

**BUKU TUTORIAL OBJECT DETECTION**

**“KLASIFIKASI GENDER ATAU JENIS  
KELAMIN MENGGUNAKAN METODE *YOLO*  
(*You Only Look Once*)”**

Buku ini dibuat untuk memenuhi persyaratan kelulusan  
matakuliah Program Internship I



**Oleh,**

**1.16.4.035     “Eko Cahyono Putro”**

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA  
POLITEKNIK POS INDONESIA  
BANDUNG  
2019**

## KATA PENGANTAR

---

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, hidayah serta kekuatan sehingga laporan internship 1 ini dengan judul “KLASIFIKASI GENDER ATAU JENIS KELAMIN MENGGUNAKAN METODE *YOLO (You Only Look Once)*” dapat terselesaikan.

Banyak kendala yang dihadapi dalam penyusunan Laporan Internship I ini dan penulis menyadari bahwa penyusunan laporan 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 proposal intership 1.
2. Kedua orang tua dan keluarga penulis yang telah mendorong dan memberi semangat kepada penulis.
3. Ibu Sendylenvi Regia selaku pembimbing eksternal di PT.Telekomunikasi Indonesia Tbk (Telkom).
4. Bapak Rolly Maulana Awangga, S.T., M.T. selaku dosen pembimbing internship I.
5. Ibu Nisa Hanum Harani, S.Kom., M.T.selaku Koordinator Internship I.
6. Bapak M. Yusril Helmi Setyawan, S.Kom., M.Kom. selaku Ketua Prodi D4 Teknik Informatika.
7. Teman-teman serta rekan yang ada di kampus maupun di P.T. Telekomunikasi Indonesia Tbk(Telkom)

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

Bandung, 20 Januari 2020

Penulis

## **DAFTAR ISI**

---

## **DAFTAR GAMBAR**

---

## DAFTAR TABEL

---

# BAB I

---

## Pengenalan Artificial Intelligence, Machine Learning, Neureal Network dan Deep Learning.

### 1.1 Pengantar Artificial Intelligence

#### 1.1.1 Pengertian Artificial Intelligence (Kecerdasan Buatan)

Menurut Stuart J. Russel dan Peter Norvig, AI atau kecerdasan buatan bisa dipahami sebagai sebuah perangkat komputer yang mampu memahami lingkungan di sekitarnya, sekaligus memberikan respons yang sesuai dengan tujuan tindakannya tersebut. Dalam hal ini, Minsky memberikan pengertian yang hampir sama. Menurutnya, kecerdasan buatan adalah satu ilmu yang mempelajari cara membuat komputer melakukan atau memproduksi tindakan sama seperti yang dilakukan manusia.



*Gambar 1.1 1 Stuart J. Russel dan Peter Norvig*

Melihat dua pengertian di atas, bisa disimpulkan jika Artificial Intelligence adalah satu ilmu untuk merancang, membangun, dan mengonstruksi satu mesin (komputer) atau program komputer hingga memiliki kecerdasan layaknya manusia. Kecerdasan dalam hal ini adalah kemampuan untuk mengambil tindakan, atau menyelesaikan masalah layaknya manusia menggunakan kecerdasannya.

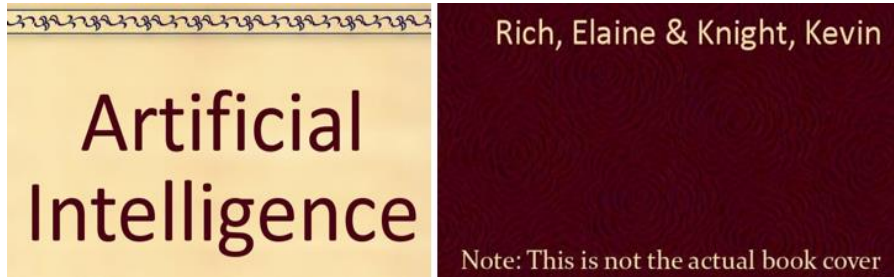
Lingkup penelitian Artificial Intelligence meliputi banyak aspek kemampuan kecerdasan manusia seperti penalaran, pengetahuan, perencanaan, pembelajaran, pemrosesan bahasa alami, hingga kemampuan untuk memanipulasi objek. Akhirnya, Artificial Intelligence diharapkan bisa menjadi sebuah mesin yang benar-benar memiliki kecerdasan umum layaknya manusia. Adapun definisi Artificial Intelligence atau kecerdasan buatan dari beberapa pendapat para tokoh yaitu:

“Kecerdasan buatan (Artificial Intelligence) merupakan kawasan penelitian, aplikasi dan instruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang –dalam pandangan manusia adalah– cerdas”– H. A. Simon (1987)



*Gambar 1.1 2 H. A. Simon*

“Kecerdasan Buatan (AI) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia”– Rich and Knight (1991)



*Gambar 1.1 3 Book from elaine & kevin*

“Bagian dari ilmu komputer yang mempelajari bagaimana membuat mesin (komputer) dapat melakukan pekerjaan seperti dan sebaik yang dilakukan oleh manusia bahkan bisa lebih baik daripada yang dilakukan manusia”– Idhawati Hestinationsih

Dari beberapa definisi diatas maka dapat ditarik kesimpulan bahwa AI (Artificial Intelligence) / kecerdasan buatan merupakan sebuah teknologi komputer atau mesin yang memiliki kecerdasan layaknya manusia. Sederhananya gini sebuah instruksi pintar yang diberikan kepada program maupun mesin.

Artificial intelligence adalah suatu konsep yang sebenarnya sudah muncul sejak jaman dahulu tepatnya di zaman Yunani Kuno. Pada zaman Yunani Kuno sudah ada Talos, raksasa dari perunggu dan Galatea, patung buatan Pygmalion yang dibuat dengan konsep kecerdasan buatan.



Konsepnya sudah muncul sejak jaman dahulu tapi istilah ini baru muncul di pertengahan abad 20 tepatnya di tahun 1950. Seorang ahli matematika, Alan Turing lah salah satu yang ikut berperan dalam mencetuskan ide tentang teknologi artificial intelligence. Karena idenya tersebut, istilah kecerdasan buatan atau AI menjadi populer di kalangan peneliti dan ilmuwan.

Ide tentang teknologi dengan kecerdasan buatan itu pun terus dikembangkan dan diteliti sehingga bisa dimanfaatkan sepenuhnya oleh manusia. Nah Anda sekarang pasti bisa merasakan perkembangan penggunaan kecerdasan buatan, salah satu contohnya adalah teknologi pintar, adanya prediksi di mesin pencarian, ramalan cuaca, penyaring email yang termasuk spam, sugesti aplikasi, dan masih banyak lainnya.

*Artificial Intelligence* atau hanya disingkat *AI*) didefinisikan sebagai kecerdasan *entitas* ilmiah. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Beberapa macam bidang yang menggunakan kecerdasan buatan antara lain sistem pakar, permainan komputer (*games*), *logika fuzzy*, jaringan saraf tiruan dan robotika.

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 dalam AI menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas. Termasuk contohnya adalah pengendalian,

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.

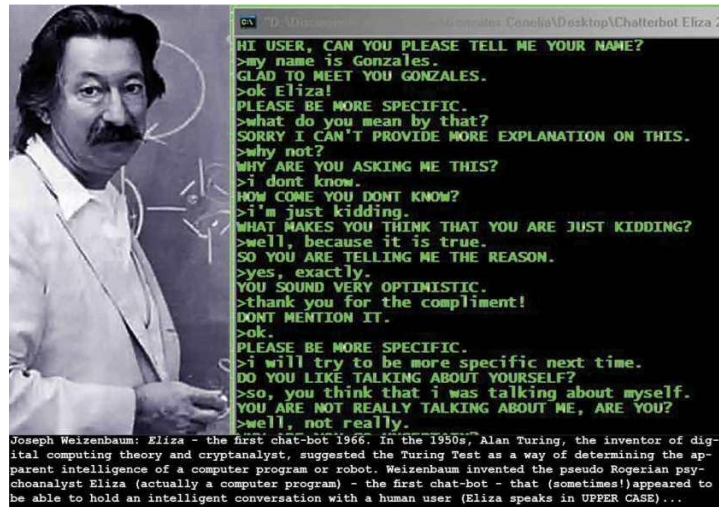
### **1.1.2 Sejarah Artificial Intelligence (Kecerdasan Buatan)**

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 fondasi untuk jaringan saraf.

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.



