:\tensorflow3\models;C  
:\tensorflow3  
\models\research;C:\tensorflow3\models\research\slim
```

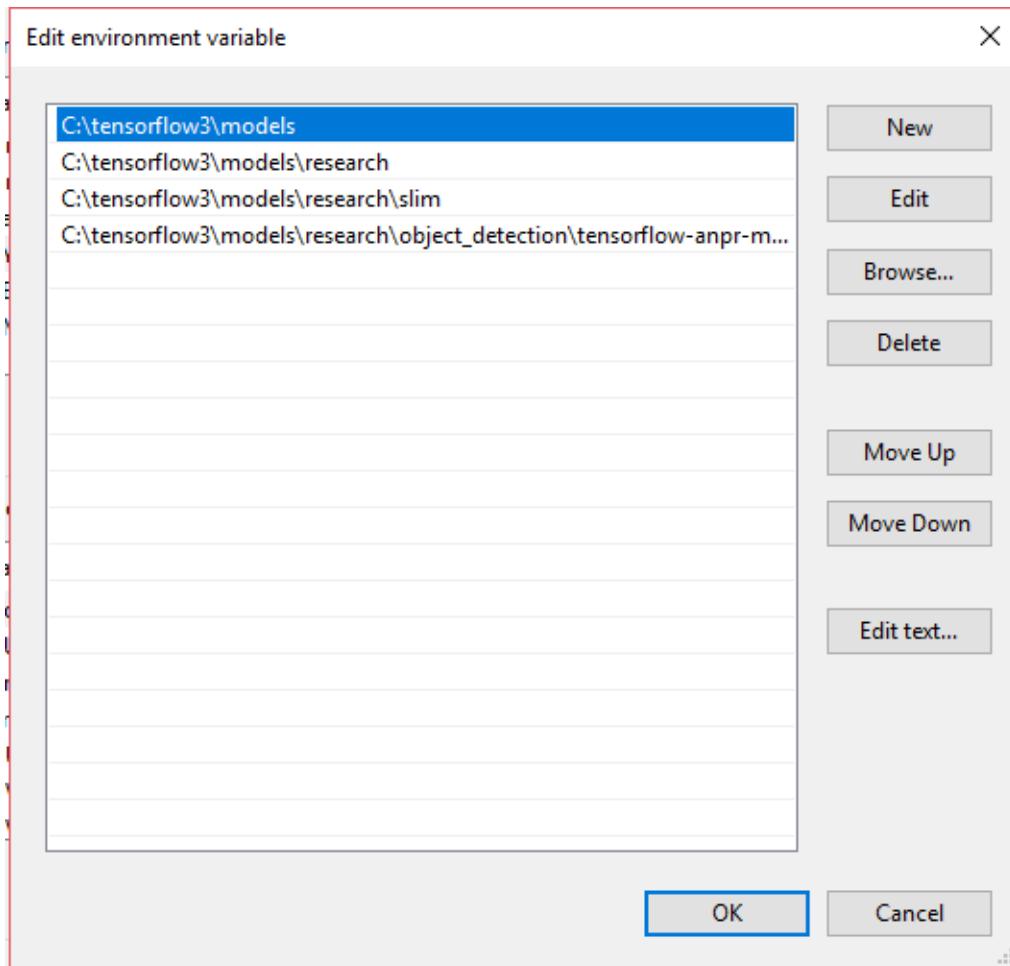
Catatan: Setiap kali lingkungan virtual "tensorflow3" keluar, variabel PYTHONPATH diatur ulang dan perlu disetel lagi. Anda dapat menggunakan "echo% PYTHONPATH%" untuk melihat apakah telah disetel atau tidak. Berikut contoh pengaturan pythonpath untuk objek deteksi:



Gambar 6. 2 Environment Variabel



Gambar 6. 3 Environment Variabel



Gambar 6. 4 Python Path Anaconda

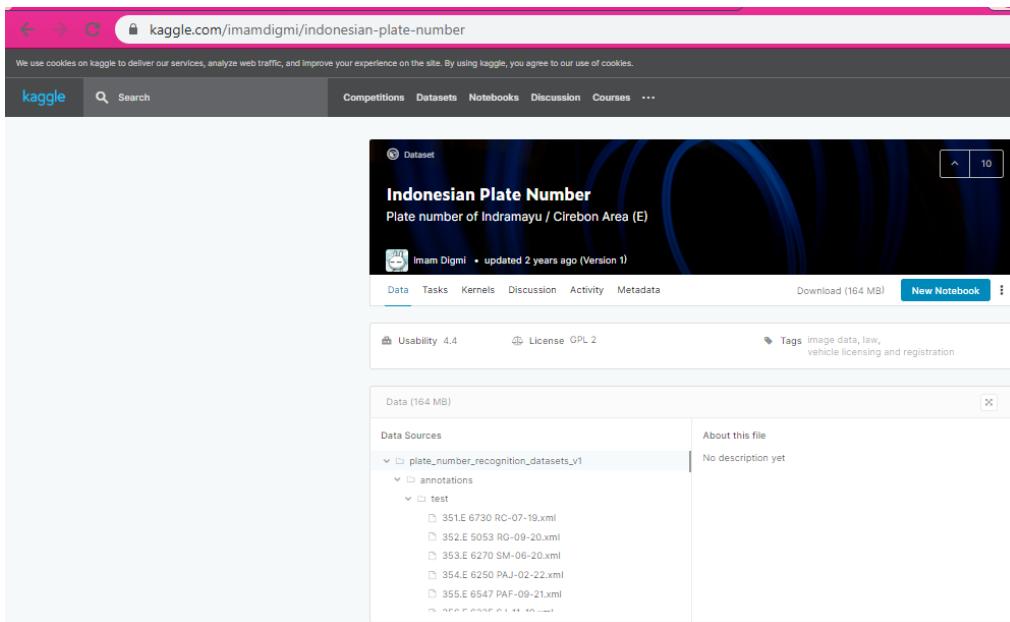
### 6.3 Pemahaman Bisnis

Pada saat sekarang ini masih banyak terjadi *ditempat parkir*, dimana awalnya pencatatan nomor plat kendaraan menggunakan cara manual sekarang bisa menggunakan gambar sebagai alat input. Teknologi ini secara tidak langsung mengurangi peran manusia dalam sistem parkir. Penelitian mengenai pengenalan karakter pada plat nomor kendaraan terus mengalami perkembangan. Dimulai dengan penelitian untuk deteksi lokasi plat nomor sampai dengan pengenalan karakter pada plat nomor. Metode *Deep Learning* saat

ini memiliki hasil paling berarti dalam pengenalan citra adalah *Convolutional Neural Network* (CNN). Hal tersebut karena metode CNN berusaha meniru sistem pengenalan citra pada *visual cortex* pada manusia sehingga mempunyai kemampuan mengerjakan informasi citra.

#### **6.4 Pemahaman Data/Pengumpulan Data**

Jika kita ingin mengenali objek “khusus” yang ingin kita deteksi, tentu kita memerlukan dataset untuk proses training, sehingga Neural Network yang akan kita latih untuk mengenali objek tersebut dapat mengenali objek yang kita maksud. Anda dapat menggunakan telepon Anda untuk mengambil gambar objek atau mengunduh gambar objek dari Google Image Search. Apabila anda tidak ingin membuat data set maka anda dapat menggunakan dataset penulis pada link github yang sudah dipaparkan sebelumnya atau mendapatkan dataser dari repository online yaitu di situs Kaggle dengan link berikut “<https://www.kaggle.com/imamdigmi/indonesian-plate-number>”.



