

BUKU TUTORIAL OBJECT DETECTION

DETEKSI OBJEK DAN PENGENALAN KARAKTER PLAT NOMOR KENDARAAN INDONESIA MENGGUNAKAN METODE *CONVOLUTIONAL NEURAL NETWORK (CNN)* 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. Hedi Krishna selaku pembimbing external di PT.Telkomunikasi Indonesia Tbk.
4. Nisa Hanum Harani, S.Kom., M.T. selaku dosen pembimbing internship I.
5. Nisa Hanum Harani, S.Kom., M.T. selaku Koordinator Internship I.
6. M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Prodi DIV Teknik Informatika.

Akhir kata, penulis berharap Tuhan Yang Maha Esa berkenan membalaq 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.....	ii
DAFTAR GAMBAR.....	iv
BAB I.....	1
1.1 PENGENALAN AI, ML, NN, DAN DL.....	1
1.1.1 Pengantar <i>Artificial Intelligence</i>	1
1.1.1.1 Definisi ArtificialIntelligence	1
1.1.1.2 Sejarah ArtificialIntelligence	4
1.1.1.3 Konsep Dasar ArtificialIntelligence (Kecerdasan Buatan)	6
1.1.1.4 Disiplin Ilmu ArtificialIntelligence.....	7
1.1.1.5 Manfaat Artificial Intelligence.....	8
1.1.1.6 Kelebihan Artificial Intelligence.....	10
1.1.1.7 Kelemahan Artificial Intelligence.....	11
1.1.2 Pengantar <i>Machine Learning</i>	11
1.1.2.1 Definisi Machine Learning.....	11
1.1.2.2 Sejarah Machine Learning.....	13
1.1.2.3 Perkembangan Machine Learning	14
1.1.2.4 Bagaimana Machine Learning Bekerja?	15
1.1.2.5 Konsep Dasar Machine Learning	17
1.1.2.6 Penerapan Machine Learning.....	19
1.1.2.7 Metode Algoritma Machine Learning.....	21
1.1.2.8 Dampak Machine Learning di Masyarakat	22
1.1.3 Pengantar <i>Neural Network</i>	23
1.1.3.1 Definisi Neural Network.....	23
1.1.3.2 Sejarah Neural Network.....	24
1.1.3.3 Konsep Neural Network	26
1.1.3.4 Algoritma Dalam Neural Network	29
1.1.3.5 Arsitektur Neural Network	30

1.1.4 Pengantar <i>Deep Learning</i>	31
1.1.4.1 Definisi Deep Learning	31
1.1.4.2 Sejarah Deep Learning	32
1.1.4.3 Perbedaan Neural Network Dengan Deep Learning	33
1.1.4.4 Jenis-Jenis Algorithma Deep Learning.....	34
BAB II	36
2.1 Pengenalan CNN, FAST R-CNN, Faster R-CNN . Error! Bookmark not defined.	
2.1.1 Pengantar <i>Convolutional Neural Network</i> (CNN).....	36
2.1.1.1 Definisi Convolutional Neural Network.....	36
2.1.1.2 Sejarah Convolutional Neural Network.....	39
2.1.1.2.1 Input Layer.....	41
2.1.1.2.2 Convolution Layer	41
2.1.1.2.3 Pooling Layer	45
2.1.1.2.4 Activation Function Layer	46
2.1.2 Pengantar Region - Convolution Neural Network (R-CNN)	47
2.1.2.1 Region Proposal.....	47
2.1.2.2 Masalah Dengan R-CNN	49
2.1.3 Pengantar Fast Region - Convolution Neural Network (Fater R-CNN)	
50	
2.1.4 Faster Region - Convolution Neural Network (Faster R-CNN).....	52
BAB III.....	58
3.1 PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN	58
3.1.1 Tools Yang Digunakan	58
3.1.1.1 Anaconda	58
3.1.1.2 LabelImg	62
3.1.1.3 Sublime.....	63
3.1.1.4 CUDA	66
3.1.2 Bahasa Pemrograman Yang Digunakan.....	69
3.2.1.1 Python	69

DAFTAR GAMBAR

Gambar 1. 1 <i>Perceptron</i>	25
Gambar 1. 2 Struktur <i>Neuron</i> pada otak manusia.....	26
Gambar 1. 3 Struktur ANN.....	27
Gambar 1. 4 Arsitektur <i>Neural Network</i>	30
Gambar 1.5 Perbedaan antara lapisan <i>layer</i> pada Jaringan Saraf Tiruan(<i>Neural Network</i>) dengan Jaringan <i>Deep Learning</i>	34
Gambar 2. 1 Arsitektur MLP Sederhana.....	36
Gambar 2. 2 Proses Konvolusi pada CNN.....	37
Gambar 2. 3 Perbedaan arsitektur antara <i>Neural Network</i> pada umumnya dengan CNN.....	38
Gambar 2. 4 Arsitektur CNN	40
Gambar 2. 5 Ilustrasi <i>Input RGB</i>	41
Gambar 2. 6 Image RGB <i>Convolutional Layer</i>	42
Gambar 2. 7 Operasi <i>Convolution Matrix dot</i>	43
Gambar 2. 8 Zero Padding pada Matrix 4x4 menjadi Matrix 6x6	44
Gambar 2. 9 <i>Maximum Pooling Layer</i>	45
Gambar 2. 10 <i>Average pooling layer</i>	46
Gambar 2. 11 Rumus ReLU.....	47

BAB I

PENGENALAN ARTIFICIAL INTELLIGENCE, MACHINE LEARNING, NEURAL NETWORK, DAN DEEP LEARNING

Artificial Intelligence, Neural Network, Machine Learning dan *Deep Learning* adalah istilah dalam dunia teknologi informasi yang sangat populer saat ini. Pada buku ini penulis akan memaparkan materi tentang teknologi tersebut dengan penerapan teknologi *Deep Learning* yaitu *object detection* dan pengenalan karakter pada plat nomor kendaraan Indonesia berbasis *python*.

1.1 Pengantar Artificial Intelligence

1.1.1 Definisi Artificial Intelligence

Kecerdasan buatan atau biasa disebut dengan *Artificial Intelligence* adalah kecerdasan yang ditambahkan kepada suatu sistem yang bisa diatur dalam konteks ilmiah atau bisa disebut juga intelegensi artifisial (bahasa Inggris: *Artificial Intelligence*) atau hanya disingkat AI, didefinisikan sebagai kecerdasan entitas ilmiah. Andreas Kaplan dan Michael Haenlein mendefinisikan kecerdasan buatan sebagai “kemampuan sistem untuk menafsirkan data eksternal dengan benar, untuk belajar dari data tersebut, dan menggunakan pembelajaran tersebut guna mencapai tujuan dan tugas tertentu melalui adaptasi yang fleksibel”.[1]

Berikut ini adalah definisi kecerdasan buatan menurut para ahli:

- **John McCarthy, 1956**

Kecerdasan buatan adalah usaha memodelkan proses berpikir manusia dan mendesain mesin agar dapat menirukan perilaku manusia.

- **Herbert Simon, 1987**

Kecerdasan buatan adalah tempat suatu penelitian, aplikasi dan instrukksi yang terkait dengan pemrograman komputer dalam melakukan suatu hal yang menurut pandangan manusia – cerdas.

- **Rich dan Knight, 1991**

Kecerdasan buatan adalah suatu studi mengenai bagaimana membuat komputer mampu melakukan hal-hal yang pada saat ini masih bisa dilakukan lebih baik oleh manusia.[2]

Pengertian lain dari kecerdasan buatan adalah bagian ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik yang dilakukan manusia. Pada awalnya komputer diciptakan hanya difungsikan sebagai alat hitung saja. Namun seiring berkembangnya zaman, maka peran komputer semakin mendominasi kehidupan manusia. Saat ini komputer dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. Agar komputer bisa bertindak seperti manusia maka komputer juga harus diberi bekal pengetahuan dan mempunyai kemampuan untuk bernalar.[3]

Sistem ini umumnya dianggap komputer. Kecerdasan diciptakan kemudian dimasukkan ke dalam sebuah mesin (komputer) supaya bisa mengerjakan pekerjaan seperti pekerjaan yang dilakukan oleh manusia. Beberapa jenis bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), logika *fuzzy*, jaringan saraf tiruan dan robotika.

Banyak hal yang kelihatannya sulit untuk kecerdasan manusia, tetapi untuk Informatika relatif tidak bermasalah. Contohnya mentransformasikan persamaan, menyelesaikan persamaan integral, membuat permainan catur atau Backgammon. Di lain sisi hal yang bagi manusia kelihatannya menuntut sedikit kecerdasan, sampai sekarang masih sulit untuk direalisasikan dalam Informatika. Seperti contoh: Pengenalan Objek/Muka, bermain sepak bola. Teknologi AI menciptakan cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin. [1]

Secara garis besar, *Artificial Intelligence* (AI) terbagi dalam dua paham pemikiran yaitu AI konvensional dan kecerdasan komputasional. AI konvesional kebanyakan melibatkan metode-metode yang sekarang diklasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalism dan analisis statistic. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama. Metode-metodenya meliputi:

1. Sistem pakar: menerapkan kapabilitas pertimbangan untuk mencapai kesimpulan. Sebuah sistem pakar dapat memproses sejumlah besar informasi yang diketahui dan menyediakan kesimpulan-kesimpulan berdasarkan pada informasi-informasi tersebut.
2. Pertimbangan berdasar kasus
3. Jaringan Bayesian
4. AI berdasar tingkah laku: metode modular pada pembentukan sistem AI secara manual

Kecerdasan komputasional melibatkan pengembangan atau pembelajaran iteratif, misalnya penalaran parameter seperti dalam sistem koneksi. Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis. AI yang tak teratur dan perhitungan lunak. Metode-metode pokoknya meliputi:

1. Jaringan saraf: sistem dengan kemampuan pengenalan pola yang sangat kuat
2. Sistem fuzzy: teknik-teknik untuk pertimbangan dibawah ketidakpastian, teknik digunakan secara meluas dalam industri modern dan sistem kendali produk konsumen
3. Komputasi evolusioner: menerapkan konsep-konsep yang terinspirasi [3]

Walaupun AI memiliki konotasi fiksi ilmiah yang kuat, AI membentuk cabang yang sangat penting pada ilmu komputer, berhubungan dengan perilaku, pembelajaran dan adaptasi yang cerdas dalam sebuah mesin.

Penelitian pada AI melibatkan pembuatan mesin berguna untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas. Contohnya yaitu penanganan, perencanaan dan penjadwalan, kemampuan untuk menjawab diagnosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah. Hal-hal seperti itu telah menjadi disiplin ilmu tersendiri, yang memusatkan perhatian pada penyediaan solusi masalah kehidupan yang nyata. Sistem AI sekarang ini sering digunakan dalam bidang ekonomi, obat-obatan, teknik dan militer, seperti yang telah dibangun dalam beberapa aplikasi perangkat lunak komputer rumah dan video game.[4]

1.1.2 Sejarah Artificial Intelligence

Pada awal abad 17, René Descartes mengemukakan bahwa tubuh hewan bukanlah apa-apa melainkan hanya mesin-mesin yang rumit. Blaise Pascal menciptakan mesin penghitung digital mekanis pertama pada 1642. Pada 19, Charles Babbage dan Ada Lovelace bekerja pada mesin penghitung mekanis yang dapat diprogram.

Bertrand Russell dan Alfred North Whitehead menerbitkan Principia Mathematica, yang merombak logika formal. Warren McCulloch dan Walter Pitts menerbitkan "Kalkulus Logis Gagasan yang tetap ada dalam Aktivitas " pada 1943 yang meletakkan pondasi untuk jaringan syaraf.

Tahun 1950-an adalah periode usaha aktif dalam AI. Program AI pertama yang bekerja ditulis pada 1951 untuk menjalankan mesin Ferranti Mark I di University of Manchester (UK): sebuah program permainan naskah yang ditulis oleh Christopher Strachey dan program permainan catur yang ditulis oleh Dietrich Prinz. John McCarthy membuat istilah "kecerdasan buatan " pada konferensi pertama yang disediakan untuk pokok persoalan ini, pada 1956. Dia juga menemukan bahasa pemrograman Lisp. Alan Turing memperkenalkan "Turing test" sebagai sebuah cara untuk

mengoperasionalkan test perilaku cerdas. Joseph Weizenbaum membangun ELIZA, sebuah chatterbot yang menerapkan psikoterapi Rogerian.

Selama tahun 1960-an dan 1970-an, Joel Moses mendemonstrasikan kekuatan pertimbangan simbolis untuk mengintegrasikan masalah di dalam program Macsyma, program berbasis pengetahuan yang sukses pertama kali dalam bidang matematika. Marvin Minsky dan Seymour Papert menerbitkan Perceptrons, yang mendemonstrasikan batas jaringan syaraf sederhana dan Alain Colmerauer mengembangkan bahasa komputer Prolog. Ted Shortliffe mendemonstrasikan kekuatan 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.[5]

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 kesalahan dala, prinsip dan prakteknya,system ini dikenal dengan penalaran komputasi.

4. Acting Rationally

Yaitu *system* yang melakukan aksi dengan cara menciptakan suatu robotika cerdas yang mengantikan 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.[6]

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 industri 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 diasistensi 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 menggantikan 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 perekruit. [7]

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.[8]

1.2 Pengantar *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.[9]

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”.[10]

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 parameteranya bernilai kecil sehingga algoritma *gradient descent* berjalan lambat.[11]

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.[9]

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 perangkingan 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 penggunaanya 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.[12]

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.[13]

1.3 Pengantar *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.[14]

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.

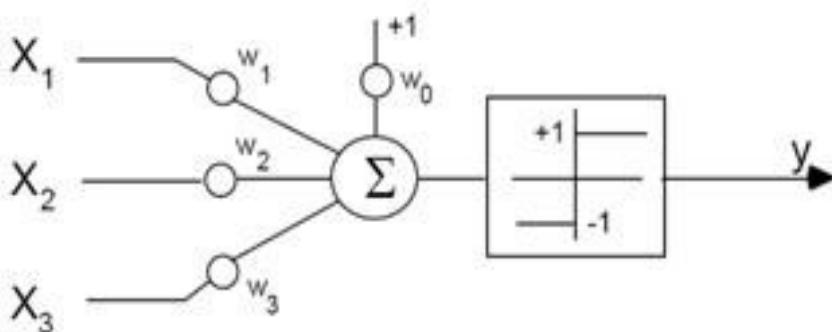
Fungsi dari *Neural Network* diantaranya adalah:

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 [15]

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

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.

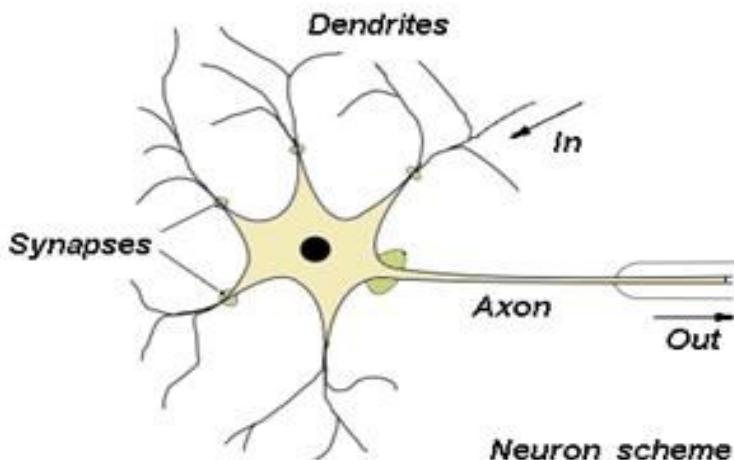
Untuk saat ini, *Neural Network* sudah dapat diterapkan pada beberapa task, diantaranya classification, recognition, approximation, prediction,

clusterization, memory simulation dan banyak task-task berbeda yang lainnya, dimana jumlahnya semakin bertambah seiring berjalannya waktu.[16]

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 dendrit. Setiap sel syaraf terhubung dengan syaraf lain, jumlahnya mencapai sekitar 104 sinapsis. 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

Dari gambar di atas, bisa dilihat ada beberapa bagian dari otak manusia, yaitu:

1. Dendrit (*Dendrites*) berfungsi untuk mengirimkan impuls yang diterima ke badan sel syaraf.

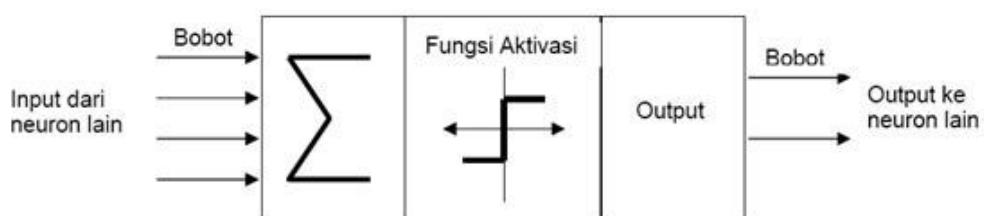
2. Akson (*Axon*) berfungsi untuk mengirimkan impuls dari badan sel ke jaringan lain
3. Sinapsis berfungsi sebagai unit fungsional di antara dua sel syaraf.