*Gambar 1.1 4 Joseph Weizenbaum*

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 mendemostrasikan batas jaringan saraf 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

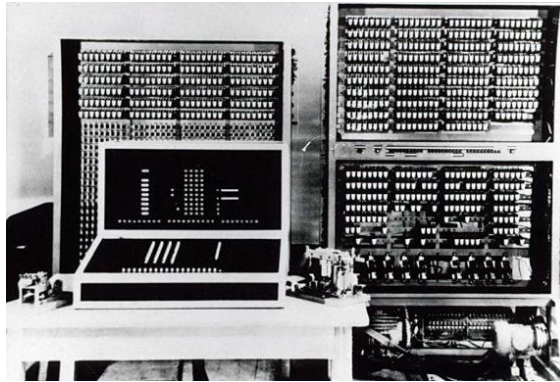
terkendali komputer pertama untuk mengatasi jalan berintang yang kusut secara mandiri.

Pada tahun 1980-an, jaringan saraf digunakan secara meluas dengan algoritme 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 saraf. Para ahli psikologi, David Rumelhart dan Geoff Hinton, melanjutkan penelitian mengenai model jaringan saraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan kembali algoritme pembelajaran propagansi balik (Back-Propagation learning). Algoritme 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.

Adapun sejarah mengenai Kecerdasan Buatan dari tahun ke tahun yaitu:

- **Era komputer elektronik (1941)**

Ditemukannya pertama kali alat penyimpanan dan pemrosesan informasi yang disebut komputer elektronik. Penemuan ini menjadi dasar pengembangan program yang mengarah ke AI.



*Gambar 1.1 5 Era komputer elektronik*

- **Masa persiapan AI (1943-1956)**

Tahun 1943, Warren McCulloch dan Walter Pitts berhasil membuat suatu model saraf tiruan di mana setiap neuron digambarkan sebagai 'on' dan 'off'. Mereka menunjukkan bahwa setiap fungsi dapat dihitung dengan suatu jaringan sel saraf dan bahwa semua hubungan logis dapat diimplementasikan dengan struktur jaringan yang sederhana. Pada tahun 1950, Norbert Wiener membuat penelitian mengenai prinsip-prinsip teori feedback. Contoh terkenal adalah thermostat.

Pada tahun 1956, John McCarthy meyakinkan Minsky, Claude Shannon, dan Nathaniel Rochester untuk membantunya melakukan penelitian dalam bidang automata, jaringan saraf, dan pembelajaran intelijensia. Mereka mengerjakan proyek ini selama 2 bulan di Universitas Dartmouth. Hasilnya adalah program yang mampu berpikir non-numerik dan menyelesaikan masalah pemikiran, yang dinamakan Principia

Mathematica. Hal ini menjadikan McCarthy disebut sebagai father of Artificial Intelligence/ Bapak Kecerdasan Buatan.

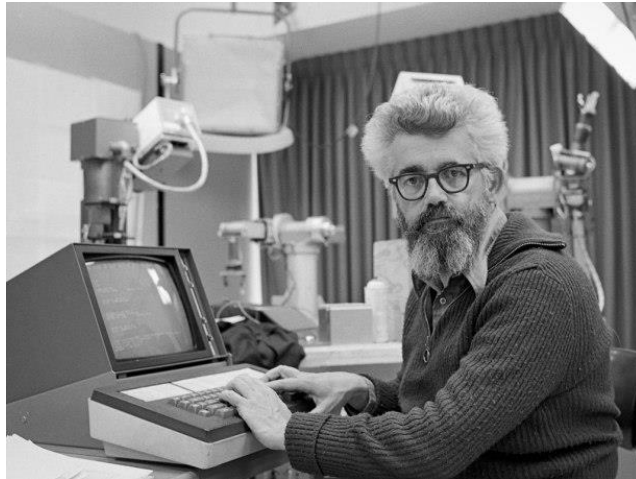


*Gambar 1.1 6 Bapak Kecerdasan Buatan.*

- **Awal perkembangan (1952-1969)**

Pada tahun 1958, McCarthy di MIT AI Lab mendefinisikan bahasa pemrograman tingkat tinggi yaitu LISP, yang sekarang mendominasi pembuatan program-program AI. Kemudian, McCarthy membuat program yang dinamakan programs with common sense. Di dalam program tersebut, dibuat rancangan untuk menggunakan pengetahuan dalam mencari solusi. Pada tahun 1959, Program komputer General Problem Solver berhasil dibuat oleh Herbert A. Simon, J.C. Shaw, dan Allen Newell. Program ini dirancang untuk memulai penyelesaian masalah secara manusiawi. Pada tahun yg sama Nathaniel Rochester dari IBM dan para mahasiswanya merilis program AI yaitu geometry theorem prover. Program ini dapat membuktikan suatu teorema menggunakan aksioma-aksioma yang ada. Pada tahun 1963, program yang dibuat James Slagle

mampu menyelesaikan masalah integral tertutup untuk mata kuliah Kalkulus. Pada tahun 1968, program analogi buatan Tom Evan menyelesaikan masalah analogi geometri yang ada pada tes IQ.



*Gambar 1.1 7 John Mccarthy*

- **Perkembangan AI melambat (1966-1974)**

Perkembangan AI melambat disebabkan adanya beberapa kesulitan yang di hadapi seperti Program-program AI yang bermunculan hanya mengandung sedikit atau bahkan tidak mengandung sama sekali pengetahuan pada subjeknya, banyak terjadi kegagalan pada pembuatan program AI, terdapat beberapa batasan pada struktur dasar yang digunakan untuk menghasilkan perilaku inteligensia.

- **Sistem berbasis pengetahuan (1969-1979)**

Pada tahun 1960an, Ed Feigenbaum, Bruce Buchanan, dan Joshua Lederberg merintis proyek DENDRAL yaitu program

untuk memecahkan masalah struktur molekul dari informasi yang didapatkan dari spectrometer massa. Dari segi diagnosa medis juga sudah ada yang menemukan sistem berbasis Ilmu pengetahuan, yaitu Saul Amarel dalam proyek computer ini biomedicine. Proyek ini diawali dari keinginan untuk mendapatkan diagnosa penyakit berdasarkan pengetahuan yang ada pada mekanisme penyebab proses penyakit.



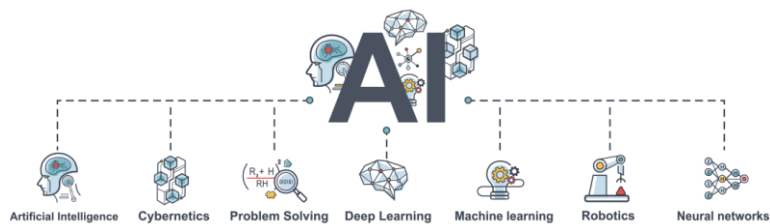
*Gambar 1.1 8 Ed Feigenbaum bersama anggota direksi The Computation Center*

- **AI menjadi sebuah industry (1980-1988)**

Industralisasi AI diawali dengan ditemukannya sistem pakar yang dinamakan R1 yang mampu mengkonfigurasi sistem-sistem komputer baru. Program tersebut mulai dioperasikan di Digital Equipment Corporation (DEC), McDermott, pada tahun 1982. Pada tahun 1986, program ini telah berhasil menghemat US\$ 40 juta per tahun.



Pada tahun 1988, kelompok AI di DEC menjalankan 40 sistem pakar. Hampir semua perusahaan besar di USA mempunyai divisi Ai sendiri yang menggunakan ataupun mempelajari sistem pakar. Booming industry Ai juga melibatkan perusahaan-perusahaan besar seperti Carnegie Group, Inference, IntelliCorp, dan Technoledge yang menawarkan software tools untuk membangun sistem pakar. Perusahaan hardware seperti LISP Machines Inc., Texas Instruments, Symbolics, dan Xerox juga turut berperan dalam membangun workstation yang dioptimasi untuk pembangunan program LISP. Sehingga, perusahaan yang sejak tahun 1982 hanya menghasilkan beberapa juta US dollar per tahun meningkat menjadi 2 milyar US dollar per tahun pada tahun 1988.

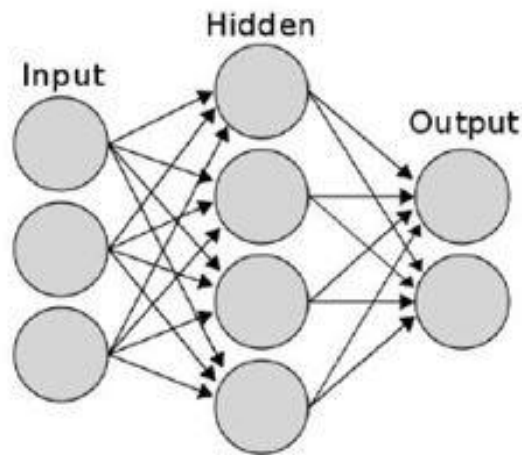


*Gambar 1.1 9 ilustrasi indusry AI*

- **Kembalinya jaringan saraf tiruan (1986 – sekarang)**

Meskipun bidang ilmu komputer menolak jaringan saraf tiruan setelah diterbitkannya buku ‘Perceptrons’ karangan Minsky dan Papert, tetapi para ilmuwan masih mempelajari bidang ilmu tersebut dari sudut pandang yang lain, yaitu fisika. Ahli fisika seperti Hopfield (1982) menggunakan teknik-teknik mekanika statistika untuk menganalisa sifat-sifat penyimpanan

dan optimasi pada jaringan saraf. Para ahli psikolog, David Rumelhart dan Geoff Hinton melanjutkan penelitian mengenai model jaringan saraf pada memori. Pada tahun 1985-an sedikitnya empat kelompok riset menemukan algoritma Back-Propagation. Algoritma ini berhasil diimplementasikan ke dalam ilmu bidang komputer dan psikologi.