Gambar 6. 5 Dataset Kaggle

*Tensorflow* membutuhkan ratusan gambar suatu objek untuk melatih pengenalan karakter plat yang baik. Untuk melatih pengklasifikasi yang kuat, gambar pelatihan harus memiliki objek acak dalam gambar bersama dengan objek yang diinginkan, dan harus memiliki berbagai latar belakang dan kondisi pencahayaan.

Pengumpulan data yang digunakan peneliti merupakan data berupa gambar. Data tersebut meliputi data gambar plat nomor kendaraan Indonesia yang diambil langsung dari *kaggle* dan data yang diambil langsung dari tempat parkir dan jalanan oleh peneliti dengan menggunakan *smartphone*. Data tersebut dapat dilihat pada tabel 6.1 sebagai berikut:

Tabel 6. 1 Data Awal

No.	Sumber Data	Jenis Data	Lokasi Data	Jumlah Data	Format Data
1.	Kaggle [53]	Data <i>Training</i>	Kaggle.com	470	JPG
2.	Data dari Internet [54]	Data <i>Training</i>	Google	100	JPG
2.	Data diambil dengan <i>Smartphone</i>	Gambar <i>Training</i>	Tempat parkir dan jalan	939	JPG
3.	Data diambil dengan <i>Smartphone</i>	Data <i>Test</i>	Tempat parkir dan jalan	106	JPG

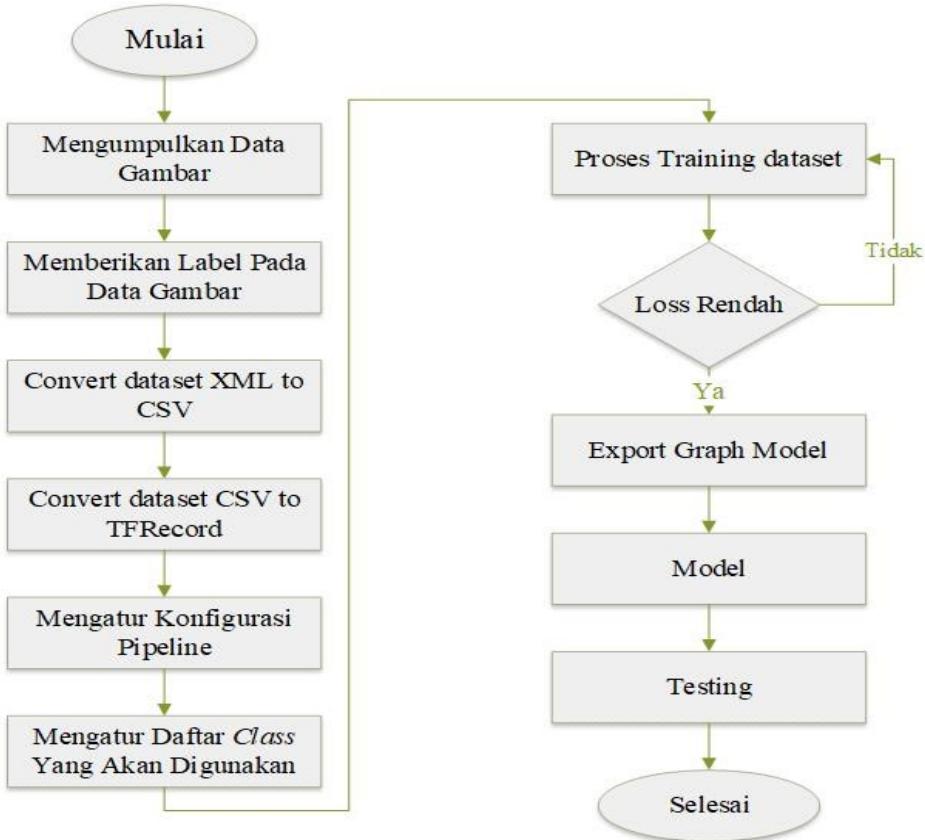


Gambar 6. 6 Contoh Dataset Plate Nomor Kendaraan Indonesia

Berdasarkan data pada tabel 5.1 diatas, data yang digunakan berasal dari *kaggle* dengan jumlah data sebanyak 470 gambar untuk data *training*, data yang berasal dari internet sebanyak 100 data gambar, data yang diambil langsung dengan smartphone peneliti dengan jumlah data sebanyak 939 gambar untuk data *training* dan 106 data gambar yang diambil dengan menggunakan *smartphone* peneliti yang akan digunakan untuk data *test*. Dari data tersebut didapatkan total jumlah data yang digunakan pada penelitian adalah sebanyak 1615 gambar.

## 6.5 Persiapan Data

Sebelum masuk ke tahap selanjutnya, penulis akan menjelaskan terlebih dahulu alur dari penelitian mengenai pengenalan karakter pada plat nomor kendaraan Indonesia sebagai berikut:



Gambar 6. 7 Diagram Alur Penelitian