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

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 konektivitas.

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 (Σ). 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.[15]

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*.

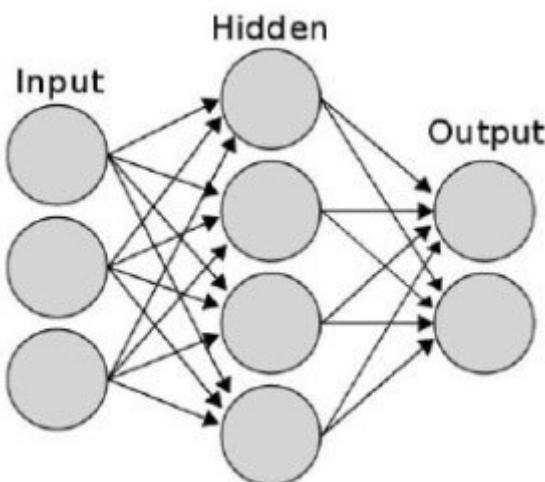
2. Algoritma *Quickprop*

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.

1.3. 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

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 koneksi pada jaringan saraf biologis. 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.[17]

1.4 Pengantar 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 algoritmanya 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. Algoritma 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. Algoritma semacam ini sangat penting dalam sebuah kecerdasan buatan karena mampu mengurangi beban pemrograman dalam memilih fitur yang eksplisit. Dan, algoritma 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. Algoritma 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.[18]

1.4.2 Sejarah Deep Learning

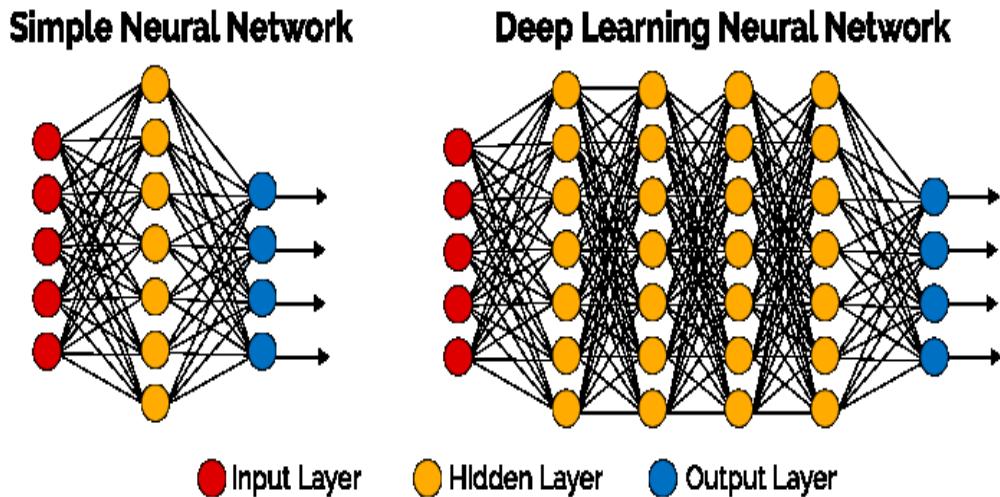
Pada tahun 2006, Geoffrey Hinton memperkenalkan salah satu varian jaringan saraf tiruan yang disebut deep belief nets, ide untuk men-train model jaringan saraf tiruan ini adalah dengan men-train dua *layer* kemudian tambahkan satu *layer* di atasnya, kemudian train hanya *layer* teratas dan begitu seterusnya. 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.**[19]

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 algoritma mana yang dapat digunakan. Dalam Jaringan Saraf Tiruan terdiri dari beberapa algoritma yaitu Model Hebb, Perceptron, Adeline, Propagation Maju, dll. Sedangkan pada algoritma 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.



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*)

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*. [20]

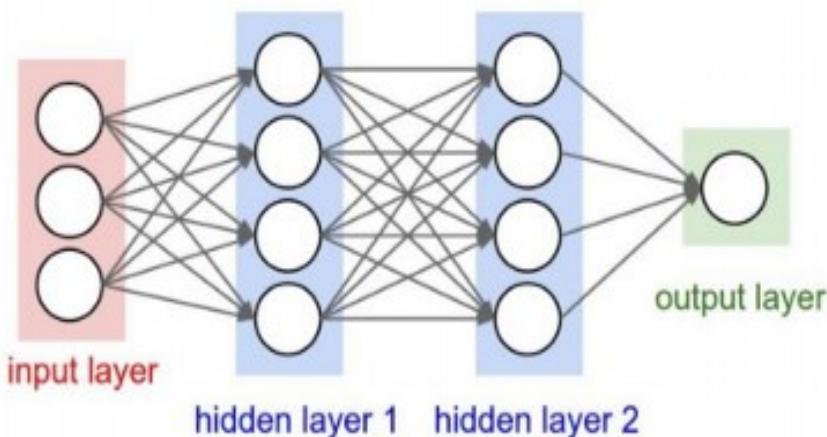
BAB II

PENGENALAN CNN, FAST R-CNN, DAN FASTER R-CNN

2.1 Pengantar Convolutional Neural Network (CNN)

2.1.1. Definisi Convolutional Neural Network

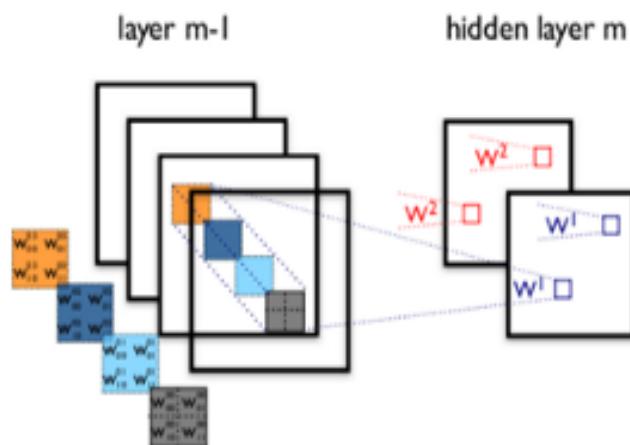
Convolutional Neural Network adalah salah satu metode *Machine Learning* dari pengembangan Multi *Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep *Neural Network* karena dalamnya tingkat jaringan dan banyak diimplementasikan dalam data citra. CNN memiliki 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

Sebuah MLP seperti pada Gambar. 6. 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 dipropagaskan 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 seperti pada Gambar.7. Dimensi bobot pada CNN adalah:

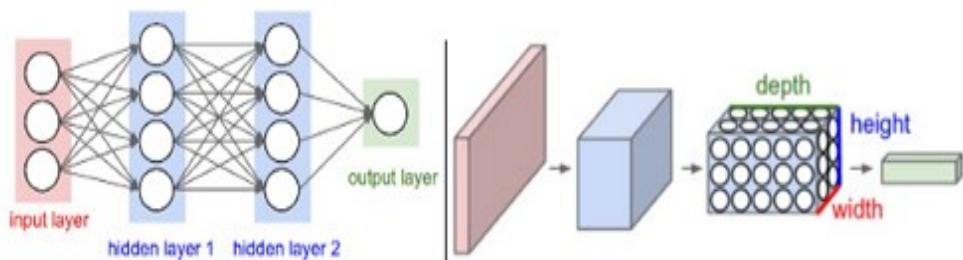


Gambar 2. 2 Proses Konvolusi pada CNN

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

CNN terdiri dari berbagai lapisan yang dimana setiap lapisan memiliki Application Program Interface (API) alias antarmuka program aplikasi sederhana. Pada Gambar 8, 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.[21]

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*.

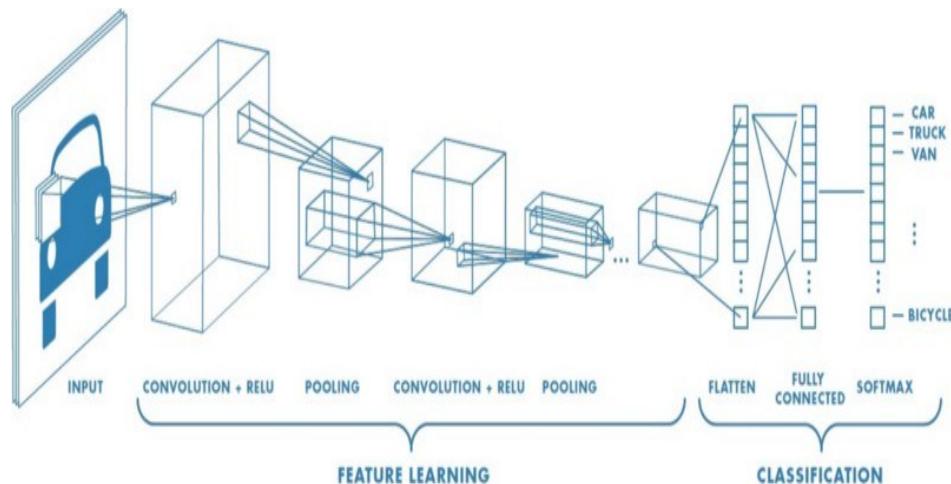
Pada tahun 1950-an Hubel dan Wisel melakukan eksperimen pada salah satu bagian otak kucing yaitu visual cortex. mereka menemukan bahwa visual cortex memiliki bagian kecil berupa sel-sel yang sensitif terhadap area tertentu pada pandangan mata. Hubel dan Wisel menemukan 2 tipe visual cortex, yaitu simple 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 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

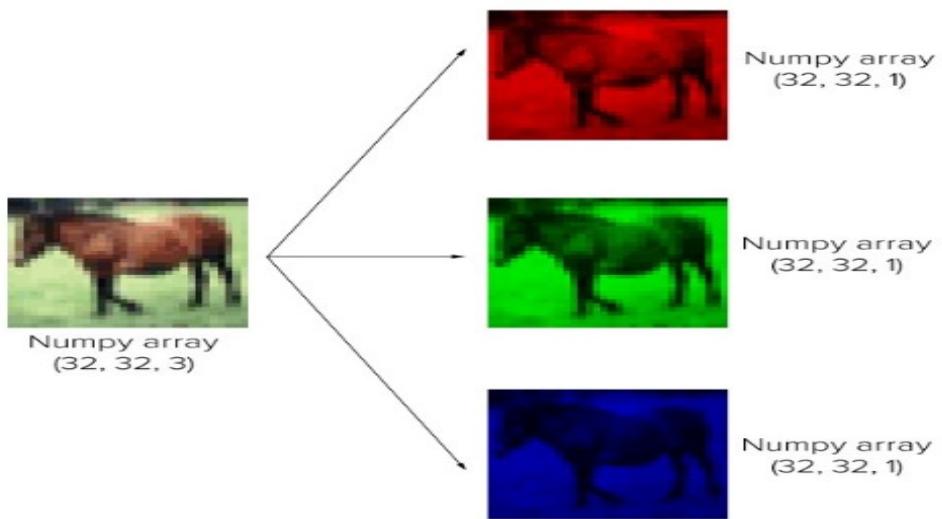
Input Layer menampung nilai piksel dari citra yang menjadi masukan. Untuk citra dengan ukuran 64×64 dengan 3 channel warna, RGB(Red, Green, Blue) maka yang menjadi masukan akan adalah piksel array yang berukuran $64 \times 64 \times 3$.



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 *inputan*. Filter pada *layer* ini yaitu array 2 dimensi dapat berukuran 5×5 , 3×3 atau 1×1 . Proses convolution dengan memakai filter pada *layer* ini akan menghasilkan feature map yang akan digunakan pada activation *layer*. Gambar 2.6 dibawah ini menunjukkan alur pada Convolution *Layer*



Gambar 2. 6 Image RGB Convolutional Layer

Gambar 2.6 diatas merupakan *RGB (Red, Green, Blue)* dari sebuah gambar yang memiliki ukuran 32×32 *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	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																																																					

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

Untuk menghitung dimensi dari Feature Map kita bisa gunakan rumus seperti dibawah ini:

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

Gambar 2. 5 Rumus Menghitung Dimensi Feature Map

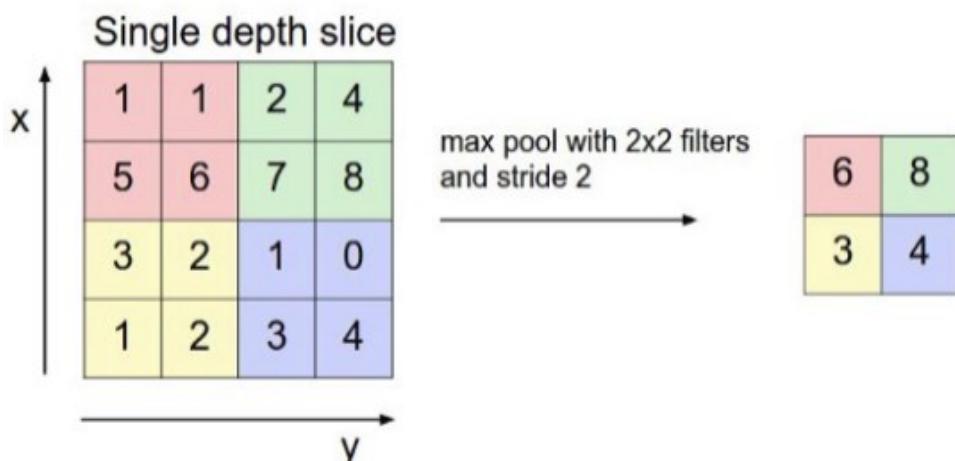
- (a) W = Panjang/Tinggi Input
- (b) N = Panjang/Tinggi Filter
- (c) P = Zero Padding
- (d) S = Stride
- c. *Dropout*

Dropout merupakan proses mencegah terjadinya overfitting dan juga mempercepat proses learning. *Dropout* mengacu kepada menghilangkan

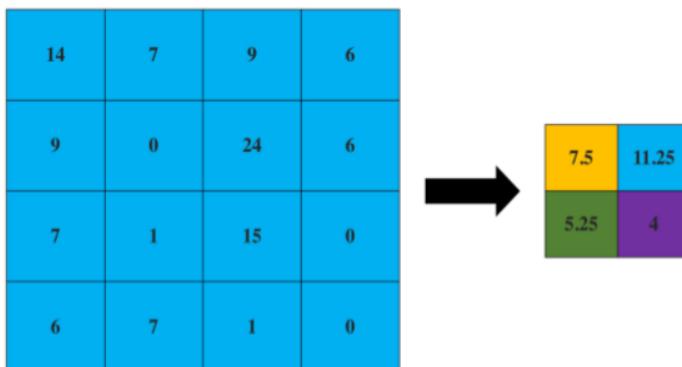
neuron yang berupa hidden maupun *layer* yang visible di dalam jaringan. Dengan menghilangkan suatu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap *neuron* akan diberikan probabilitas yang bernilai antara 0 dan 1.

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*.



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* 0 jika *input* kurang dari 0, jika *input* lebih besar dari 0 maka *output* sama dengan *input*.