*Gambar 1.1 10 jaringan saraf tiruan*

### **1.1.3 Tujuan Artificial Intelligence (Kecerdasan Buatan)**

Artificial intelligence adalah suatu perkembangan teknologi yang tujuan dibentuknya karena beberapa hal berikut ini.

- **Membentuk teknologi atau mesin yang lebih pintar**

Tujuan utama dikembangkannya kecerdasan buatan yaitu untuk membuat suatu teknologi atau mesin menjadi lebih pintar sehingga akan memudahkan pekerjaan manusia. Contoh gampangnya bisa anda lihat di komputer. Selain untuk mengetik, komputer sekarang dikembangkan menjadi lebih

pintar sehingga bisa untuk bermain game, mengedit video dan foto, serta bisa digunakan untuk banyak kegiatan bermanfaat lainnya.

- **Memahami Kecerdasan**

Tujuan ilmiah dibentuknya teknologi kecerdasan ini ialah untuk membuat suatu mesin yang mampu memahami kecerdasan seperti bisa memecahkan masalah dengan lebih cepat, lebih teliti, lebih efektif, dan efisien. Adanya AI ini tentu memudahkan semua orang.

- **Membuat suatu teknologi yang lebih bermanfaat**

Dari segi entrepreneurial kecerdasan buatan mampu membuat suatu usaha mendapatkan hasil lebih maksimal karena teknologi dengan AI akan mempermudah pekerjaan. Suatu pekerjaan akan terasa ringan dan bisa selesai dalam waktu cepat. AI juga mampu mengumpulkan dan menganalisis suatu data sehingga anda bisa mengetahui peluang terbaru untuk bisnis anda.

Tujuan pembuatan mesin yang canggih ini adalah untuk mengurangi waktu kerja sehingga kegiatan dapat lebih efisien. Dengan menggunakan kecerdasan buatan, manusia menjadi lebih mudah dalam mengambil keputusan dan melakukan kegiatannya dibandingkan pada masa lalu.

Bukan hanya lebih mudah penggunaan kecerdasan buatan dalam komputer juga membuat waktu pengerjaan sebuah kegiatan menjadi lebih cepat. Contohnya. Dahulu untuk melakukan searching di mesin

pencari kita perlu mengetikkan satu persatu huruf di kolom pencarian bukan?

Saat ini mesin pencarian seperti Google sudah memiliki implementasi kecerdasan buatan sehingga hanya dengan berbicara dengan memerintah melalui suara ponsel anda sudah bisa menjalankan perintah anda.

#### **1.1.4 Keuntungan dan Kerugian Artificial Intelligence (Kecerdasan Buatan)**

Setelah mengetahui beberapa tujuan dibentuknya kecerdasan buatan, anda pasti penasaran bukan apa rahasia kemampuannya? Nah penjelasan di bawah ini mungkin bisa membantu anda memahaminya.

- **Menganalisis**

Kemampuan yang pertama adalah teknologi yang mampu menganalisis. Algoritma canggih yang digunakan dalam AI membuat AI mampu menganalisis data yang dikumpulkannya. Kemampuan menganalisis data ini bisa dilihat contohnya dalam prediksi aplikasi yang anda suka dalam play store.

Saran aplikasi yang muncul dalam akun play store anda merupakan hasil analisis AI setelah mengumpulkan data aplikasi yang anda install, buka, ataupun inginkan. Anda bisa menemukan contoh lainnya di aplikasi kalender periode haid wanita. Di aplikasi tersebut, semakin lama anda menggunakan, prediksi akan semakin tepat karena data yang dikumpulkan semakin banyak.

Dalam urusan bisnis skala besar, AI juga bisa digunakan dalam menganalisis prediksi penjualan yang dilakukan oleh suatu perusahaan, tingkat konsumsi masyarakat, bisa menganalisis tingkat keuntungan, dan masih banyak lainnya.

- **Mengelola Big Data**

Kemampuan selanjutnya ialah teknologi yang mampu mengolah big data. Big data merupakan istilah yang digunakan untuk menggambarkan data yang sangatlah besar. Data yang dimaksud bisa data apa saja yang membanjiri suatu bisnis. Contohnya adalah instagram.

Di instagram akan ada banyak data masuk baik data pengguna, konten yang sering dilihat, akun yang diikuti oleh pengguna, dan banyak data lainnya. Dengan menggunakan kecerdasan buatan, para pemilik bisnis besar akan mendapatkan kemudahan dan keuntungan dalam menentukan keputusan berbisnis selanjutnya.

Anda bisa melihat contoh penerapan artificial intelligence di instagram. Konten yang anda lihat akan mempengaruhi konten yang muncul di halaman saran. Iklan yang muncul pun akan berkaitan dengan konten atau akun yang sering anda lihat. Intinya aktivitas anda di instagram akan direkam, dikumpulkan, dan diolah oleh AI sehingga menghasilkan tampilan yang sesuai dengan minat dan kesenangan anda.

- **Merespon dengan cepat**

Kemampuan lain dari kecerdasan buatan adalah mampu merespon dengan cepat. Contoh kemampuan merespon bisa anda

lihat di email balasan otomatis atau chatbot yang biasanya ada di situs e-commerce. Saat anda tanya di chatbot, akan ada balasan cepat. Nah balasan tersebut merupakan salah satu bentuk penggunaan AI. Adanya AI dengan respon cepat ini lama kelamaan mampu menggantikan pekerjaan seorang customer service.

- **Memperbaiki suatu dokumen dengan akurat**

AI juga mampu memperbaiki suatu dokumen dengan akurat. Saat ini banyak ditemui aplikasi atau situs yang bisa mengetahui kesalahan penulisan maupun pengetikan suatu dokumen. Tidak hanya menemukan kesalahan, perbaikan dokumen pun bisa dilakukan. Akan ada saran akan kata yang tepat untuk menggantikan kata yang salah. Adanya AI ini tentu mengancam keberadaan editor karena AI lebih teliti dibanding editor.

Sangatlah banyak keuntungan yang bisa didapatkan dari teknologi tersebut. keuntungan yang didapat antara lain yaitu:

- **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.

- **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.

- **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.

- **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.

Artificial Intelegen adalah teknologi masa depan, namun tetap saja sebuah teknologi pasti ada kelemahan tersendiri dan berikut adalah kelemahan dari artificial intelegence.

Buatan manusia pasti masih memiliki sedikit cela meskipun hanya sedikit. *Nah* 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.

AI tidak memiliki common sense. common sense sendiri merupakan sesuatu yang membuat kita tidak sekedar memproses informasi, namun kita mengerti informasi tersebut. Kemengertian ini hanya dimiliki oleh kita sebagai manusia.

Kecerdasan AI terbatas pada apa yang diberikan kepadanya (terbatas pada program yang diberikan). Alat teknologi artificial intelligence tidak dapat mengolah informasi yang tidak ada dalam sistemnya. Sebagaimana sistem yang hanya digunakan untuk mengenali suara manusia dalam bahasa indonesia, selamanya tidak akan pernah mampu mengenali bahasa yunani tanpa ada fungsi didalamnya.

Mengapa kecerdasan buatan itu penting berikut beberapa penjelasan mengapa kecerdasan buatan itu penting.



- **AI mengautomasi pembelajaran dan penemuan berulang melalui data.** Tetapi AI berbeda dengan automasi robotik yang digerakkan oleh perangkat keras. Alih-alih mengautomasi tugas manual, AI melakukan tugas-tugas yang sering, bervolume tinggi, terkomputerisasi dengan andal dan tanpa mengalami kelelahan. Untuk jenis automasi ini, penyelidikan manusia masih penting untuk mengatur sistem dan mengajukan pertanyaan yang tepat.
- **AI menambahkan kecerdasan** pada produk-produk yang ada. Di sebagian besar kasus, AI tidak dijual sebagai aplikasi individu. Akan tetapi, produk yang sudah Anda gunakan akan ditingkatkan dengan kemampuan AI, mirip seperti Siri yang ditambahkan sebagai fitur pada generasi baru produk Apple. Automasi, platform percakapan, bot, dan mesin pintar dapat dikombinasikan dengan sejumlah besar data untuk meningkatkan banyak teknologi di rumah dan di tempat kerja, mulai dari intelijen keamanan hingga analisis investasi.
- **AI beradaptasi melalui algoritme pembelajaran progresif** guna memungkinkan data melakukan pemrograman. AI menemukan struktur dan keteraturan dalam data sehingga algoritme memperoleh keterampilan: Algoritme menjadi pengklasifikasi atau prediktor. Jadi, sama seperti algoritme yang dapat mengajarkan dirinya sendiri cara bermain catur, AI dapat mengajarkan sendiri produk apa yang akan direkomendasikan berikutnya secara online. Dan model-model beradaptasi saat memberikan data baru. Propagasi belakang merupakan teknik AI yang memungkinkan model untuk beradaptasi, melalui

pelatihan dan data yang ditambahkan, saat jawaban pertama tidak terlalu tepat.