Pada gambar 6.7 merupakan diagram alur penelitian pada pengenalan karakter plat nomor, dijelaskan tahapan peneliti dari mendapatkan data awal hingga terbentuknya *model* sehingga dapat dilakukan proses *testing* data plat nomor kendaraan Indonesia. Dimana tahapan awalnya adalah penulis mengumpulkan data gambar plat nomor kendaraan Indonesia terlebih dahulu, jumlah *dataset* yang telah peneliti

kumpulkan adalah sebanyak 1615 gambar plat nomor kendaraan, setelah penulis mendapatkan *dataset* gambar, selanjutnya akan masuk ke tahap selanjutnya yaitu proses memberi label pada data gambar, menggunakan *tools LabelImg* yang sudah

dijelaskan pada bab 2.18, pada proses label gambar tersebut akan menghasilkan file dengan format *xml*. Setelah berhasil memberi label pada gambar, langkah selanjutnya adalah *convert* data dari format *xml* menjadi format *csv*. Setelah berhasil mengubah file dengan format *csv*, selanjutnya adalah *convert* data dari *file* dengan format *csv* menjadi *file* dengan format *record*. Kemudian langkah selanjutnya adalah konfigurasi *object detection training pipeline* untuk mengonfigurasi proses pelatihan dan evaluasi yang dilakukan menggunakan berkas *protobuf*. Kemudian dilakukan mengatur daftar *class* yang digunakan dan *dataset (file TFRecord)* yang sesuai dengan proses pelabelan data. Setelah berhasil kita masuk ke tahapan selanjutnya yaitu proses *training dataset*, setelah berhasil melakukan proses *training dataset*, apabila *loss/error* nya rendah kita bisa ke tahapan selanjutnya yaitu proses *export* untuk mendapatkan model yang kita inginkan, apabila *error/loss* nya masih tinggi maka akan kita lakukan proses *training* lagi dan akan dilakukan tahap evaluasi data. Selanjutnya kita sudah dapat melakukan proses *testing* data karena kita sudah mendapatkan model yang kita inginkan.

Untuk *source code* mengenai deteksi karakter pada plat nomor kendaraan Indonesia ini bisa kita dapatkan melalui *GitHub*. Peneliti menggunakan *source code* dari Steve fielding mengenai *Automatic Number (License) Plate Recognition* yang sudah dimodifikasi disesuaikan dengan kepentingan dan keperluan oleh peneliti. Untuk link *GitHub* dari *Automatic Number (License) Plate*