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

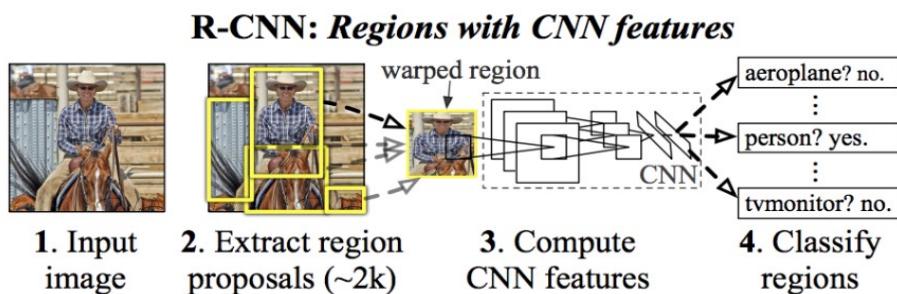
Ada berbagai algoritma proposal wilayah yang dapat kita pilih. Ini adalah algoritma "normal" yang bekerja di luar kotak. Kami tidak harus melatih mereka atau apa pun. Dalam kasus makalah ini, mereka menggunakan metode pencarian selektif untuk menghasilkan proposal wilayah. Saya menemukan penjelasan yang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

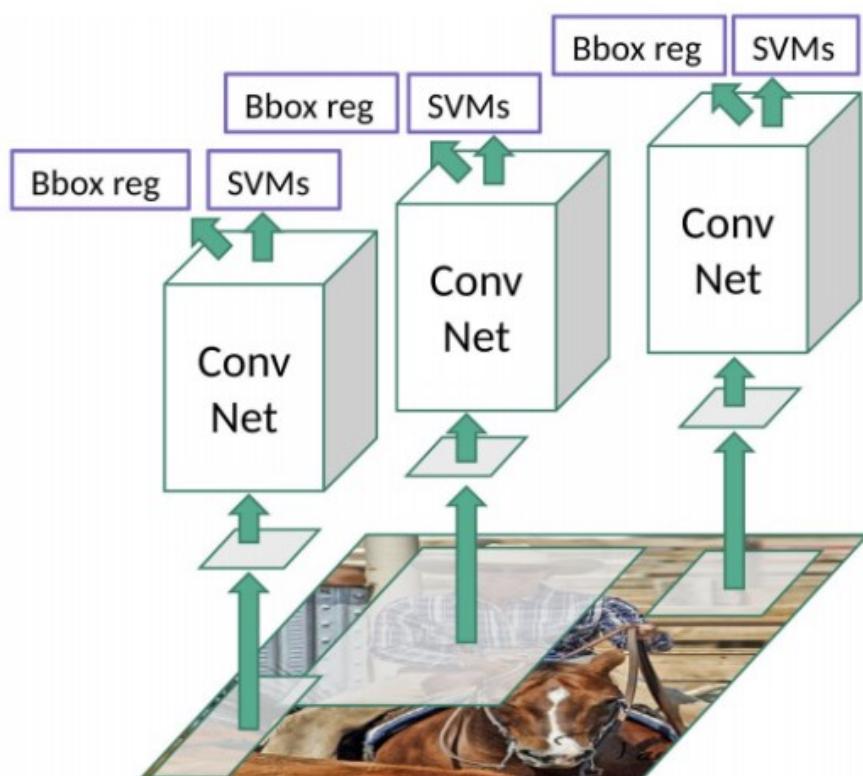
- Menghasilkan sub-segmentasi awal, kami menghasilkan banyak kandidat daerah
- item Menggunakan algoritma serakah untuk secara rekursi menggabungkan daerah yang sama menjadi yang lebih besar
- Menggunakan daerah yang dihasilkan untuk menghasilkan proposal daerah kandidat akhir



Gambar 2. 12 Arsitektur 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.



Gambar 2. 13 Proses R-CNN

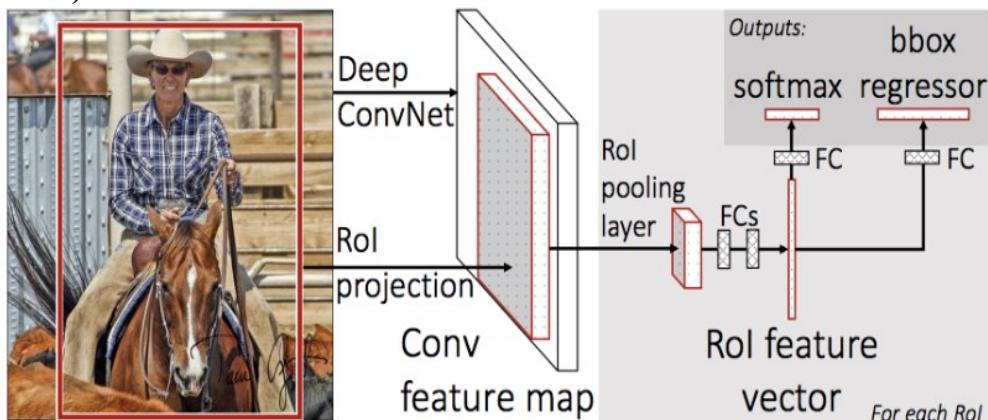
2.2.1 Masalah Dengan R-CNN

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)

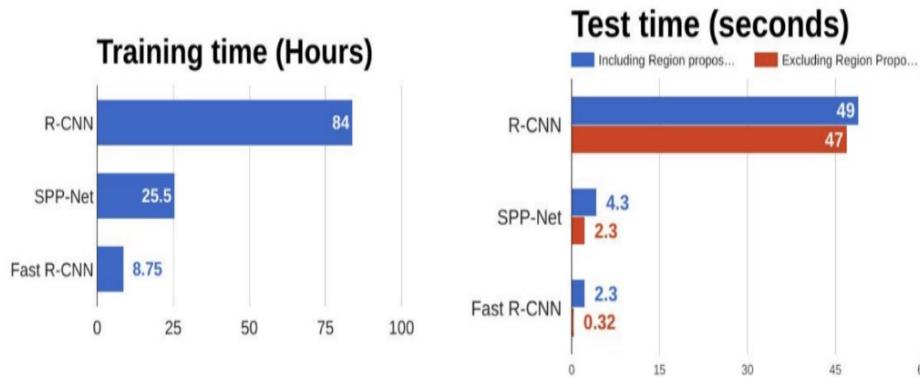


Gambar 2. 14 Fast R-NN

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 membengkokkannya 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

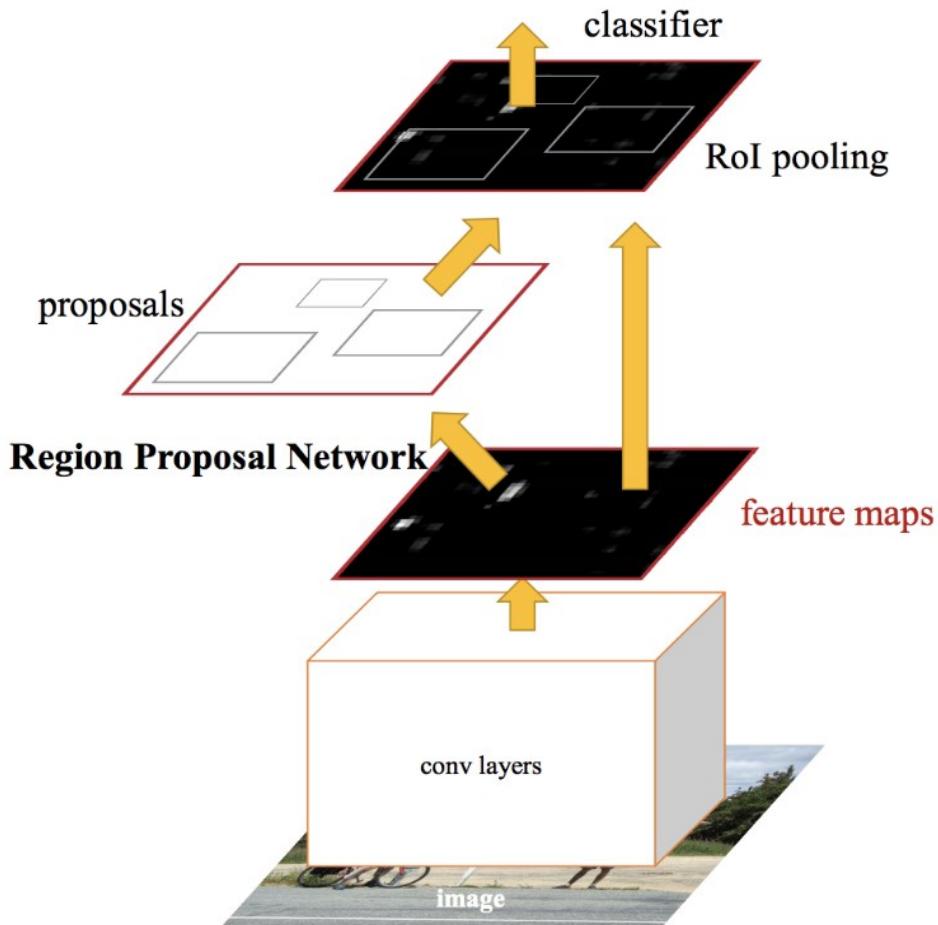
Convolutional setiap waktu. Sebaliknya, operasi konvolusi dilakukan hanya sekali per gambar dan peta fitur dihasilkan darinya.



Gambar 2. 15 Perbandingan Algoritma Objek Deteksi

Dari gambar grafik 2.20 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)

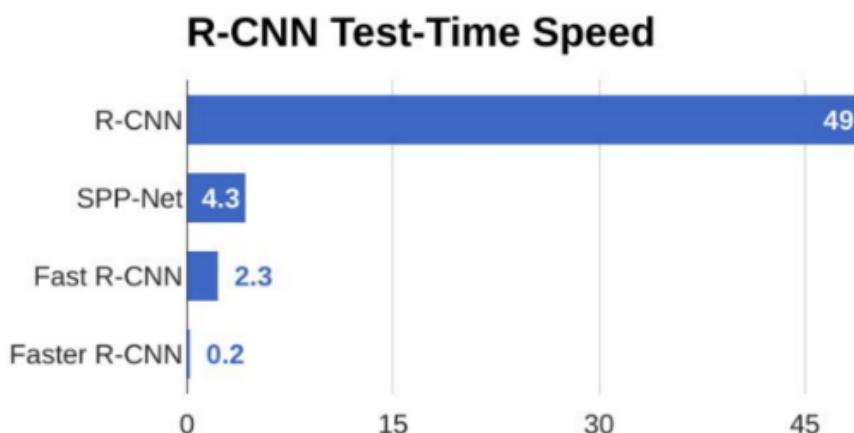


Gambar 2. 16 Arsitektur Faster R-CNN

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 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 mendekripsi 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

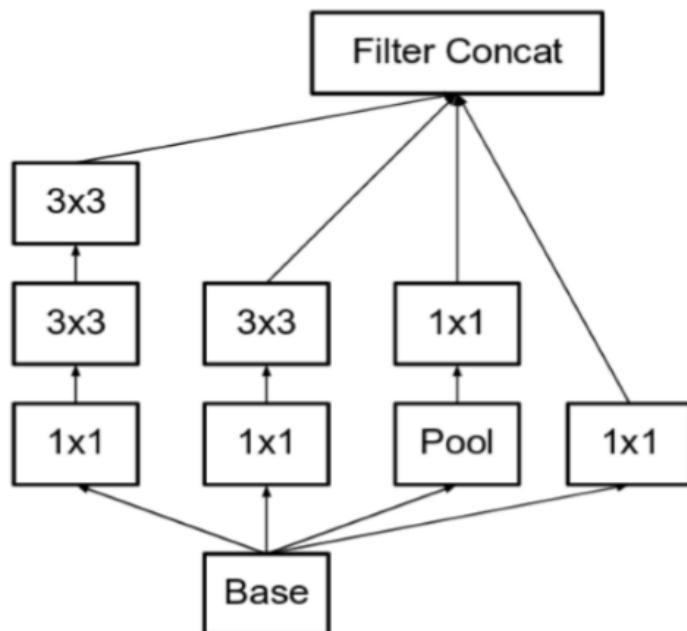
2. Jalankan CNN untuk setiap bounding box yang dihasilkan dari RPN
3. Setiap output yang dihasilkan CNN menjadi input SVM (*Support Vector Machine*) untuk klasifikasi kelasnya dan juga sebagai input untuk linear regression untuk memperbaiki bounding box agar lebih presisi

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).

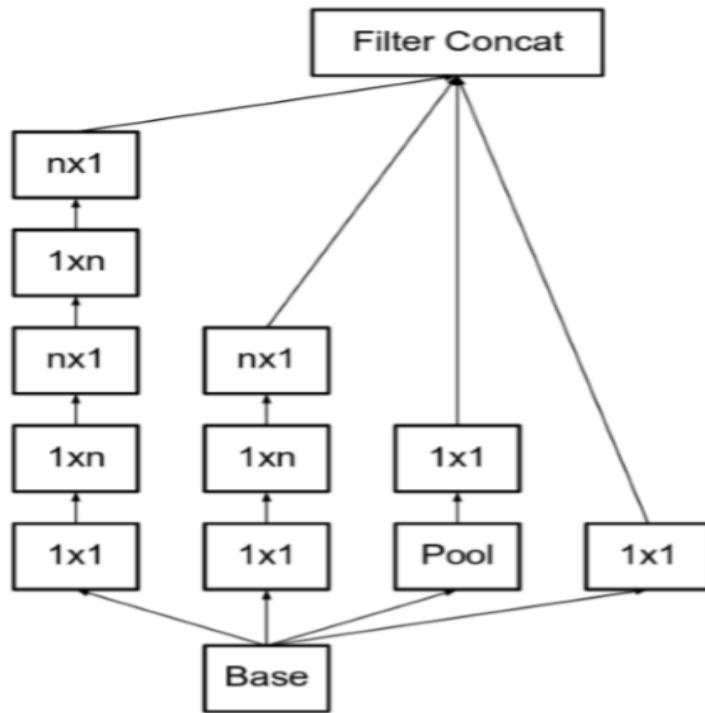
CNN memiliki berbagai arsitektur, salah satunya dalam penelitian ini adalah *Inception* V2. Arsitektur dari *Inception* V2 dirancang untuk mengurangi kompleksitas CNN, yang dilakukan dengan cara menyusun arsitektur yang lebih melebar dari pada mendalam. *Inception* V2 memiliki 3 modul yang ditunjukkan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

a. Modul Pertama *Inception* V2



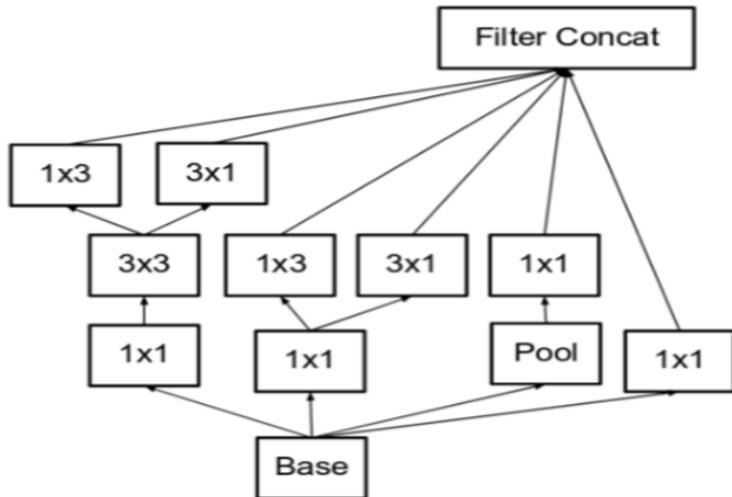
Gambar 2. 19 Modul Pertama *Inception* V2

b. Modul kedua *Inception V2*



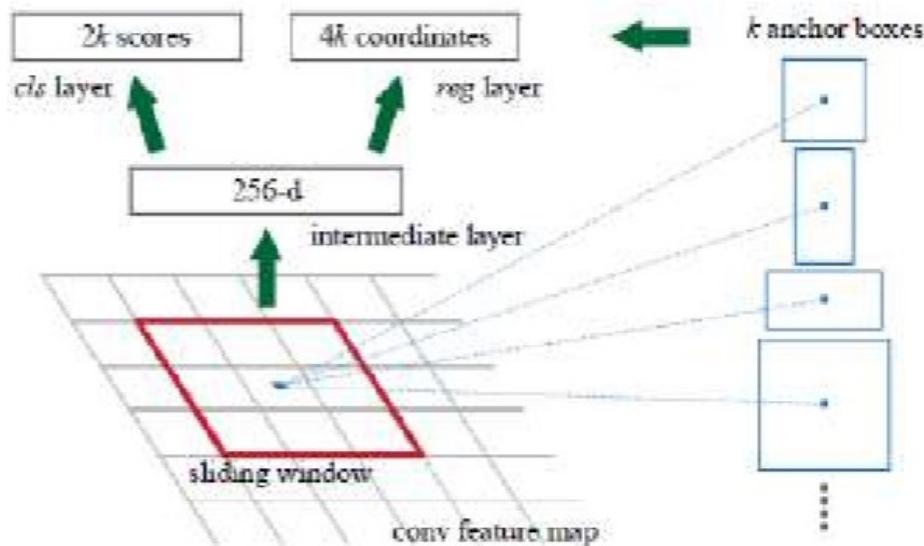
Gambar 2. 20 Modul kedua Inception V2

c. Modul Ketiga 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 berikut:



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

BAB III

PENJELASAN TOOLS DAN BAHASA PEMOGRAMAN YANG DIGUNAKAN

3.1 Tools Yang Digunakan

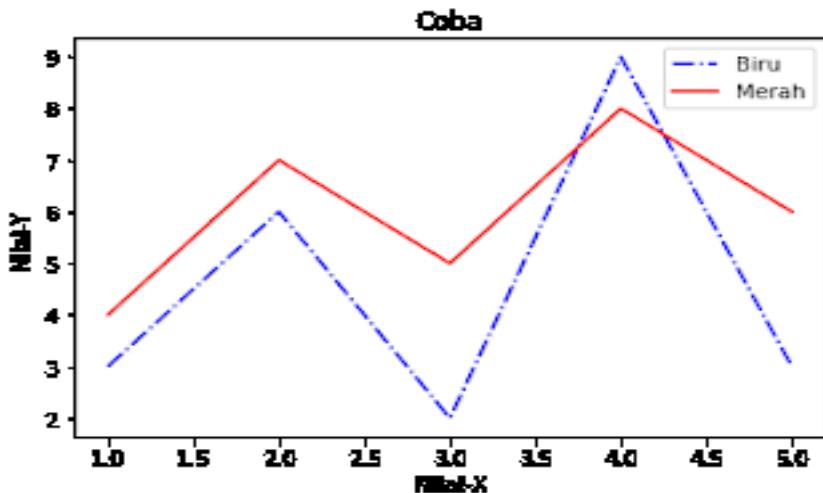
3.1.1 Anaconda



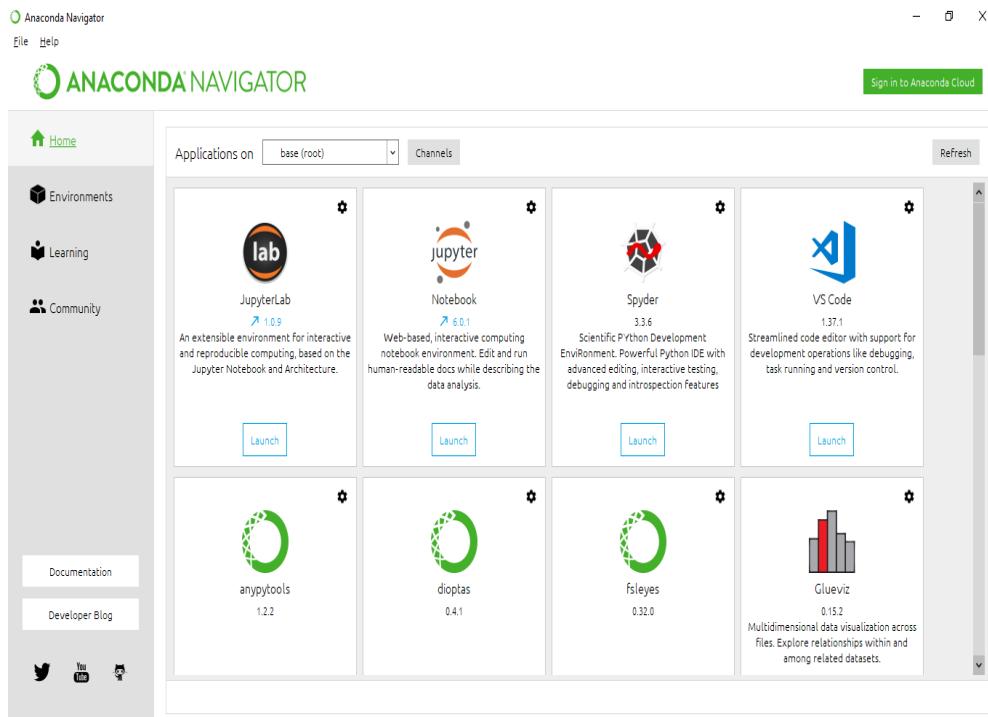
Gambar 3. 1 Logo Anaconda

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.



Gambar 3. 2 grafik implementasi matplotlib pada Python

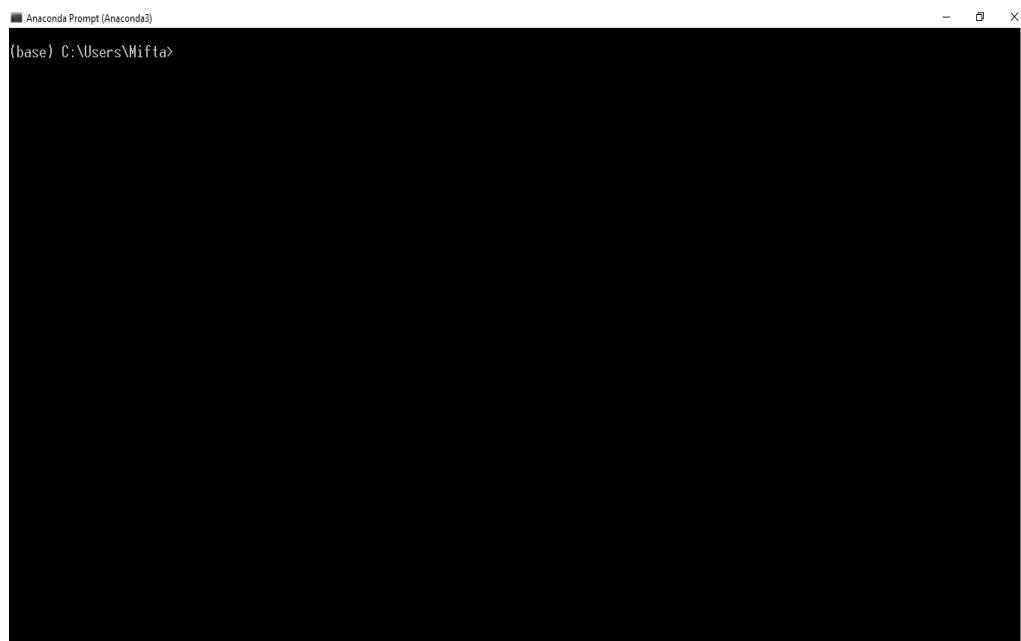


Gambar 3. 3 Tampilan Anaconda Navigator

Gambar 3.2 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-perbari. 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 3.3. 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 :

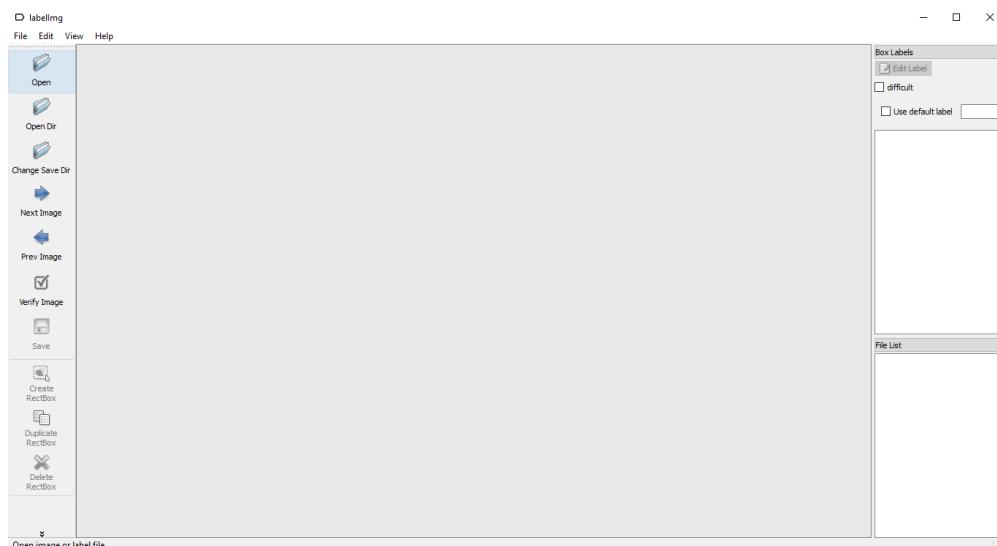
- *Anaconda Navigator*
- Pengkompilasian dengan rilisan Python terbaru
- Keandalan yang lebih baik
- Peningkatan kinerja CPU
- Lebih dari 230 paket yang diperbarui dan ditambahkan

3.1.2 LabelImg



Gambar 3. 5 Logo LabelImg

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

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

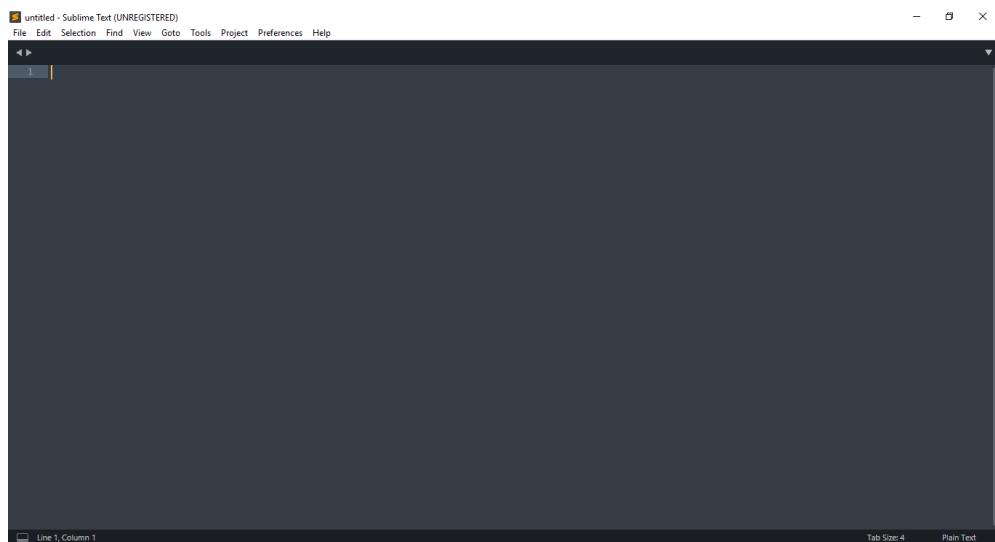


Gambar 3. 7 Logo Sublime Text

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 lisensi 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.



Gambar 3. 8 Tampilan Sublime Text

Berikut beberapa fitur yang diunggulkan dari aplikasi Sublime Text:

- **Goto 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 Goto 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.

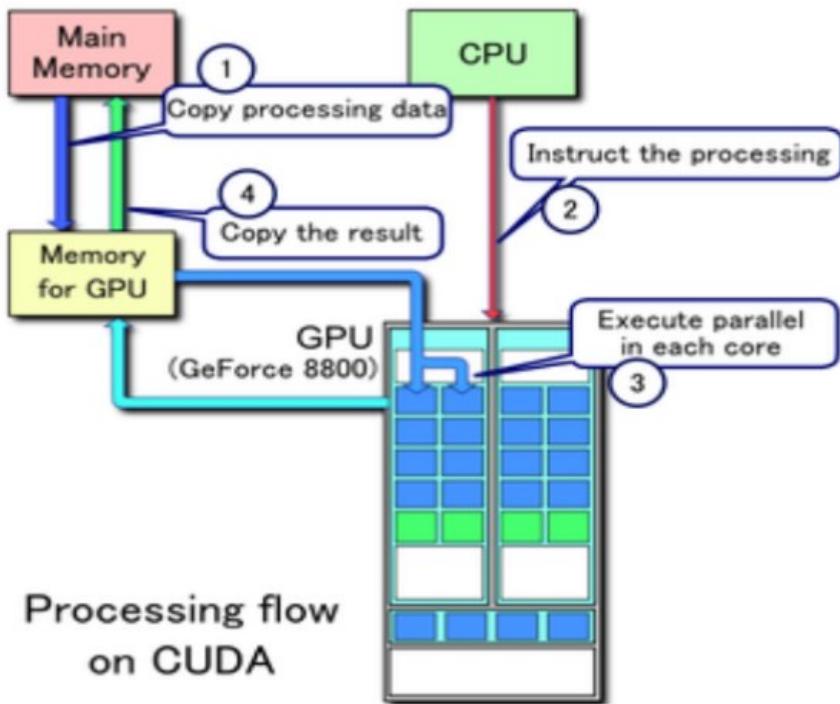
Selain perpustakaan, arahan kompiler, CUDA C / C ++ dan CUDA Fortran, platform CUDA mendukung antarmuka komputasi lainnya, termasuk OpenCL Khronos Group, Microsoft DirectCompute, OpenGL Compute Shaders, dan C ++ AMP. Pembungkus pihak ketiga juga tersedia untuk

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

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 pemrograman 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 pemrograman 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 IV

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/>

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	e9845ccf6754d2523c5be09552311666e
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