- **AI menganalisis data lebih banyak dan lebih dalam** menggunakan jaringan neural yang memiliki banyak lapisan tersembunyi. Membangun sistem deteksi penipuan dengan lima lapisan tersembunyi hampir tidak mungkin beberapa tahun yang lalu. Semuanya berubah dengan kekuatan komputer yang luar biasa dan big data. Anda memerlukan banyak data untuk melatih model pembelajaran mendalam karena model tersebut belajar langsung dari data. Semakin banyak data yang Anda umpankan kepada model, semakin akurat model tersebut.
- **AI mencapai keakuratan mengagumkan** melalui jaringan neural mendalam – yang sebelumnya tidak dimungkinkan. Misalnya, interaksi Anda dengan Alexa, Google Search, dan Google Photos semuanya didasarkan pada pembelajaran yang mendalam – dan ketiganya terus menjadi semakin akurat karena kita semakin sering menggunakannya. Di bidang medis, teknik AI dari pembelajaran mendalam, klasifikasi citra, dan pengenalan objek sekarang dapat digunakan untuk menemukan kanker pada MRI dengan akurasi yang sama seperti ahli radiologi yang terlatih.
- **AI memanfaatkan sebagian besar data.** Jika algoritme merupakan pembelajaran mandiri, data itu sendiri dapat menjadi kekayaan intelektual. Jawabannya ada dalam data; Anda hanya perlu menerapkan AI untuk mendapatkannya. Karena peran data kini semakin penting dari sebelumnya, data dapat menciptakan

keunggulan kompetitif. Jika Anda memiliki data terbaik dalam industri kompetitif, bahkan jika seseorang menerapkan teknik serupa, data terbaiklah yang akan menang.

### **1.1.5 Jenis Artificial Intelligence (Kecerdasan Buatan)**

AI memiliki 2 kategori yaitu lemah atau kuat. AI lemah (weak AI) yang juga dikenal sebagai AI sempit adalah system AI yang dirancang dan dilatih untuk tugas tertentu. Asisten pribadi virtual, seperti Apple Siri, adalah bentuk AI yang lemah. Sedangkan AI kuat (strong AI), juga dikenal sebagai kecerdasan buatan umum adalah system AI dengan kemampuan kognitif manusia secara umum. Ketika disajikan dengan tugas khusus, system AI kuat dapat menemukan solusi tanpa campur tangan manusia.

Arend Hintze, asisten profesor biologi integratif dan ilmu komputer dan teknik di Michigan State University, mengkategorikan AI menjadi 4 jenis, dari jenis sistem AI yang ada saat ini hingga sistem yang hidup, yang belum ada. Kategorinya adalah sebagai berikut:

- Tipe 1: Mesin reaktif.

Contohnya, Deep Blue, program catur IBM yang mengalahkan Garry Kasparov pada 1990-an. Deep Blue dapat mengidentifikasi bagian-bagian di papan catur dan membuat prediksi, tetapi ia tidak memiliki ingatan dan tidak dapat menggunakan pengalaman masa lalu untuk memberi tahu langkah berikutnya.



*Gambar 1.1 11 Deep Blue dan Garry Kasparov pada 1990-an*

Ini menganalisis kemungkinan langkah lawan dan dirinya sendiri serta memilih langkah paling strategis. Deep Blue dan GoogleGOGO dirancang untuk tujuan yang sempit dan tidak dapat dengan mudah diterapkan pada situasi lain.

- Tipe 2: Memori terbatas.

Sistem AI ini dapat menggunakan pengalaman masa lalu untuk menginformasikan keputusan masa depan. Beberapa fungsi pengambilan keputusan dalam mobil self-driving dirancang dengan cara ini. Pengamatan menginformasikan tindakan yang terjadi di masa depan yang tidak terlalu jauh, seperti jalur penggantian mobil. Pengamatan ini tidak disimpan secara permanen.



*Gambar 1.1 12 mobil self-driving*

- Tipe 3: Teori pikiran.

Istilah psikologi ini mengacu pada pengertian bahwa orang lain memiliki keyakinan, keinginan sendiri dan niat yang memengaruhi keputusan yang mereka buat. AI jenis ini belum ada sampai saat ini.

- Tipe 4: Kesadaran diri.

Dalam kategori ini, 24 ystem AI memiliki rasa diri, memiliki kesadaran. Mesin dengan kesadaran diri memahami keadaan mereka saat ini dan dapat menggunakan informasi untuk menyimpulkan apa yang orang lain rasakan. AI jenis ini belum ada sampai saat ini.

#### **1.1.6 Contoh Artificial Intelligence (Kecerdasan Buatan)**

Banyak sekali contoh penggunaan artificial intelligence, apalagi dalam kehidupan sehari-hari di era modern saat ini. Berikut contoh artificial intelligence yang ada di kehidupan.

- **Mesin Pencari Google**

Banyak yang tidak menyadari bahwa dalam mesin pencariinya, Google telah menyematkan algoritma AI atau kecerdasan buatan sehingga apa yang dicari banyak orang bisa tepat sasaran, dalam artian sesuai dengan apa yang kita pikirkan. Sehingga tidak heran selama ini Google menjadi mesin pencari terbaik di dunia, dan paling banyak digunakan oleh mayoritas masyarakat.

Konsep kecerdasan buatan yang disematkan pada teknologi ini yakni mesin pencari mampu secara otomatis merayapi seluruh link dari sebuah situs. Hebatnya lagi Google juga mampu membaca karakteristik penggunaannya, terutama hal-hal yang dicari oleh pengguna. Google mampu menyajikan data yang memang tepat.

Berkat algoritma AI inilah kita menjadi lebih mudah dalam mencari informasi yang tersebar di internet. Pentingnya lagi Google juga mampu memfilter konten negatif. Sehingga ketika ada konten negatif, Google tidak akan menampilkannya kepada anda. Terlebih untuk anak yang masih berusia di bawah umur. Dengan filter konten negatif ini, orang tua pun tentunya menjadi lebih aman saat membiarkan anak menjelajahi internet. Itulah satu algoritma kecerdasan buatan luar biasa yang sudah diterapkan pada mesin pencari Google.



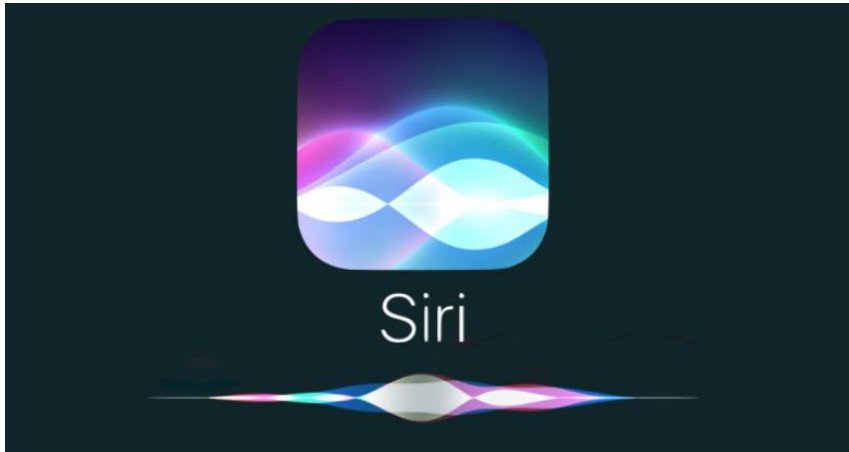
*Gambar 1.1 13 Pencarian Google*

- **SIRI**

Sama halnya dengan Google, Siri yang merupakan sebuah program cerdas pada ponsel iPhone berfungsi untuk melayani pemiliknya. Lebih tepatnya Siri bisa dibilang sebagai asisten virtual cerdas yang serba bisa dalam melayani pemiliknya untuk mengoperasikan smartphone. Dari hal sederhana hingga yang cukup sulit, seperti mengirim pesan, menemukan informasi, mencari petunjuk arah, melakukan panggilan suara, membuka aplikasi, dan masih banyak lainnya. Siri dapat melakukan itu untuk membantu anda. Pada awalnya, Siri diperkenalkan sebagai aplikasi iOS yang tersedia di App Store oleh Siri Inc, Siri Inc. kemudian diakuisisi oleh Apple Inc. pada tanggal 28 April 2010.