*Recognition* yaitu “<https://github.com/stevefielding/tensorflow-anpr>”.

### 6.5.1 Proses LabelImg

Pada proses ini, setelah gambar nya dikumpulkan pada 1 *Folder*, kemudian peneliti melakukan proses *labelling* gambar dengan menggunakan *tools LabelImg* yang sudah dijelaskan pada bab sebelumnya. *LabelImg* adalah alat untuk memberi label pada gambar, data yang dihasilkan oleh *labelimg* ini adalah titik koordinat gambar, anotasi gambar plat nomor kendaraan dimana format yang dihasilkan dari proses memberikan label pada data gambar adalah berupa .xml. *LabelImg* menyimpan *file* .xml yang berisi data label untuk setiap gambar. *File* .xml ini akan digunakan untuk menghasilkan *TFRecords*, yang merupakan salah satu input untuk pelatih *Tensorflow*. Berikut adalah proses label *image* pada *dataset* plat nomor kendaraan:



Gambar 6. 8 proses label *image* pada *dataset* plat nomor kendaraan Indonesia

*Tabel 6. 2 Hasil dari proses label image pada dataset plat nomor kendaraan Indonesia*

<p>File IMG20191109125914_fix.png</p> <pre>&lt;?xml version="1.0" ?&gt;  &lt;annotation&gt;     &lt;folder&gt;None&lt;/folder&gt;     &lt;filename&gt;image_3344.jpg&lt;/filename&gt;     &lt;path&gt;dataset_train_fix/image_3344.jpg&lt;/path&gt;     &lt;source&gt;         &lt;database&gt;Unknown&lt;/database&gt;     &lt;/source&gt;     &lt;size&gt;         &lt;width&gt;447&lt;/width&gt;         &lt;height&gt;193&lt;/height&gt;         &lt;depth&gt;3&lt;/depth&gt;     &lt;/size&gt;     &lt;segmented/&gt;     &lt;object&gt;         &lt;name&gt;E&lt;/name&gt;         &lt;pose&gt;Unspecified&lt;/pose&gt;         &lt;truncated&gt;0&lt;/truncated&gt;         &lt;difficult&gt;0&lt;/difficult&gt;         &lt;bndbox&gt;             &lt;xmin&gt;20&lt;/xmin&gt;             &lt;ymin&gt;22&lt;/ymin&gt;             &lt;xmax&gt;59&lt;/xmax&gt;             &lt;ymax&gt;97&lt;/ymax&gt;         &lt;/bndbox&gt;     &lt;/object&gt; &lt;/annotation&gt;</pre>
--