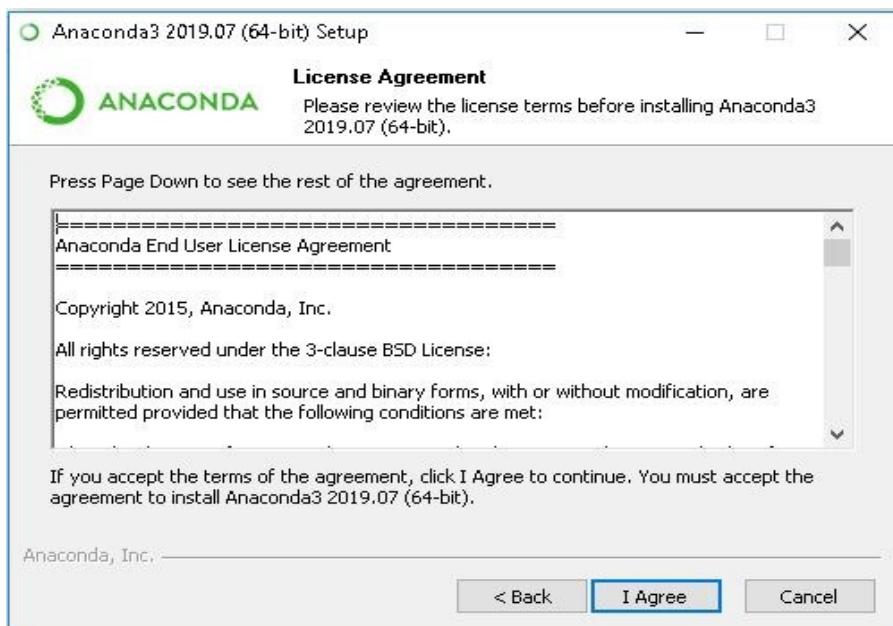
Gambar 4. 1 Download File Anaconda

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



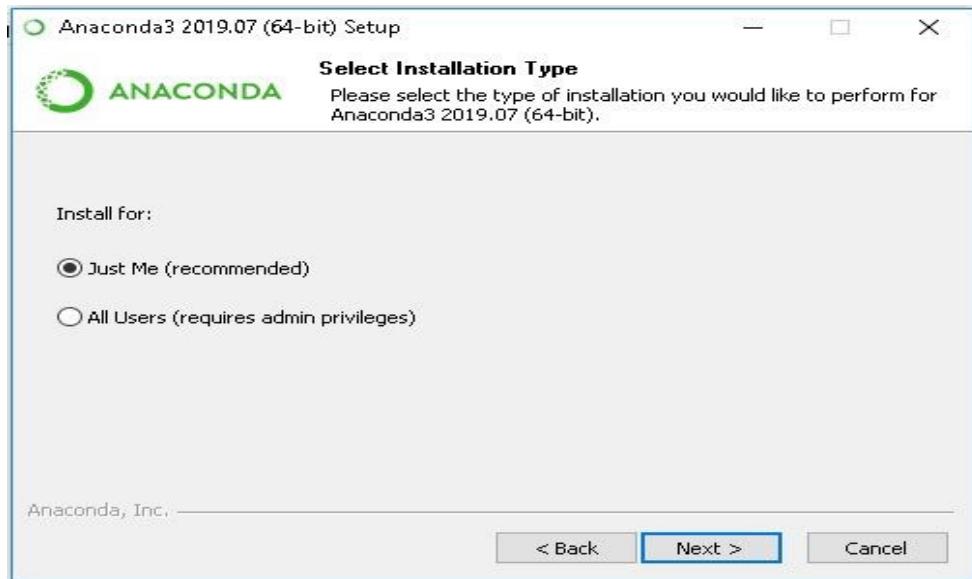
Gambar 4. 2 Tahap Awal Install Anaconda

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



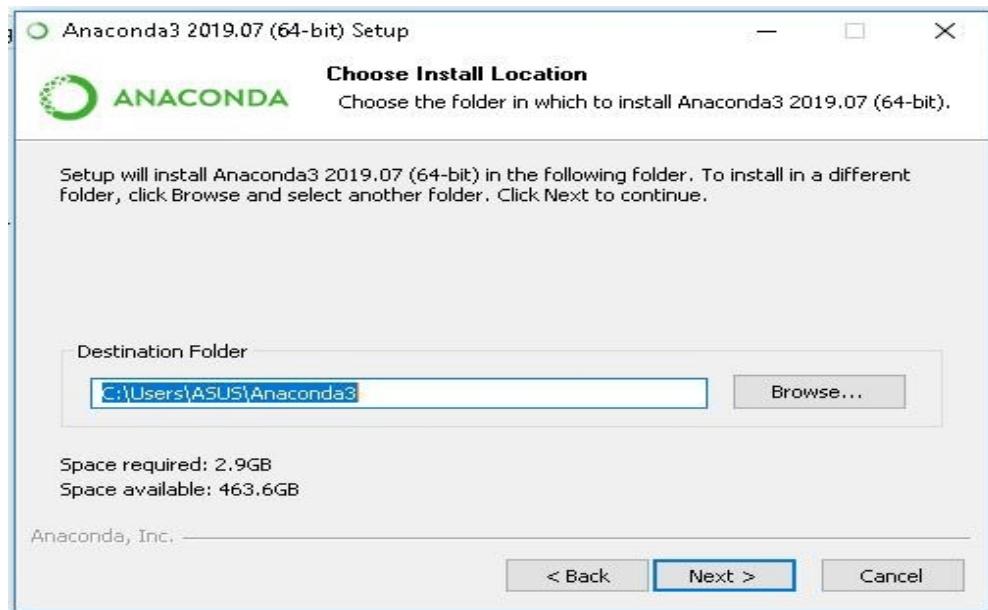
Gambar 4. 3 License Agreement

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



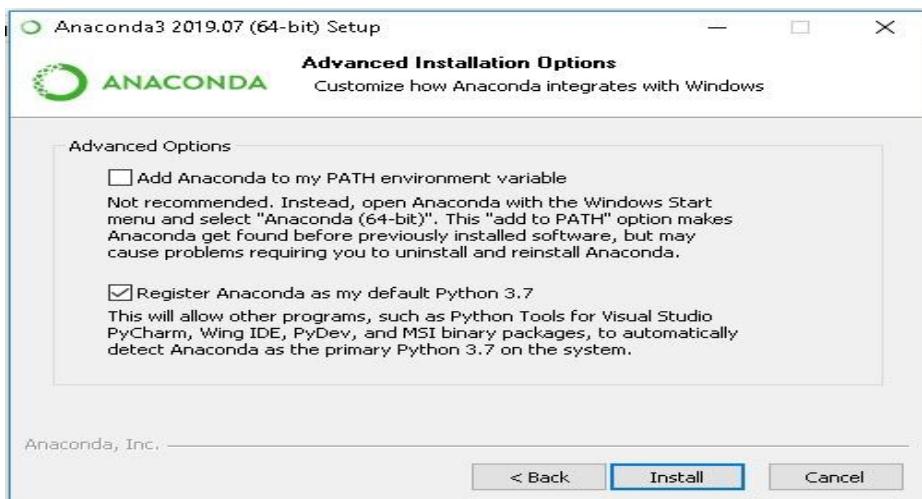
Gambar 4. 4 Select Installation Type

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



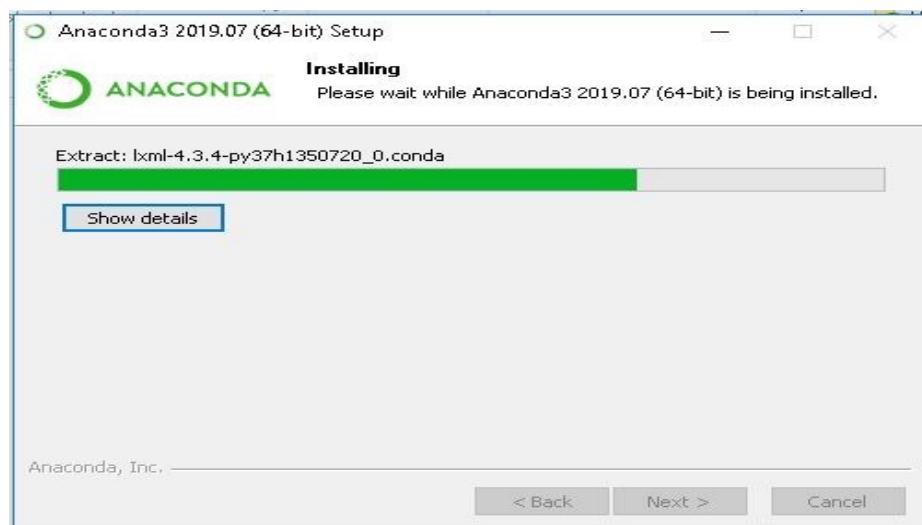
Gambar 4. 5 Choose Install Location

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.



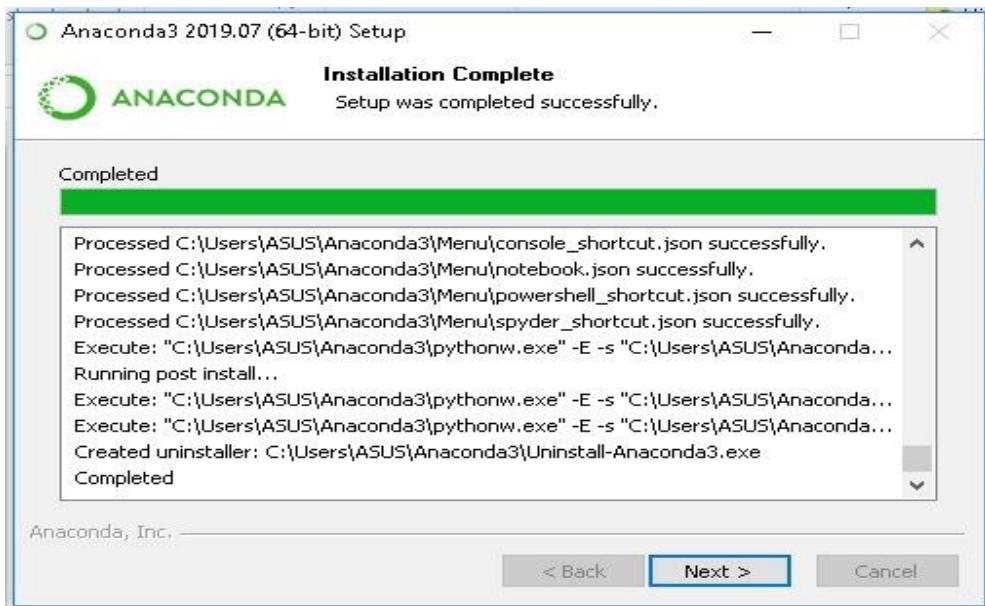
Gambar 4. 6 Advanced Installation Options

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



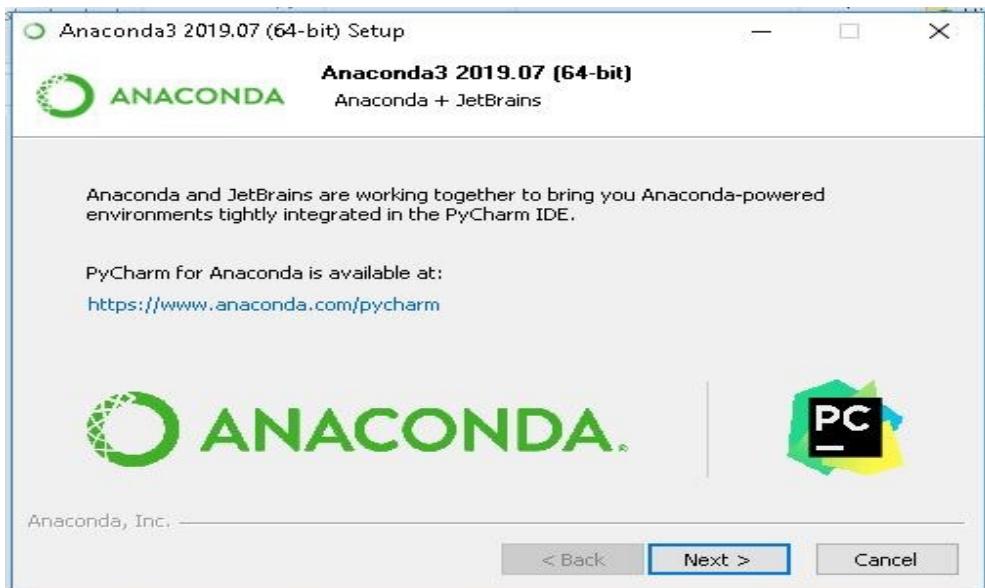
Gambar 4. 7 Proses Installing

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



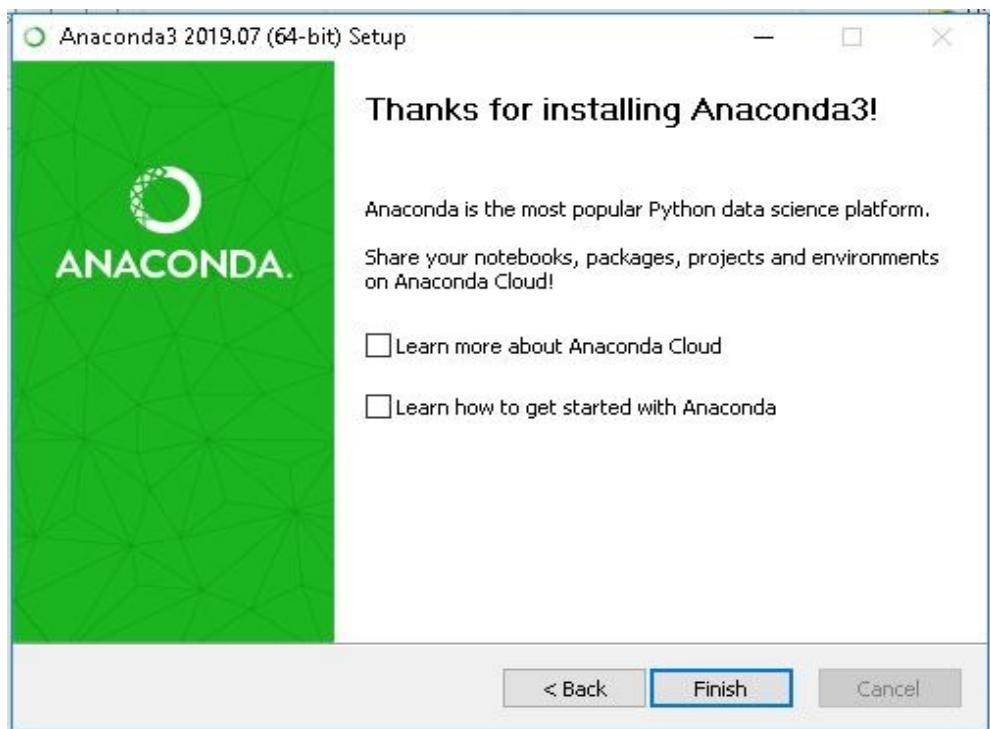
Gambar 4. 8 Installation Complete

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



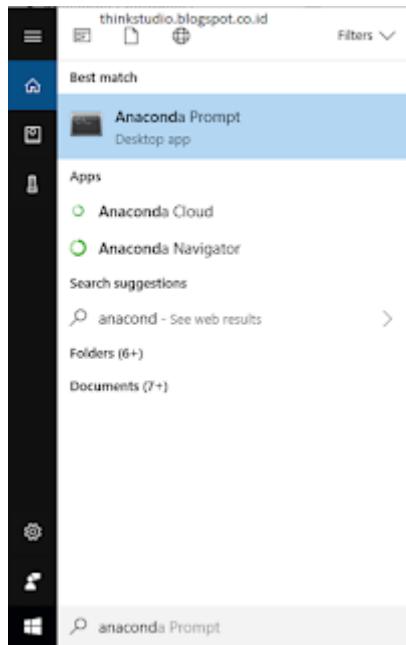
Gambar 4. 9 Anaconda3 2019.07(64-bit)