Siri Inc. telah mengumumkan bahwa perangkat lunak mereka akan tersedia untuk BlackBerry dan Android ponsel bertenaga, tetapi dalam upaya pengembangan semua untuk non-Apple platform dibatalkan setelah akuisisi oleh Apple Inc. Anda cukup memerintah Siri menggunakan suara, maka secara segera teknologi

besutan Apple ini akan menerjemahkan suara anda, dan mulai melakukan apa yang anda inginkan.



*Gambar 1.1 14 SIRI*

- **Tesla dan Smart car**

Nyatanya konsep teknologi buatan tidak hanya diterapkan pada aplikasi dan perangkat smartphone saja, akan tetapi juga mulai dicoba untuk membuat mobil pintar (Smart car). Contohnya adalah mobil tesla, yang mana mobil ini mampu beroperasi sendiri meskipun tidak ada sopir di belakang kemudi. Bahkan lebih hebatnya lagi mobil tesla memiliki kemampuan prediktif yang luar biasa sehingga potensi kemungkinan terjadinya kecelakaan sangat kecil sekali.





*Gambar 1.1 15 Tesla dan Smart car*

- **Alexa**

Kecerdasan buatan yang kedua adalah Alexa. Hampir sama dengan SIRI, Alexa juga merupakan salah satu asisten personal cerdas yang bisa membantu kebutuhan penggunanya. Bedanya, Alexa dibuat oleh Amazon untuk ditanamkan pada perangkat pintar dalam konsep Smart Home (rumah pintar) seperti Smart-lamp, Smart-watch, Smart-speaker, dan Smart-TV. Yang paling menarik dari Alexa adalah ia dapat mengetahui di ruangan dan sebelah mana pengguna memberikan perintah. Jadi anda tidak perlu memberikan perintah di depan speaker untuk mendapatkan respons Alexa. Tinggal sebutkan perintah di manapun anda berada, dan Alexa akan meresponnya.



*Gambar 1.1 16 Alexa smarthome*

Masih banyak contoh-contoh implementasi kecerdasan buatan atau artificial intelligence (AI) didalam kehidupan teknologi masa kini.

- **Dalam bidang industry**



*Gambar 1.1 17 Bidang Industri*

Keberadaannya AI sendiri juga sangat membantu dalam bidang industri seperti halnya pemanfaatan robot yang digunakan

dalam merangkai sebuah mesin sehingga mampu memberikan kemudahan dan keamanan terhadap pengelola.

- **Dalam bidang perfilman**



*Gambar 1.1 18 AI yang diangkat jadi film*

Selain AI juga berperan dalam proses pembuatan film itu sendiri, teknologi AI seperti ini juga banyak diangkat dalam cerita dunia perfilman. Misalnya dalam tokoh animasi seperti robot pengolah sampah Wall-E sampai film aksi Iron Man dan masih banyak lagi.

- **Dalam bidang pertahanan dan keamanan**



*Gambar 1.1 19 UAV menggunakan AI*

AI juga bisa dipakai untuk militer. Mungkin setelah mempertimbangkan WW1 WW2 dulu, nyawa manusia sekarang lebih diperhatikan dalam perang, makanya dibuat aneka kendaraan dan senjata dengan bantuan AI agar nyawa para prajurit lebih aman.

- **Dalam bidang Pendidikan**



*Gambar 1.1 20 Ai dibidang pendidikan*

Kalau ada tugas atau kerjaan hampir kebanyakan kita mencari-cari kata kuncinya ke google, udah nggak ke buku lagi, alasannya tentu lebih praktis, mudah, dan juga rinci. Ini kan berarti peran AI sudah melekat erat pada bidang pendidikan. Istilahnya yaitu eLearning. Sedikit penjelasan eLearning : eLearning adalah suatu metodologi pembelajaran yang memanfaatkan teknologi komputer (khususnya internet) sebagai medianya dan dipercaya dapat meningkatkan keberhasilan pembelajaran dan dari segi sumber daya akan dapat menghemat biaya.

- **Dalam bidang game**



*Gambar 1.1 21 Ai dalam game*

Setiap game pasti memiliki AI standar, misalnya jika anda memerintah ini, maka si AI akan menjalankan tugasnya. Bahkan banyak game-game yang bertemakan sci-fi seperti HALO, Mass Effect, Crysis, dll di dalam game tersebut masing-masing karakter memiliki AI nya sendiri. Jadi ini bisa dikatakan AI di dalam AI.

## **1.2 PENGANTAR MACHINE LEARNING**

### **1.2.1 Pengertian 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 merupakan cabang aplikasi dari Artificial Intelligence (Kecerdasan Buatan) yang fokus pada pengembangan sebuah sistem yang mampu belajar "sendiri" tanpa harus berulang kali di program oleh manusia. Machine learning 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 telah berkembang di mana-mana dalam kehidupan sehari-hari. Aplikasi Machine learning membutuhkan Data sebagai bahan belajar (training) sebelum mengeluarkan output. Aplikasi sejenis ini juga biasanya berada dalam domain spesifik alias tidak bisa diterapkan secara general untuk semua permasalahan.

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. 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. Pembelajaran mesin mempelajari sejumlah data (*learn from data*) sehingga dapat menghasilkan suatu model untuk melakukan proses *input-output* tanpa menggunakan kode program yang dibuat secara eksplisit. Proses belajar tersebut menggunakan algoritma khusus yang disebut *machine learning algorithms*. Terdapat banyak algoritma machine learning dengan efesiensi dan spesifikasi kasus yang berbeda-beda.

Contoh sederhana dari algoritma *machine learning* bisa dilihat pada layanan *streaming* musik *on demand*. Untuk memberikan daftar lagu baru atau musisi yang akan direkomendasikan, maka algoritmanya akan berkaitan dengan preferensi si pendengar dengan jenis musik yang mirip.

*Machine learning* telah memperkuat semua proses otomatisasi dan tersebar di berbagai industri, mulai dari perusahaan keamanan yang memburu malware hingga perusahaan *e-commerce* yang menggunakannya untuk mempelajari produk yang paling disukai

konsumen. Algoritma dari *machine learning* memang kompleks, tapi masih sangat 'mesin', artinya ia hanya mampu melakukan apa yang telah dirancang oleh penciptanya. Tidak lebih, tidak kurang. *Machine Learning* dapat diaplikasikan pada segala bidang. Di era digital saat ini, segala sesuatu telah terkomputerisasi, dan hal tersebut memainkan peranan penting bagi *Machine Learning* untuk dapat terlibat di dalamnya.

Fokus besar penelitian Machine Learning adalah bagaimana mengenali secara otomatis pola kompleks dan membuat keputusan cerdas berdasarkan data. Mesin dalam hal ini memiliki arti “sistem” bukan mesin mekanik secara harfiah.

### 1.2.2 Sejarah Machine Learning

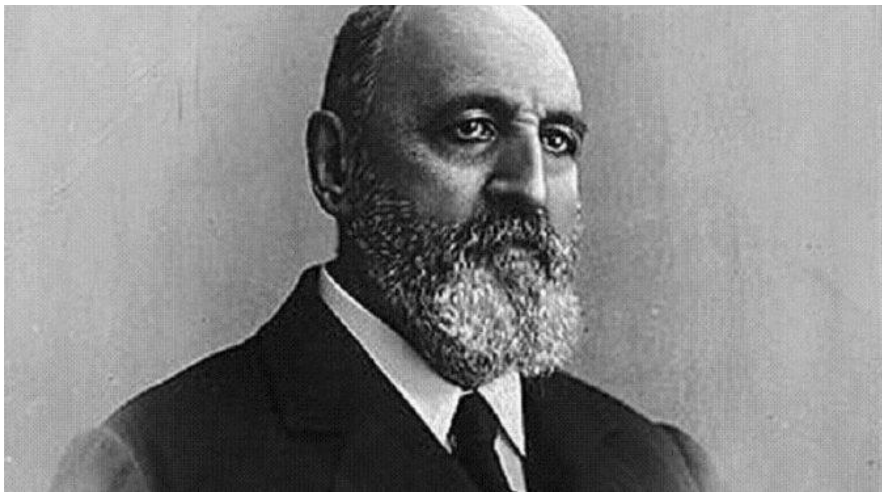
Sejak pertama kali komputer diciptakan, manusia sudah memikirkan bagaimana caranya agar komputer dapat belajar dari pengalaman. Hal tersebut terbukti pada tahun 1952, Arthur Samuel menciptakan sebuah program, Game of Checkers, pada sebuah komputer IBM. Program tersebut dapat mempelajari gerakan untuk memenangkan permainan checkers dan menyimpan gerakan tersebut kedalam memorinya.