```
</object>

<object>

<name>4</name>

<pose>Unspecified</pose>

<truncated>0</truncated>

<difficult>0</difficult>

<bndbox>

<xmin>117</xmin>

<ymin>19</ymin>

<xmax>161</xmax>

<ymax>97</ymax>

</bndbox>

</object>

<object>

<name>1</name>

<pose>Unspecified</pose>

<truncated>0</truncated>

<difficult>0</difficult>

<bndbox>

<xmin>168</xmin>

<ymin>17</ymin>

<xmax>196</xmax>

<ymax>97</ymax>

</bndbox>

</object>

<object>

<name>1</name>

<pose>Unspecified</pose>
```

```
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>207</xmin>
<ymin>14</ymin>
<xmax>233</xmax>
<ymax>101</ymax>
</bndbox>
</object>
<object>
<name>2</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>241</xmin>
<ymin>19</ymin>
<xmax>282</xmax>
<ymax>99</ymax>
</bndbox>
</object>
<object>
<name>R</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>355</xmin>
```

```
<ymin>13</ymin>
<xmax>391</xmax>
<ymax>97</ymax>
</bndbox>
</object>
<object>
<name>J</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<difficult>0</difficult>
<bndbox>
<xmin>391</xmin>
<ymin>20</ymin>
<xmax>427</xmax>
<ymax>94</ymax>
</bndbox>
</object>
</annotation>
```

### 6.5.2 Mengubah XML menjadi CSV

Pada tahap ini kita akan mengubah *file* XML menjadi *file* CSV agar bisa melanjutkan ke tahap berikutnya.

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python xml_to_csv.py
Successfully converted xml to csv.
```

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python xml_to_csv.py
Successfully converted xml to csv.
```

*Gambar 6. 9 proses convert data .xml to .csv beserta hasilnya*

Gambar 6.9 merupakan source code dimana mengubah file xml menjadi csv pada data *train* dan data *test*, dimana source code tersebut kita jalankan menggunakan python. Berikut gambar 6.10 adalah hasil dari proses *convert xml to csv*.

image_3344.jpg	447	193 E	20	22	59	97
image_3344.jpg	447	193 4	117	19	161	97
image_3344.jpg	447	193 1	168	17	196	97
image_3344.jpg	447	193 1	207	14	233	101
image_3344.jpg	447	193 2	241	19	282	99
image_3344.jpg	447	193 R	355	13	391	97
image_3344.jpg	447	193 J	391	20	427	94

Gambar 6. 10 Proses Convert XML To CSV

### 6.5.3 Membuat *TFRecord*

Selanjutnya kita akan menghasilkan file *TFRecord*, yang nantinya akan menghasilkan file *train.record* dan file *test.record*. Jika sudah mengubah data *train* dan data *test* dalam bentuk CSV maka selanjutnya adalah mengubah file csv menjadi file yang dapat dibaca oleh tensorflow yaitu *TFRecord*. Ini akan digunakan untuk melatih *classifier* deteksi pengenalan karakter pada plat nomor kendaraan Indonesia. Lalu untuk menjalankan perintah generate *TFRecord* nya kita jalankan script *generate\_train\_test\_record.py*

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python generate_tfrecord.py
--csv_input=data\train_labels.csv --type=train --output_path=train.record
C:\Users\Mifta\Anaconda3\envs\tensorflow\lib\site-packages\h5py\_init_.py:72: UserWarning: h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may cause
problems
    '{0}.{1}.{2}'.format(*version.hdf5_builtin_version_tuple)
Successfully created the TFRecords: E:\tensorflow3\models\research\object_detection\train
.record
```

Gambar 6. 11 proses *TFRecord* beserta hasil data *train.record*

```
(tensorflow) E:\tensorflow3\models\research\object_detection>python generate_tfrecord.py  
--csv_input=data\test_labels.csv --type=test --output_path=test.record  
C:\Users\Mifta\Anaconda3\envs\tensorflow\lib\site-packages\h5py\__init__.py:72: UserWarning:  
  h5py is running against HDF5 1.10.2 when it was built against 1.10.3, this may cause  
  problems  
  '{0}.{1}.{2}'.format(*version.hdf5_built_version_tuple)  
Successfully created the TFRecords: E:\tensorflow3\models\research\object_detection\test.  
record
```

*Gambar 6. 12 proses TFRecord beserta hasil data test.record*