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>
```

The image shows a terminal window titled "Anaconda Prompt (Anaconda3)". It displays two command-line sessions. The first session runs "conda -V" and outputs "conda 4.7.10". The second session runs "python -V" and outputs "Python 3.7.3". Both commands are preceded by a "(base)" prompt, indicating the current environment. The terminal has a dark background with white text.

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:

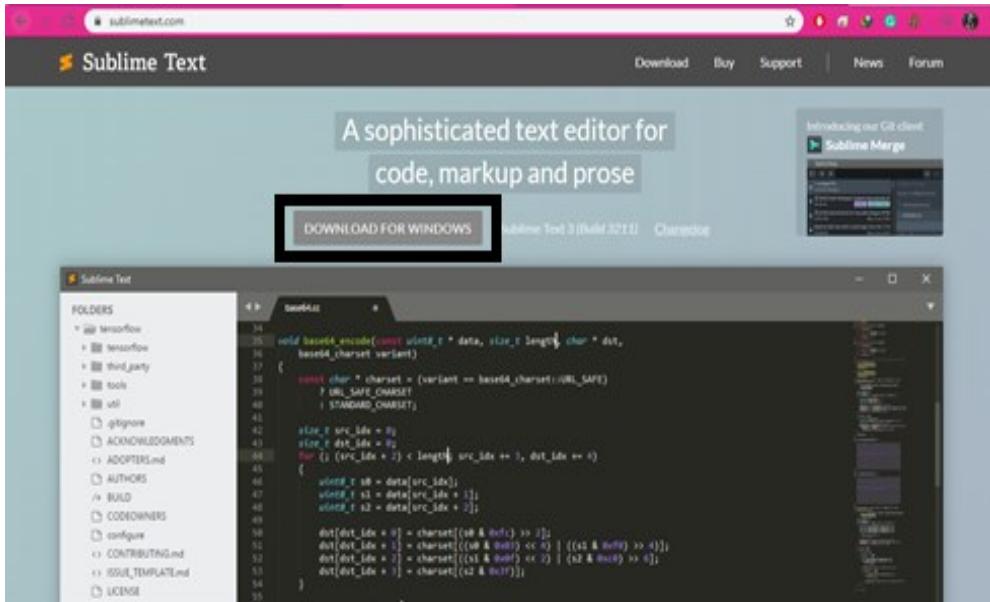
- Daftar label Anda tidak akan berubah di tengah pemrosesan daftar gambar. Saat Anda menyimpan gambar, classes.txt juga akan diperbarui, sedangkan anotasi sebelumnya tidak akan diperbarui.
- Anda seharusnya tidak menggunakan fungsi "kelas default" saat menyimpan ke format YOLO, itu tidak akan dirujuk.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

4.1.3 Instalasi Sublime

Berikut adalah langkah-langkah untuk melakukan instalasi Sublime :

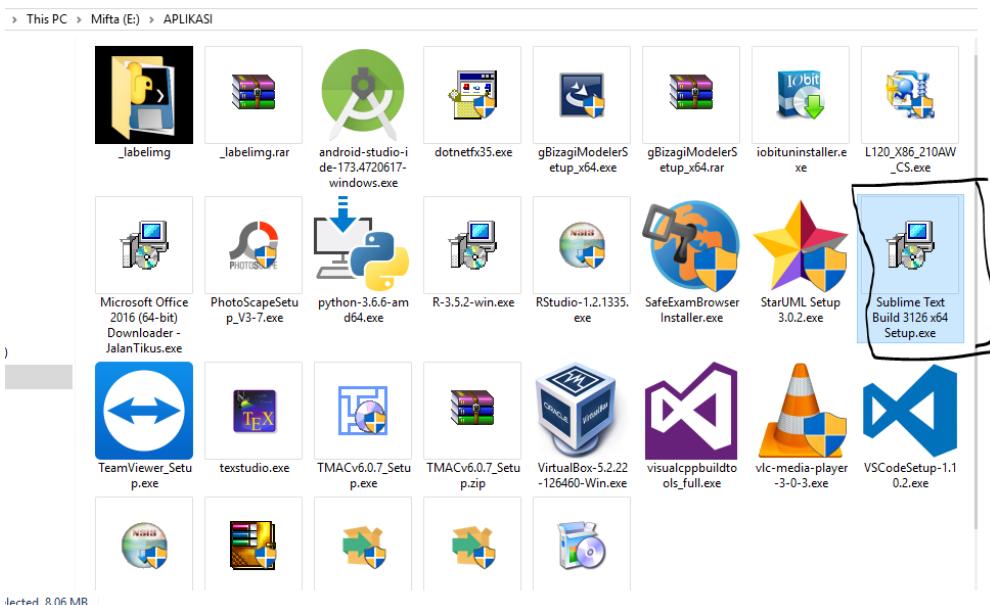
1. Tahapan yang pertama untuk melakukan instalasi maka download dahulu file .exe pada link berikut :

<https://www.sublimetext.com/>.



Gambar 4. 14 Download file .exe

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



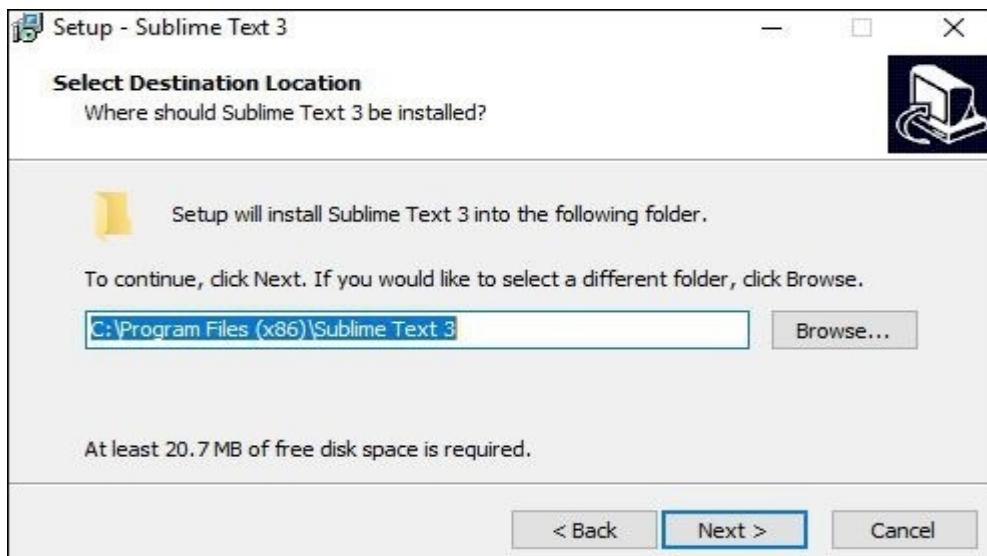
Gambar 4. 15 Double Klik File .exe

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



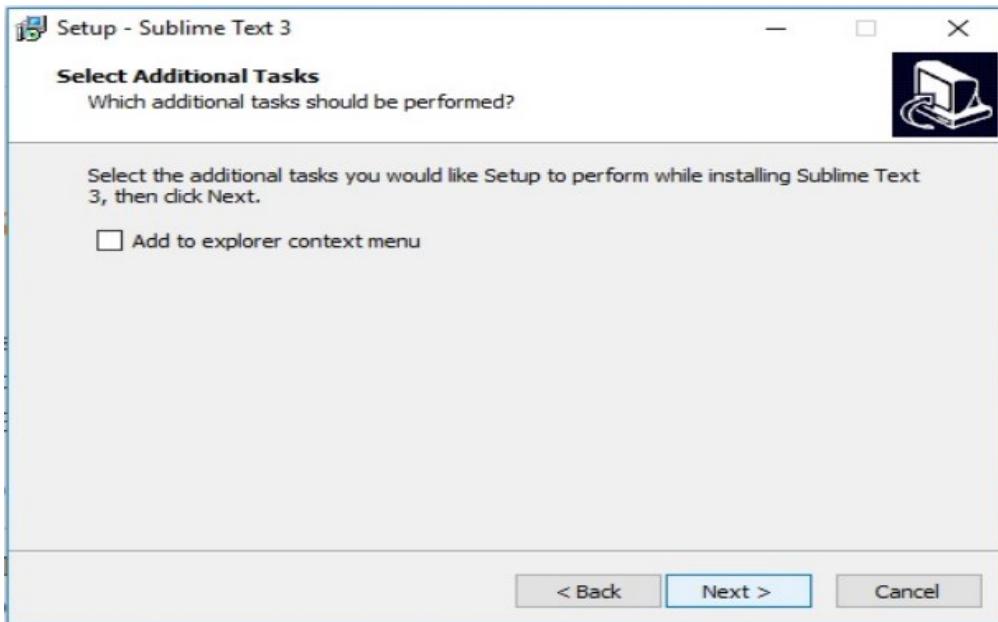
Gambar 4. 16 Proses Instalasi Sublime

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



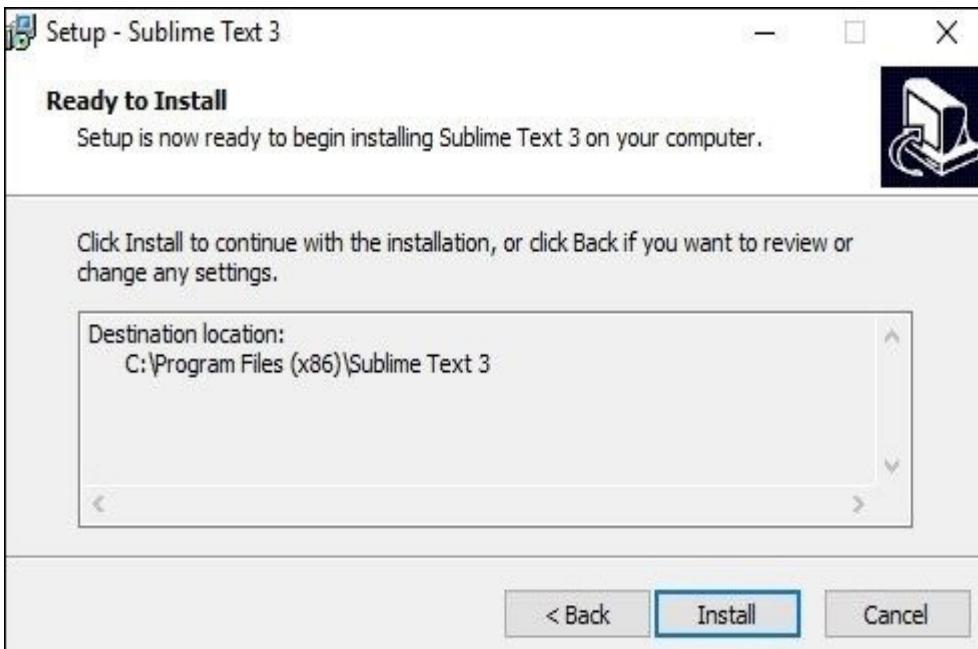
Gambar 4. 17 Select Destination Location

5. Selanjutnya klik next.



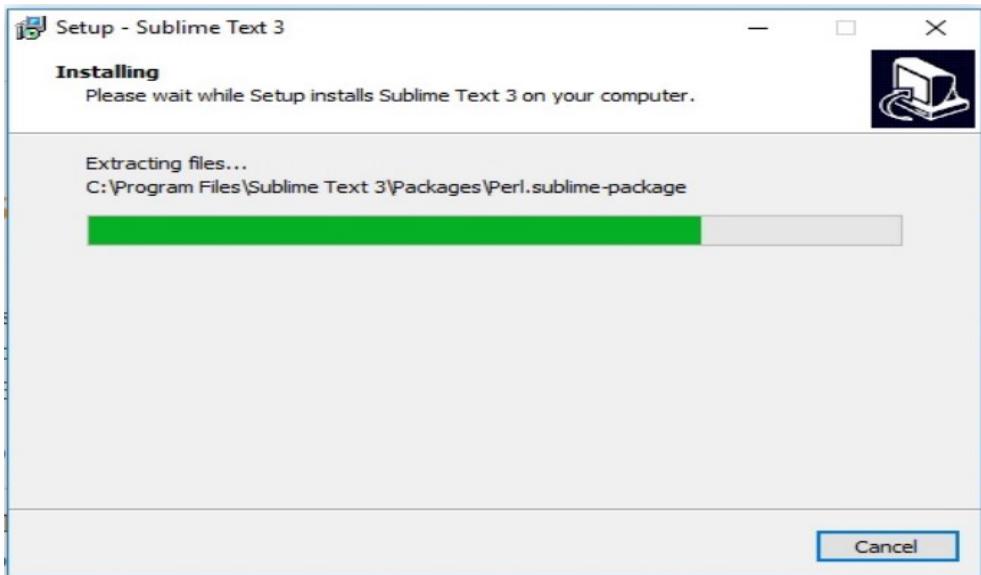
Gambar 4. 18 Select Additional Tasks

6. Verifikasi folder tujuan dan klik Instal .



Gambar 4. 19 Proses Instalasi Sublime

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



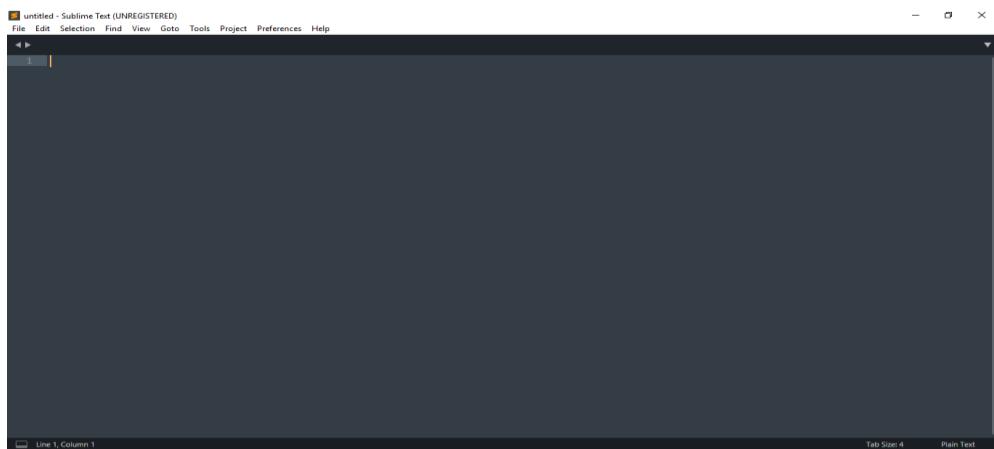
Gambar 4. 20 Proses Instalasi Sublime

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



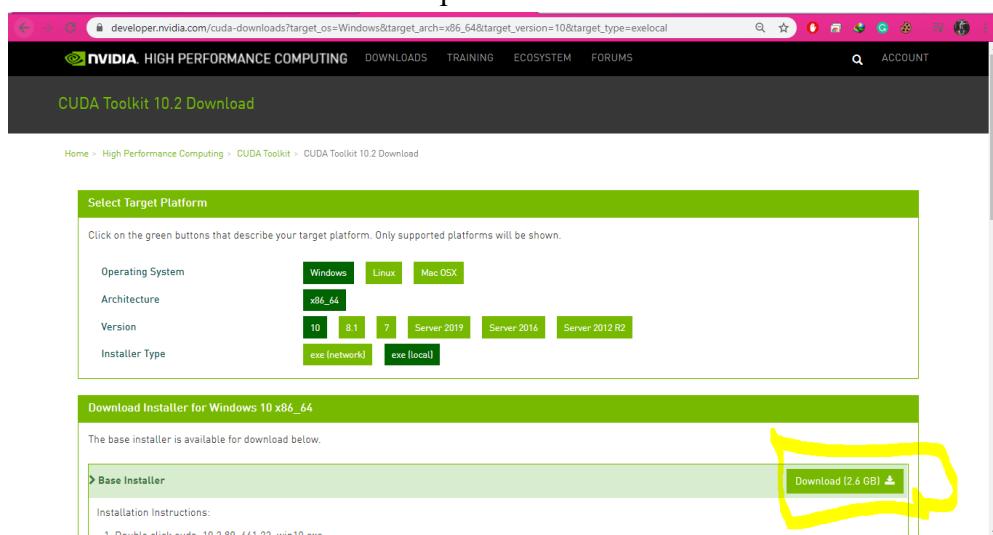
Gambar 4. 22 Tampilan Sublime

4.1.4 Instalasi CUDA

Berikut adalah langkah-langkah untuk melakukan instalasi CUDA :

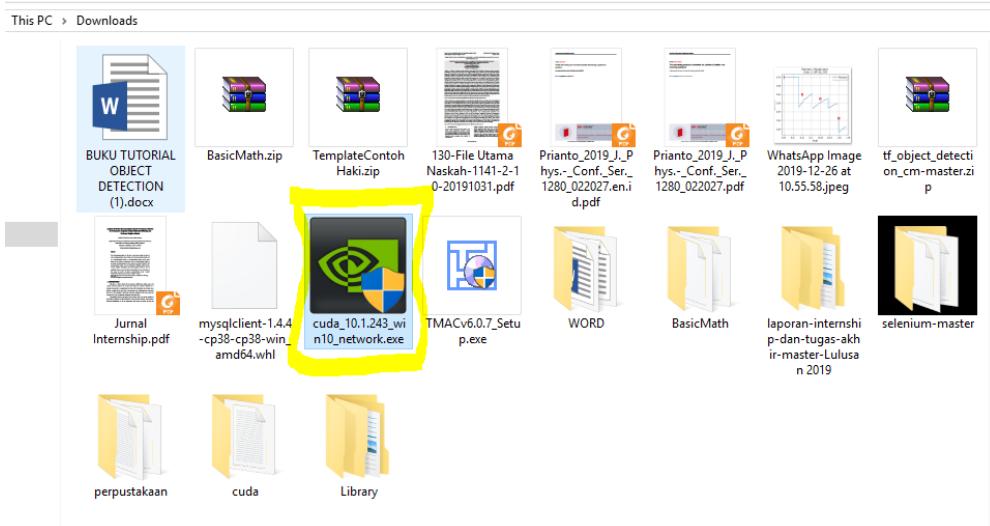
1. Langkah pertama yaitu download terlebih dahulu file .exe pada link berikut : <https://developer.nvidia.com/cuda-downloads>

Nanti kita akan diminta memilih versi yang akan kita download, pilih saja sesuai OS dan arsitektur komputer.



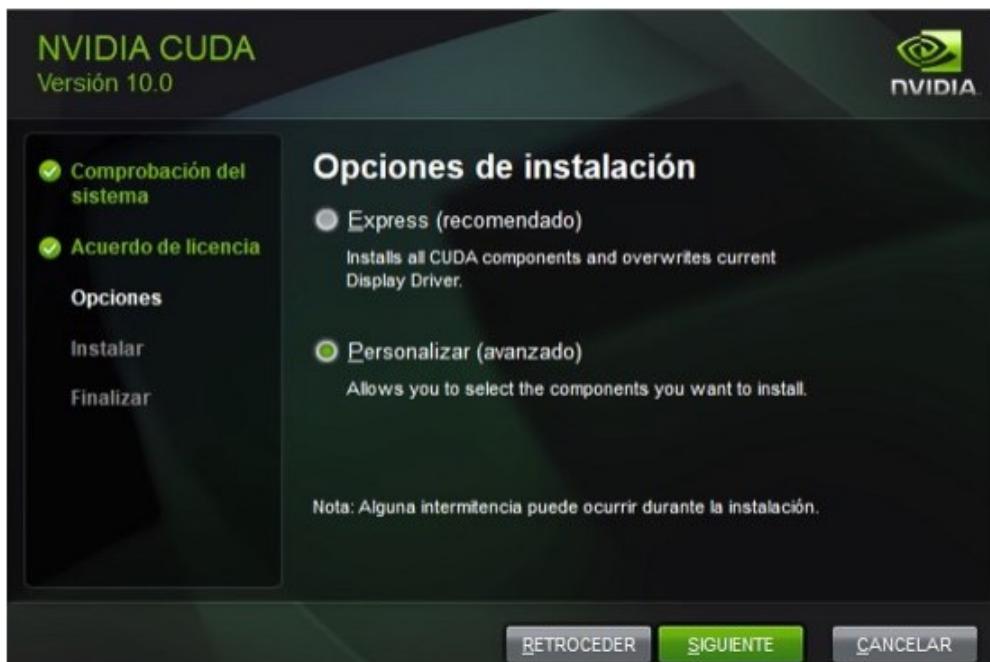
Gambar 4. 23 Download File .exe CUDA

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

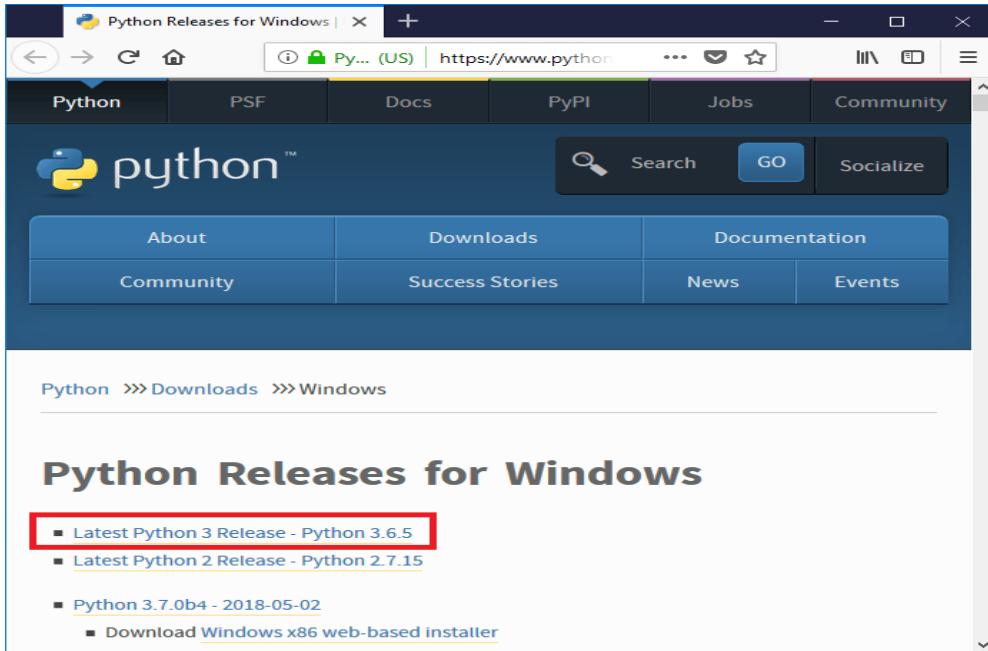
Gambar 4. 26 Menambahkan CUDA pada Environment Variable

4.2 Bahasa Pemrograman Yang Digunakan

4.2.1 Instalasi Python

Berikut adalah langkah-langkah untuk melakukan instalasi Bahasa pemrograman 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

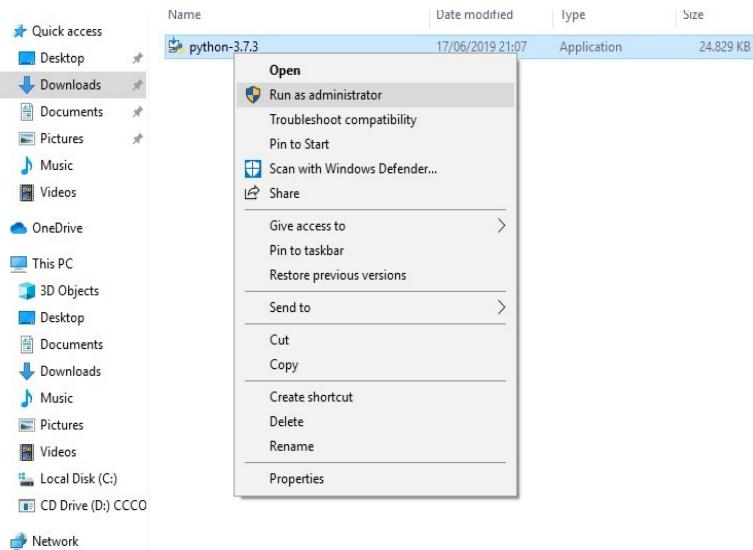
2. Kemudian sesuaikan versi arsitektur sistem operasi Windows yang digunakan. Pilih x86-64 untuk arsitektur sistem operasi 64bit atau pilih x86 untuk arsitektur sistem operasi 32bit.

The screenshot shows a table of Python release options for Windows. The columns are: Name, Platform, Description, SHA256 Hash, File Size, and Status. Several rows are highlighted with red boxes:

Name	Platform	Description	SHA256 Hash	File Size	Status
Gzipped source tarball	Source release		ab25d24b1f8cc4990ade979f6dc37883	22994617	SIG
XZ compressed source tarball	Source release		9f49654a4d6f73ff3284ab9d227e9fd	17049912	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	bf319337bc68b52fc7d227dca5b6f2f6	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		2bb6ad2eccaa6088171ef923bca483f02	30735232	SIG
Windows x86 web-based installer	Windows		596667cb91a9fb20e6f4f153f3a213a5	1294096	SIG

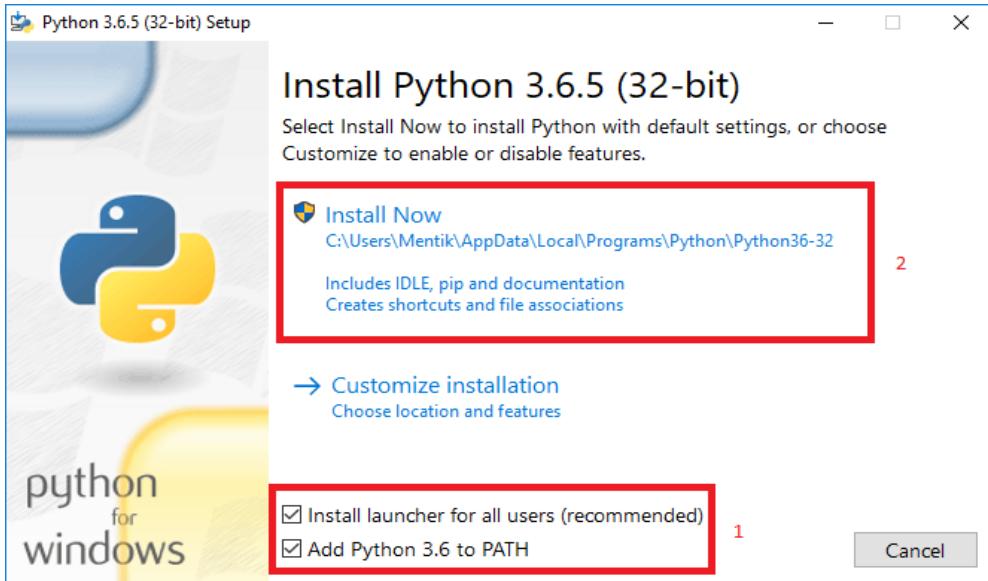
Gambar 4. 28 Sesuaikan Versi Arsitektur Sistem Operasi Windows Yang Digunakan

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 mengeksekusi nya atau bisa juga dengan pilih run as administrator.



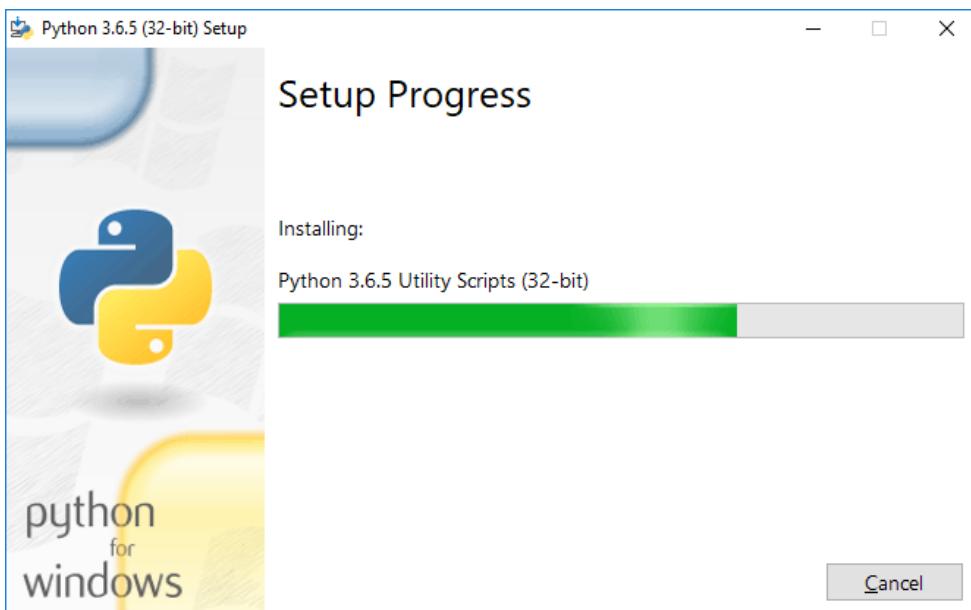
Gambar 4. 29 Instalasi Python Pada Windows

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

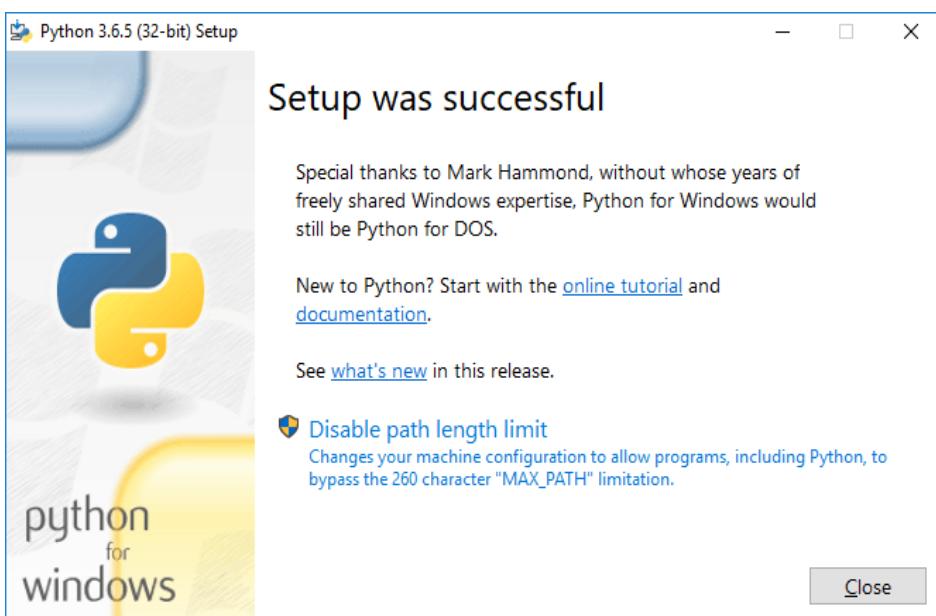
5. Centang pilihan “**Add Python 3.7 to PATH**” di bagian bawah jendela (1). Ini berguna agar python interpreter bisa diakses dari lokasi mana saja. Setelah itu klik tombol “**Install Now**” (2). Proses instalasi akan berlangsung beberapa saat.



Gambar 4. 31 Proses instalasi Python sedang berlangsung

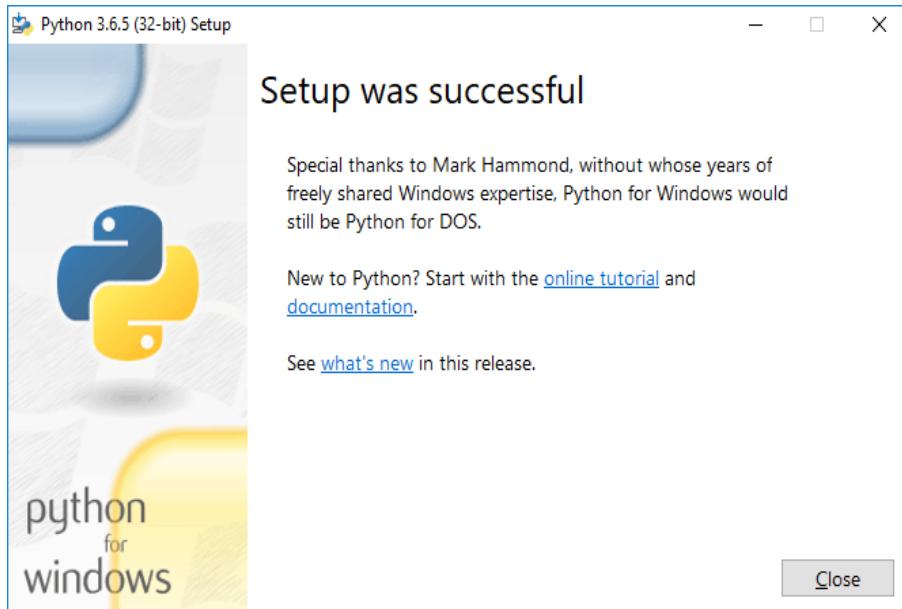
6. Opsi “Disable path length limit”

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 mendapatkan 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

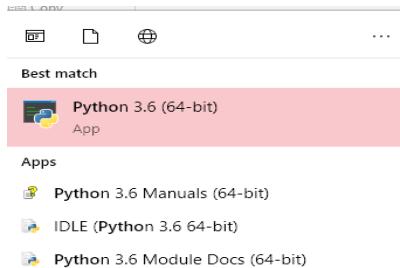
Python 3.6 untuk membuka python melalui command prompt.

- 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.

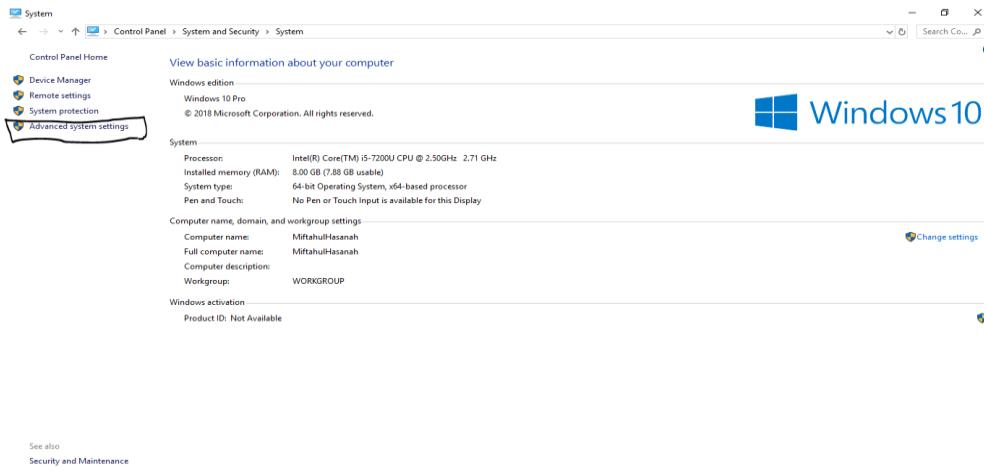
C:\Users\Mifta>python --version
Python 3.6.6

C:\Users\Mifta>
```