Gambar 1.2. 1 Arthut Samuel



Machine Learning bermula di awal abad 20, seorang penemu Spanyol, Torres y Quevedo, membuat sebuah mesin learning setelah ditemukannya komputer digital. Inilah 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.



*Gambar 1.2. 2 Torres y Quevedo*

### **1.2.3 Jenis dan Model Machine Learning**

Berikut tentang jenis-jenis *learning* dari *machine learning*, yaitu Supervised Learning, Unsupervised Learning, Semi-supervised Learning, dan Reinforcement Learning.

- ***Supervised Learning***

*Supervised Learning* adalah tipe *learning* di mana kita mempunyai variable input dan variable output, dan menggunakan satu algoritma atau lebih untuk mempelajari fungsi pemetaan dari input ke output.

*Goal*-nya adalah untuk memperkirakan fungsi pemetaannya, sehingga ketika kita mempunyai input baru, kita dapat memprediksi output untuk input tersebut. Proses dari sebuah algoritma belajar dari *training dataset* dapat diumpamakan sebagai seorang guru yang mengawasi (*supervising*) proses belajar. Kita tahu jawaban yang benar, dan algoritma secara iteratif membuat prediksi pada data latih (*training data*) dan dikoreksi oleh ‘guru’nya.

*Learning* berhenti ketika algoritma mencapai level performansi yang diterima. Permasalahan *Supervised Learning* dapat dikelompokkan menjadi masalah regresi (*regression problem*) dan masalah klasifikasi (*classification problems*).

- ***Unsupervised Learning***

Berbeda dengan *supervised learning*, *unsupervised learning* adalah tipe *learning* di mana kita hanya mempunyai data masukan (*input data*) tetapi tidak ada *output variable* yang berhubungan. *Goal* dari *unsupervised learning* adalah untuk memodelkan struktur dasar atau distribusi dalam data dengan tujuan untuk mempelajari data lebih jauh lagi, dengan kata lain,

adalah menyimpulkan fungsi yang mendeskripsikan atau menjelaskan data.

Berbeda dengan *supervised learning*, di sini tidak ada jawaban yang ‘dibenarkan secara terarah’ dan tidak ada ‘guru’. Algoritma dibiarkan dengan rancangannya sendiri untuk mengetahui dan menyajikan struktur menarik yang ada dalam data.

Lebih jauh lagi, permasalahan *unsupervised learning* dapat dikelompokkan menjadi *clustering problems* dan *association problems*.

- ***Semi-supervised Learning***

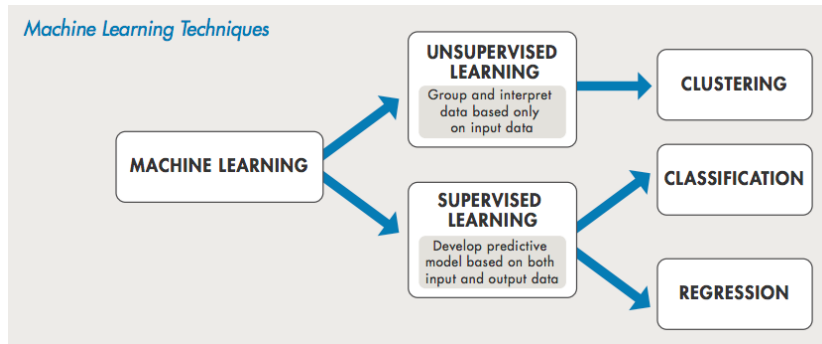
*Semi-supervised Learning* adalah tipe *learning* di mana kita mempunyai data masukan (*input data*) dalam jumlah besar dan hanya beberapa dari data tersebut yang dilabeli. Permasalahan ini berada di antara *supervised learning* dan *unsupervised learning*.

- ***Reinforcement Learning***

*Reinforcement Learning* terjadi ketika kita menyajikan algoritma dengan contoh yang kekurangan label, tetapi kita dapat menyertakan contoh dengan *feedback* positif atau negatif bergantung pada solusi yang ditawarkan oleh algoritma tersebut.

Dengan *reinforcement learning*, mesin dan *software* dapat menentukan tingkah laku ideal terhadap sebuah konteks yang spesifik secara otomatis, dengan tujuan untuk memaksimalkan performanya. Umpan balik (*feedback*) ‘*simple reward*’ dibutuhkan

untuk mesin mempelajari tingkah lakunya; hal ini disebut *reinforcement signal*.



*Gambar 1.2. 3 machine learning technique*

*machine learning* dapat dibedakan menjadi 3 model, yaitu *Geometric Models*, *Probabilistic Models*, dan *Logical Models*. Detailnya adalah sebagai berikut :

- **Geometric Models**

*Geometric Models* atau Model Geometrik adalah model yang dibuat langsung dalam ruang contoh, menggunakan konsep seperti garis (*lines*), bidang (*planes*), atau jarak (*distances*). Model Geometrik juga menggunakan gagasan-gagasan geometrik (seperti translasi dan rotasi bidang) dan sederhana, *powerful* dan memungkinkan banyak variasi dengan sedikit *effort* (usaha). Contoh dari metode model geometrik adalah Manhattan Distance.

- **Probabilistic Models**

*Probabilistic Models* atau Model Probabilistik adalah model yang akan mempelajari distribusi probabilistik dari data yang diberikan

dan digunakan. Pendekatan statistik mengasumsikan bahwa ada beberapa proses acak pokok yang menghasilkan nilai dari variable-variable yang diukur, berdasarkan distribusi probabilistik yang tidak diketahui tetapi didefinisikan dengan baik. Contoh dari model probabilistik adalah metode Naive-Bayes.

- **Logical Models**

*Logical Models* atau Model Logikal adalah model yang dapat ditranslasikan dengan mudah ke dalam aturan-aturan tertentu karena dapat dimengerti oleh manusia. Model ini membunya aturan-aturan (yang disebut *feature tree*) yang dapat diorganisir / diatur dengan mudah dalam struktur pohon (*structured tree*). Contoh dari metode atau algoritma yang menggunakan model logikal adalah *decision tree*.

Untuk menggunakan algoritma *machine learning* digunakan dalam membantu mesin mempelajari data untuk menyelesaikan masalah, perlu diketahui secara spesifik apa yang akan digunakan algoritma sebagai data. Secara umum metode dan algoritma *machine learning* ditulis untuk fitur atau variabel numerik. Fitur merupakan hasil observasi data *train* yang ditranslasikan dalam bentuk kuantitatif atau kualitatif. Banyak fitur digabungkan menjadi vektor fitur yang digunakan sebagai input untuk algoritma *machine learning* dalam menentukan kelas yang sesuai bagi data baru.

#### **1.2.4 Dampak Machine Learning**

*Dalam penerapan teknologi machine learning ini, kebanyakan orang mungkin telah merasakan dampaknya sekarang. Dalam pengembangan teknologi machine learning ada dampak yang saling bertolak belakang yaitu dampak negatif dan dampak positif. Ini yang akan memberikan masukan yang berdampak buruk dan baiknya, tergantung terhadap orang yang menilainya. Akan tetapi semua ini tidak selalu berjalan dengan mulus.*

Dampak positif dari machine learning adalah mendapat kesempatan bagi para wirausahawan dan praktisi teknologi untuk terus berkreasi dalam mengembangkan machine learning. Tentunya untuk membantu aktivitas manusia sebagai sesuatu yang menguntungkan. Itulah salah satu dampak positif dari machine learning. Contohnya adalah untuk pengecekan ejaan untuk tiap bahasa yang ada dalam microsoft Word.

Pengecekan manual akan menghabiskan waktu untuk beberapa hari, juga memerlukan banyak tenaga untuk mendapatkan penulis yang sempurna. Namun, dengan bantuan fitur pengecekan tersebut, maka secara real-time kesalahan yang terjadi saat pengetikan kita bisa langsung melihatnya.

Dampak negatifnya kita harus waspada. Yang takut di khawatirkan yaitu adanya pengurangan tenaga kerja. Kenapa? Karena pekerjaan yang seharusnya di kerjakan oleh banyak orang, sekarang telah digantikan oleh alat teknologi yang disebut sebagai machine learning. Hal tersebut merupakan suatu permasalahan yang akan kita hadapi.

Ditambah dengan ketergantungan terhadap teknologi yang semakin banyak dan berkembang di kehidupan kita. Kadang manusia lebih nyaman dengan perkembangan teknologi sekarang ini seperti gadget.