Gambar 4. 35 Versi Python Yang Digunakan

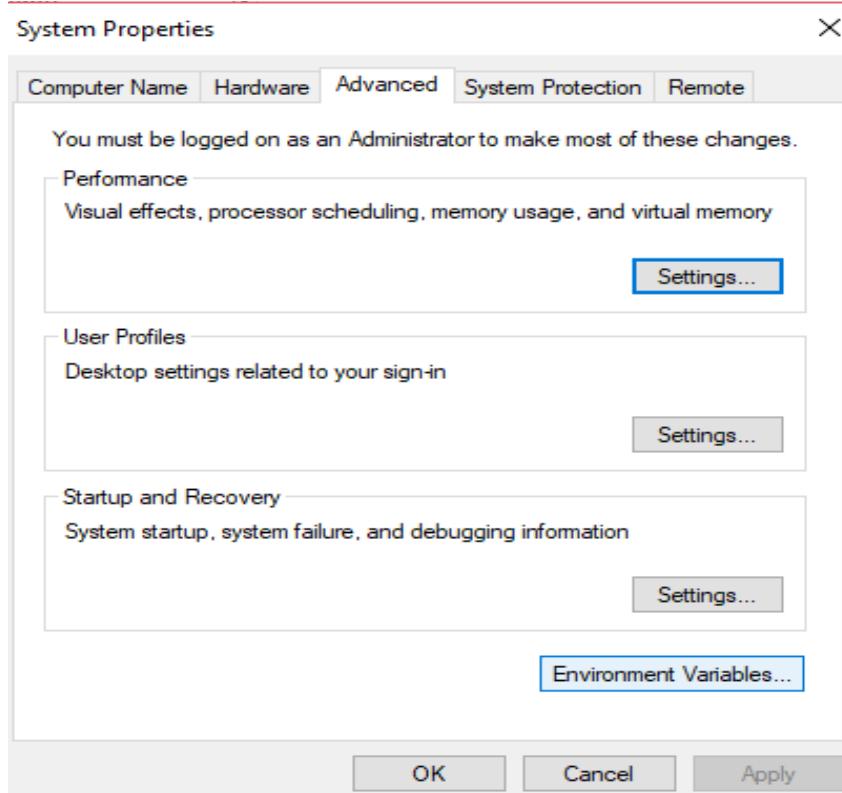
Jika tampil versi python yang digunakan seperti pada gambar diatas, maka instalasi python telah berhasil. Namun jika belum, di cek terlebih dahulu pada bagian *Environment Variables* sistem operasi windows.

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



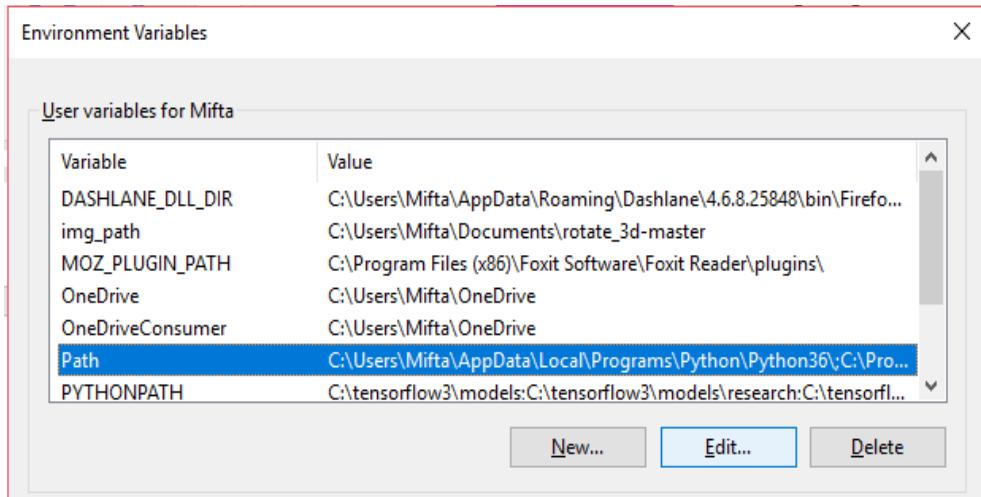
Gambar 4. 36 System Pada Pc

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



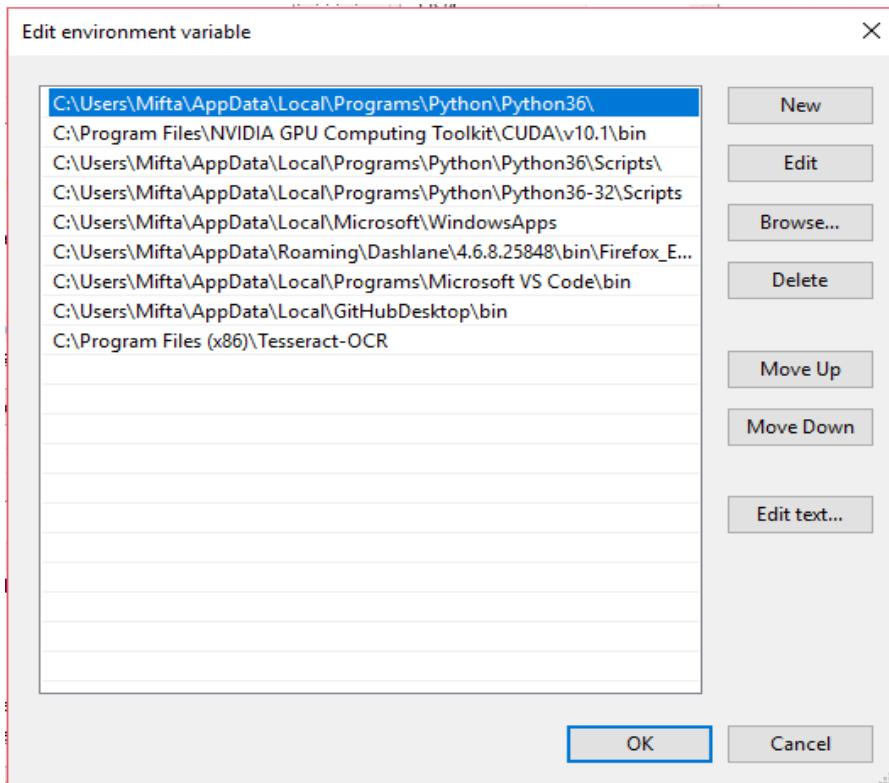
Gambar 4. 37 Konfigurasi Python pada Windows

- 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

BAB V

PENJELASAN, INSTALASI LIBRARY & FRAMEWORK 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 :

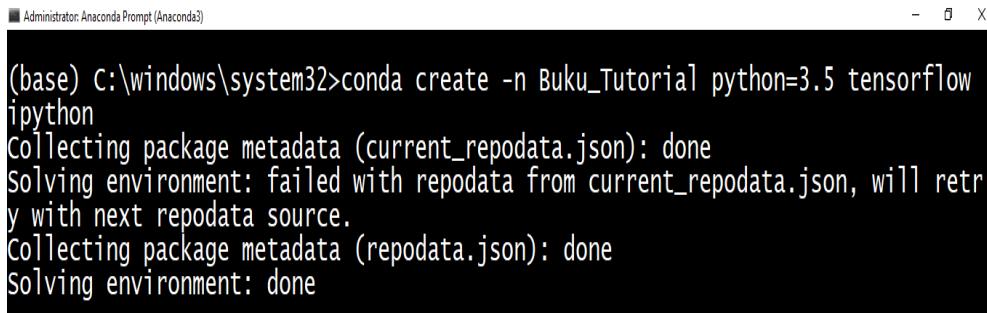


The screenshot shows a black terminal window titled "Administrator: Anaconda Prompt (Anaconda3)". The prompt "(base) C:\Windows\system32>" is visible at the top left. The rest of the screen is completely black, indicating no further output or input.

Gambar 5. 1 Tampilan Anaconda Prompt

2. Untuk membuat environment baru dalam Anaconda Anda dapat menggunakan perintah sebagai berikut :

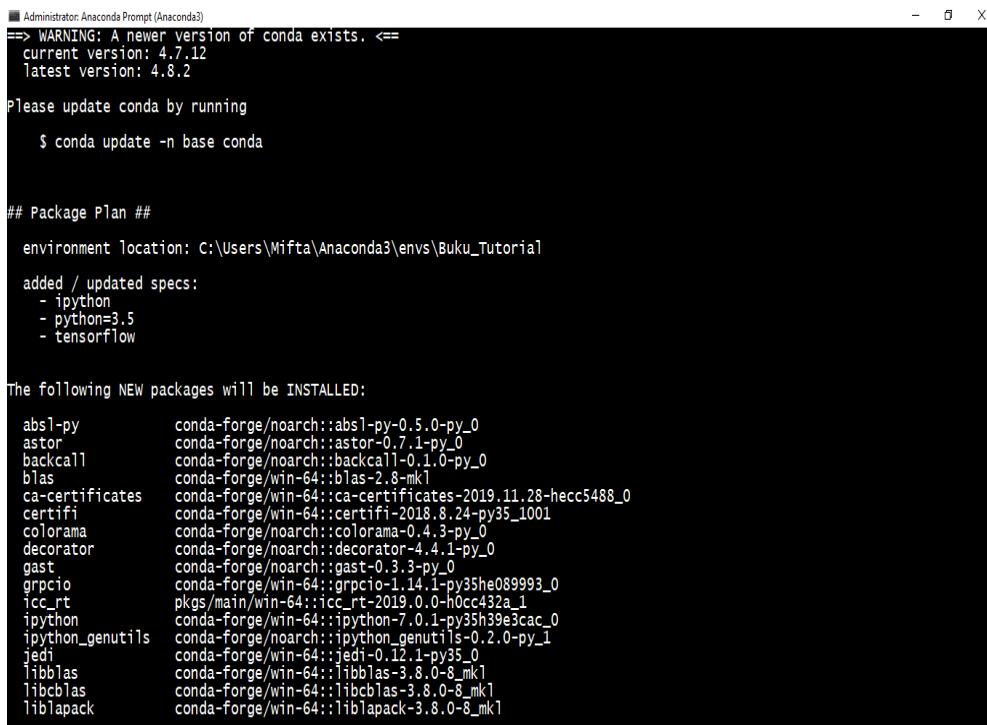
“conda create -n Buku Tutorial python=3.5 tensorflow ipython”



```
(base) C:\windows\system32>conda create -n Buku_Tutorial python=3.5 tensorflow ipython
Collecting package metadata (current_repodata.json): done
Solving environment: failed with repodata from current_repodata.json, will retry with next repodata source.
Collecting package metadata (repodata.json): done
Solving environment: done
```

Gambar 5. 2 Membuat environment Dengan Nama “Buku_Tutorial”

3. Tunggu hingga proses pembuatan environment-nya selesai kemudian jalankan perintah yang untuk menginstall package yang disediakan anaconda



```
Administrator: Anaconda Prompt (Anaconda3)
=> WARNING: A newer version of conda exists. <=
  current version: 4.7.12
  latest version: 4.8.2

Please update conda by running

$ conda update -n base conda

## Package Plan ##

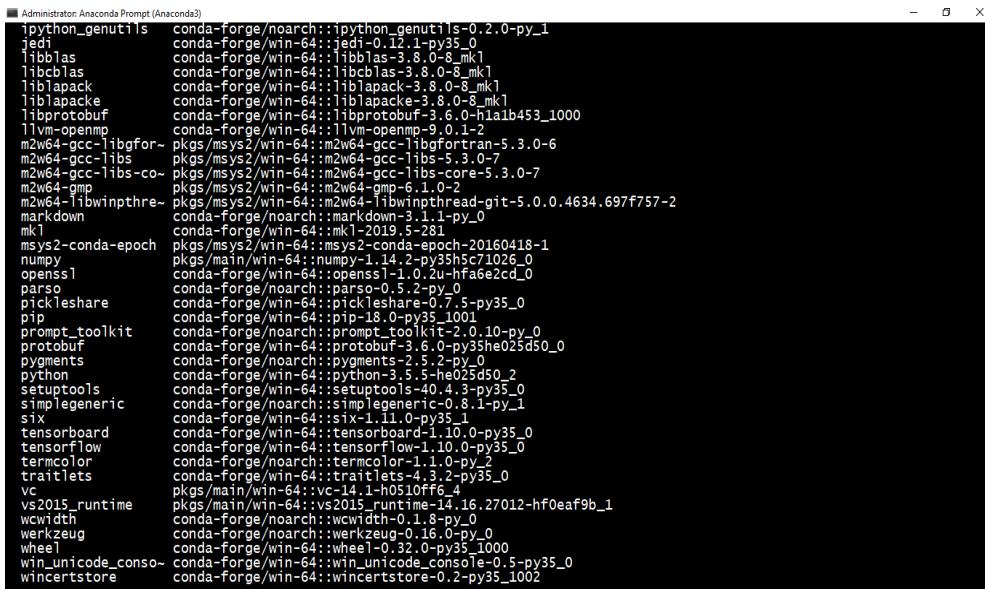
environment location: C:\Users\Mifta\Anaconda3\envs\Buku_Tutorial

added / updated specs:
- ipython
- python=3.5
- tensorflow

The following NEW packages will be INSTALLED:

absl-py      conda-forge/noarch::absl-py-0.5.0-py_0
astor        conda-forge/noarch::astor-0.7.1-py_0
backcall     conda-forge/noarch::backcall-0.1.0-py_0
blas         conda-forge/win-64::blas-2.8-mkl
ca-certificates  conda-forge/win-64::ca-certificates-2019.11.28-hecc5488_0
certifi      conda-forge/win-64::certifi-2019.8.24-py35_1001
colorama     conda-forge/noarch::colorama-0.4.3-py_0
decorator    conda-forge/noarch::decorator-4.4.1-py_0
gast         conda-forge/noarch::gast-0.3.3-py_0
grpcio       conda-forge/win-64::grpcio-1.14.1-py35he089993_0
icc_rt        pkgs/main/win-64::icc_rt-2019.0.0-h0cc432a_1
ipython       conda-forge/win-64::ipython-7.0.1-py35h39e3cac_0
ipython_genutils  conda-forge/noarch::ipython_genutils-0.2.0-py_1
jedi          conda-forge/win-64::jedi-0.12.1-py35_0
libblas       conda-forge/win-64::libblas-3.8.0-8_mkl
libcbblas    conda-forge/win-64::libcbblas-3.8.0-8_mkl
liblapack    conda-forge/win-64::liblapack-3.8.0-8_mkl
```

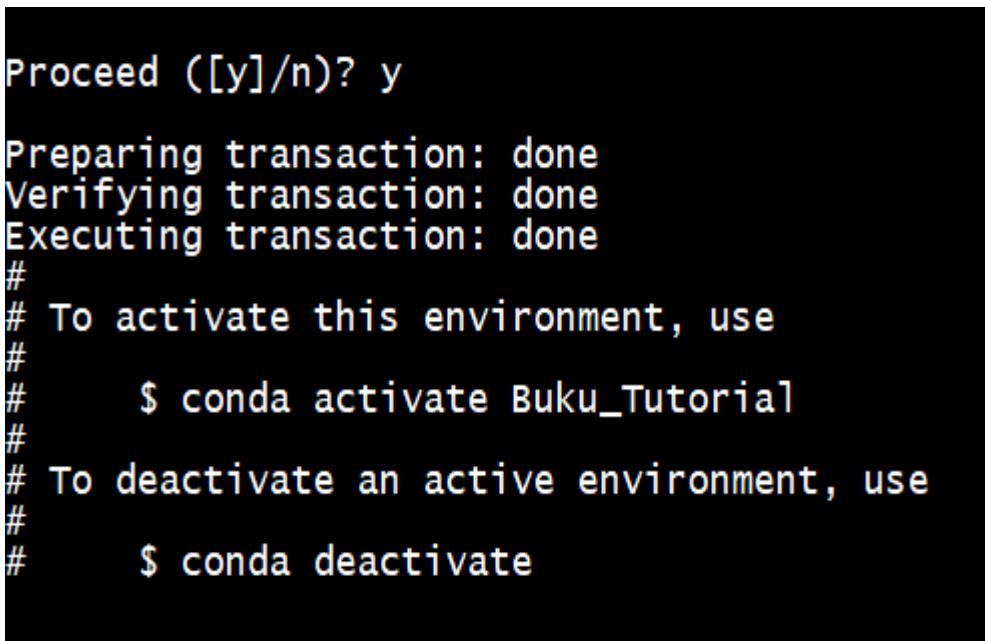
Gambar 5. 3 Proses Membuat Environment



```
Administrator: Anaconda Prompt (Anaconda3)
ipython_genutils    conda-forge/noarch::ipython_genutils-0.2.0-py_1
jedi                conda-forge/win-64::jedi-0.12.1-py35_0
libblas              conda-forge/win-64::libblas-3.8.0-8_mkl
libccblas            conda-forge/win-64::libccblas-3.8.0-8_mkl
liblapack             conda-forge/win-64::liblapack-3.8.0-8_mkl
liblapacke            conda-forge/win-64::liblapacke-3.8.0-8_mkl
libprotobuf           conda-forge/win-64::libprotobuf-3.6.0-h1a1b453_1000
llvm-openmp           conda-forge/win-64::llvm-openmp-9.0.1-2
m2w64-gcc-libgfortran- pkgs/msys2/win-64::m2w64-gcc-libgfortran-5.3.0-6
m2w64-gcc-libs        pkgs/msys2/win-64::m2w64-gcc-libs-5.3.0-7
m2w64-gcc-libs-core- pkgs/msys2/win-64::m2w64-gcc-libs-core-5.3.0-7
m2w64-gmp              pkgs/msys2/win-64::m2w64-gmp-6.1.0-2
m2w64-libwinpthread- pkgs/msys2/win-64::m2w64-libwinpthread-git-5.0.0.4634.697f757-2
markdown              conda-forge/noarch::markdown-3.1.1-py_0
mkl                  conda-forge/win-64::mkl-2019.5-281
msys2-conda-epoch      pkgs/msys2/win-64::msys2-conda-epoch-20160418-1
numpy                 pkgs/main/win-64::numpy-1.14.2-py35h5c7026_0
openssl               conda-forge/win-64::openssl-1.0.2u-hfa6e2cd_0
parso                 conda-forge/noarch::parso-0.5.2-py_0
pickleshare            conda-forge/win-64::pickleshare-0.7.5-py35_0
pip                   conda-forge/win-64::pip-18.0-py35_1001
prompt_toolkit         conda-forge/noarch::prompt_toolkit-2.0.10-py_0
protobuf               conda-forge/win-64::protobuf-3.6.0-py35he025d50_0
pygments              conda-forge/noarch::pygments-2.5.2-py_0
python                 conda-forge/win-64::python-3.5.5-he025d50_2
setuptools             conda-forge/win-64::setuptools-40.4.3-py35_0
simplegeneric          conda-forge/noarch::simplegeneric-0.8.1-py_1
six                   conda-forge/win-64::six-1.11.0-py35_1
tensorflowboard        conda-forge/win-64::tensorflowboard-1.10.0-py35_0
tensorflow             conda-forge/win-64::tensorflow-1.10.0-py35_0
termcolor               conda-forge/noarch::termcolor-1.1.0-py_2
traitlets              conda-forge/win-64::traitlets-4.3.2-py35_0
vc                     pkgs/main/win-64::vc-14.1-h0510ff6_4
vs2015_runtime          pkgs/main/win-64::vs2015_runtime-14.16.27012-hf0eaf9b_1
wcwidth                conda-forge/noarch::wcwidth-0.1.8-py_0
werkzeug               conda-forge/noarch::werkzeug-0.16.0-py_0
wheel                  conda-forge/win-64::wheel-0.32.0-py35_1000
win_unicode_console- pkgs/main/win-64::win_unicode_console-0.5-py35_0
wincertstore            conda-forge/win-64::wincertstore-0.2-py35_1002
```

Gambar 5. 4 Proses Membuat Environment

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



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

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