### **1.2.5 Konsep Utama Machine Learning**

Konsep ini mampu meningkatkan kecerdasan yang meliputi kemampuan dari suatu individu untuk belajar tanpa terkecuali pada sebuah mesin. Produktivitas manusia akan meningkat jika suatu mesin mampu untuk belajar. Dan mesin tersebut akan berbeda dengan mesin lain nya, karena ia mempunyai kemampuan yang tidak di miliki oleh mesin lain nya.

- **Manfaat pembelajaran mesin dalam memprediksi**

Jika kamu hanya melihat wajah teman mu dari dalam gambar, berarti kamu belum menggunakan model pembelajaran mesin. Pembelajaran mesin ialah suatu faktor lain yang telah di latih dan meramalkan hal-hal berdasarkan pola. Sesuatu akan lebih gampang untuk di kuasai jika dilakukan pengenalan dengan cara yang sederhana (simple) tanpa membuang – buang banyak waktu.

- **Pembelajaran mesin membutuhkan pelatihan**

Pikirkanlah bagaimana anak manusia bisa belajar dan kamu juga harus memberi tahu bagaimana model *pembelajaran dari mesin learning* yang akan di prediksi. Semua ini mungkin adalah penyederhanaan yang berlebihan, karena saya harus meninggalkan hal – hal yang mana kamu juga harus berkata bahwa itu bukan lah pisang

dan kamu harus menunjukkan berbagai macam pisang, dari segi warna yang berbeda, sudut yang berbeda, gambar dari perspektif, dan lain sebagainya.

- **Pembelajaran mesin berbeda dengan AI**

Sebagian orang banyak mengatakan bahwa AI (Artificial Intelligence) dan Machine Learning itu sama dan sederhana. Tapi, kenyataan nya AI (Artificial Intelligence) dan Machine Learning itu memiliki perbedaan menurut para ahli. Perbedaan nya yaitu :

Machine Learning (ML) merupakan Suatu metode untuk mencapai AI. Yang berarti dapat menerapkan perangkat pelatihan dengan berbagai cara yang berbeda di platform machine learning untuk memprediksi waktu. Dan akan membuat prediksi mengenai sesuatu yang berdasarkan pada pelatihan dari kumpulan – kumpulan data parsing.

Sedangkan Artificial Intelligence (AI) Yang berarti untuk melakukan pekerjaan atau pun tugas – tugas tertentu komputer lebih baik dari pada manusia. Contoh nya saja pada robot yang mampu membuat suatu keputusan berdasarkan banyak nya masukan – masukan, bukan seperti C3PO atau terminator. Sebenarnya tidak terlalu berguna istilah yang sangat luas itu.

- **Memberikan struktur yang jelas terhadap AI**

untuk melakukan apa yang akan dikerjakan *mesin pembelajaran*, maka sebagian besar model Machine learning



bergantung pada manusia. Ini lah yang menyebabkan kamu bergantung terhadap teknologi tersebut, karena sesuatu yang akan kamu kerjakan. Dan itu biasanya masih salah, saat kamu memberikan instruksi yang jelas. Jika kamu begitu eksplisit dengan sistem, sehingga kesempatan itu tiba-tiba menjadi lebih gampang.

Di era digital sekarang ini, machine learning bisa di aplikasi kan ke semua bidang, karena segala sesuatu telah terkomputerisasi dan machine learning terlibat di dalam nya. Itulah ulasan kita tentang definisi dan konsep utama Machine Learning semoga menambah pengetahuan dan bermanfaat.

### **1.2.6 Contoh penerapan Machine Learning**

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 beberapa contoh penerapan teknologi menggunakan machine learning.

- **Hasil Pencarian Search Engine**

Google dan mesin pencari lainnya seperti Bing dan Yandex sudah dari dulu menerapkan machine learning untuk melakukan perangkian 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. Begitu pula, saat kita melihat halaman pencarian berikutnya misalnya halaman 2, 3, dan seterusnya. Google akan mendeteksi adanya ketidaksesuaian kata kunci dengan hasil pencarian yang dihasilkan. Begitulah data tersebut terkumpul dan dianalisis menggunakan machine learning oleh Google Search Engine, untuk menghasilkan hasil pencarian yang dinamis dan berkualitas.

- **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 akunnya.

- **Pengalaman Pengguna Sosial Media**

Tidak bisa dipungkiri, 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. Seperti Orang yang Mungkin Anda

kenal yang sering muncul di halaman facebook, twitter, Instagram dan lain-lain, Pengenalan Wajah yang ada pada kamera, Pos pada Beranda dan masih banyak lagi.

- **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.

- **Videos Surveillance**

*Videos surveillance* atau pengawasan video merupakan teknologi baru yang merupakan penerapan dari machine learning yang disematkan pada CCTV untuk mendeteksi suatu tindak kejahatan atau kecelakaan. Di negara-negara maju, CCTV sudah digunakan untuk melakukan pencarian penjahat yang masih buron.

## **1.3 PENGANTAR NEUREAL NETWORK**

### **1.3.1 Pengertian Neureal Network**

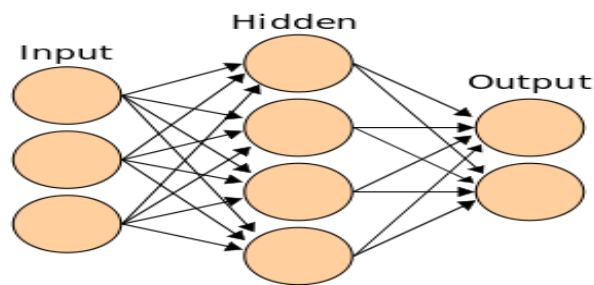
Jaringan syaraf tiruan (JST) atau yang biasa disebut Artificial Neural Network (ANN) atau Neural Network (NN) saja, merupakan sistem pemroses informasi yang memiliki karakteristik mirip dengan jaringan syaraf pada makhluk hidup. Neural network berupa suatu model sederhana dari suatu syaraf nyata dalam 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 disebarkan pada beberapa neuron yang lain.

Artificial Neural Network Artificial (ANN) atau Jaringan Syaraf Tiruan merupakan sebuah teknik atau pendekatan pengolahan informasi yang terinspirasi oleh cara kerja sistem saraf biologis, khususnya pada sel otak manusia dalam memproses informasi. Elemen kunci dari teknik ini adalah struktur sistem pengolahan informasi yang bersifat unik dan beragam untuk tiap aplikasi. Neural Network terdiri dari sejumlah besar

elemen pemrosesan informasi (neuron) yang saling terhubung dan bekerja bersama-sama untuk menyelesaikan sebuah masalah tertentu, yang pada umumnya dalam masalah klasifikasi ataupun prediksi.

Jaringan Syaraf Tiruan atau Artificial Neural Network (NN) adalah teknik dalam ML yang menirukan syaraf manusia yang merupakan bagian fundamental dari otak. NN terdiri atas lapis masukan (input layer) dan lapis keluaran (output layer). Setiap lapis terdiri atas satu atau beberapa unit neuron yang mempunyai sebuah fungsi aktivasi yang menentukan keluaran dari unit tersebut. Kita bisa menambahkan lapis tersembunyi (hidden layer) untuk menambah kemampuan dari NN tersebut.



*Gambar 1.3. 1 Neureal network*

NN bisa dilatih dengan menggunakan data training. Semakin banyak data training maka akan semakin bagus unjuk kerja dari NN tersebut. Namun, kemampuan NN juga terbatas pada jumlah lapisan, semakin banyak jumlah lapisan semakin tinggi kapasitas NN tersebut. Semakin banyak lapisan juga membawa kekurangan yaitu semakin banyaknya jumlah iterasi atau training yang dibutuhkan. Untuk mengatasi hal ini, dikembangkanlah teknik Deep Learning. Beberapa aplikasi NN antara lain untuk Principal Component Analysis, regresi, klasifikasi citra, dan lain-lain.

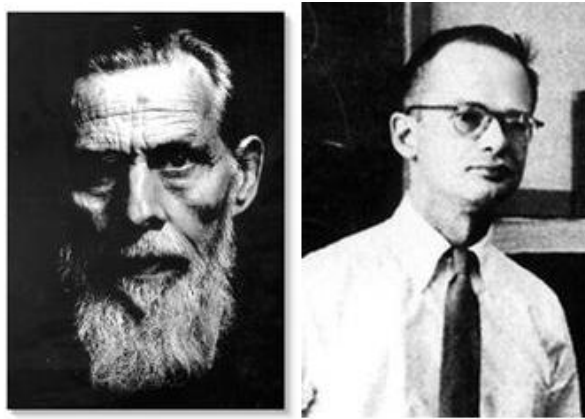
Neural Network juga merupakan kategori ilmu Soft Computing. Neural Network sebenarnya mengadopsi dari kemampuan otak manusia yang mampu memberikan stimulasi/rangsangan, melakukan proses, dan memberikan output. Output diperoleh dari variasi stimulasi dan proses yang terjadi di dalam otak manusia. Kemampuan 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.

### **1.3.2 Sejarah Neureal Network**

Saat ini bidang kecerdasan buatan dalam usahanya menirukan intelegensi manusia, belum mengadakan pendekatan dalam bentuk fisiknya melainkan dari sisi yang lain. Pertama-tama diadakan studi mengenai teori dasar mekanisme proses terjadinya intelegensi. Bidang ini disebut ‘Cognitive Science’. Dari teori dasar ini dibuatlah suatu model untuk disimulasikan pada komputer, dan dalam perkembangannya yang lebih lanjut dikenal berbagai sistem kecerdasan buatan yang salah satunya adalah jaringan saraf tiruan. Dibandingkan dengan bidang ilmu yang lain, jaringan saraf tiruan relatif masih baru.

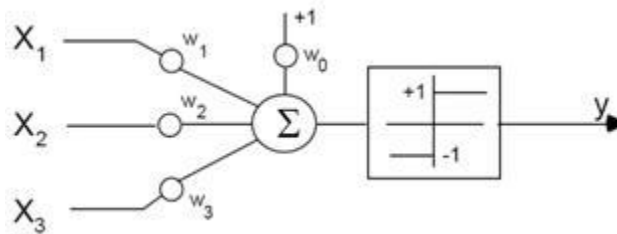
Perkembangan ilmu Neural Network sudah ada sejak tahun 1943 ketika Warren McCulloch dan Walter Pitts memperkenalkan perhitungan model neural network yang pertama kalinya. Mereka melakukan kombinasi beberapa *processing unit* sederhana bersama-

sama yang mampu memberikan peningkatan secara keseluruhan pada kekuatan komputasi.



*Gambar 1.3. 2 McCulloch dan Walter*

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 1.3. 3 Perceptron*

Keberhasilan perceptron dalam pengklasifikasian pola tertentu ini tidak sepenuhnya sempurna, masih ditemukan juga beberapa

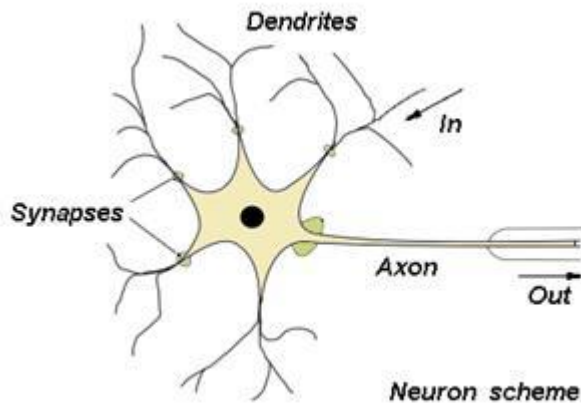
keterbatasan didalamnya. Perceptron tidak mampu untuk menyelesaikan permasalahan XOR (*exclusive-OR*). Penilaian terhadap keterbatasan neural network ini membuat penelitian di bidang ini sempat mati selama kurang lebih 15 tahun. Namun demikian, perceptron berhasil menjadi sebuah dasar untuk penelitian-penelitian 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.

### **1.3.3 Konsep dan Struktur Neureal Network**

- **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  $10^4$  sinapsis. Masing-masing sel itu saling berinteraksi satu sama lain yang menghasilkan kemampuan tertentu pada kerja otak manusia.





Gambar 1.3. 4 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.

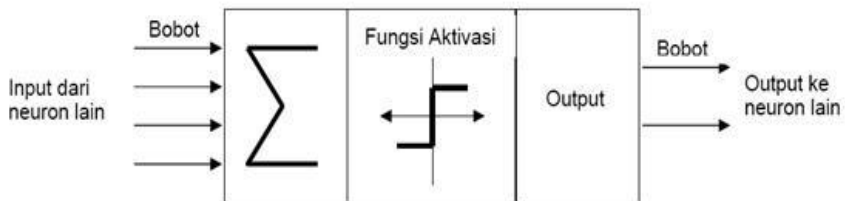
Proses yang terjadi pada otak manusia adalah:

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).

- **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. 5 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 ( $\Sigma$ ). Hasil penjumlahan akan diproses oleh fungsi aktivasi setiap neuron, disini akan dibandingkan hasil penjumlahan dengan *threshold* (nilai ambang) tertentu. Jika nilai melebihi *threshold*, maka aktivasi neuron akan dibatalkan, sebaliknya, jika masih dibawah nilai *threshold*, neuron akan diaktifkan. Setelah aktif, neuron akan mengirimkan nilai output melalui bobot-bobot outputnya ke semua neuron yang berhubungan dengannya. Proses ini akan terus berulang pada input-input selanjutnya.

ANN terdiri dari banyak neuron di dalamnya. Neuron-neuron ini akan dikelompokkan ke dalam beberapa layer. Neuron yang terdapat pada tiap layer dihubungkan dengan neuron pada layer lainnya. Hal ini tentunya tidak berlaku pada layer input dan output, tapi hanya layer yang berada di antaranya. Informasi yang diterima di layer input dilanjutkan ke layer-layer dalam ANN secara satu persatu hingga

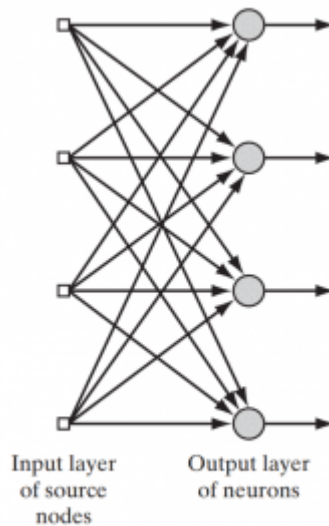
mencapai layer terakhir/layer output. Layer yang terletak di antara input dan output disebut sebagai *hidden layer*. Namun, tidak semua ANN memiliki hidden layer, ada juga yang hanya terdapat layer input dan output saja.

#### **1.3.4 Model Neureal Network**

Berikut macam – macam ANN dan metode – metode dari ANN dan Permodelan jaringan pada ANN ada beberapa macam, yaitu:

- ***Single layer***

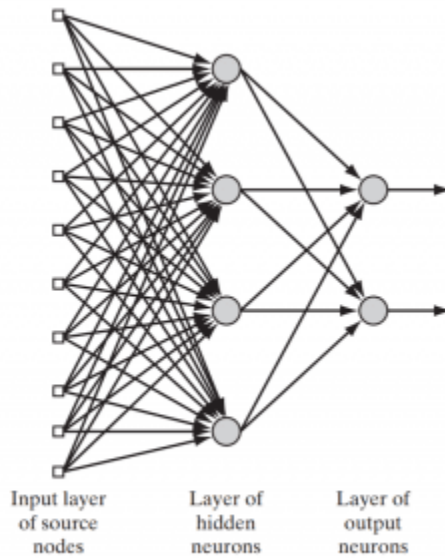
Dalam ANN, neuron disusun dalam bentuk lapisan (layer). Pembentukan ANN yang paling sederhana yaitu single layer. Cara kerja dari *single layer*, input layer yang berasal dari sumber node di proyeksikan langsung ke output layer dari neuron ( node komputasi), tetapi tidak berlaku sebaliknya. Permodelan ini merupakan jenis jaringan *feedforward* yang dapat dilihat pada gambar dibawah. Pada gambar tersebut input dan output memiliki 4 node, namun yang dimaksud dengan single layer yaitu output dari jaringan, sedangkan inputnya tidak memiliki pengaruh karena pada saat melakukan input tidak terjadi proses komputasi



Gambar 1.3. 6 Single layer

- **Multi layer**

Pada *single layer* apabila terdapat tambahan satu atau dua hidden layer maka jaringan akan terganggu karena *input* dan *output* dari jaringan tidak dapat melihat *hidden layer* yang di masukkan. Sehingga memerlukan jaringan yang bisa menampung nya yaitu bernama *multi layer*. Cara kerja *multi layer* adalah *input layer* menyuplai *input vektor* pada jaringan, kemudian *input* yang dimasukkan melakukan komputasi pada layer yang kedua, lalu output dari layer yang kedua digunakan sebagai input dari layer yang ketiga dan seterusnya. Ilustrasi jaringan multi layer dapat di lihat pada gambar dibawah.



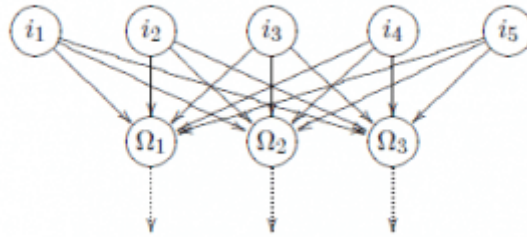
Gambar 1.3. 7 Multi layer

Berikut macam – macam metode – metode dari ANN

- **Single Layer Perceptron (SLP)**

*perceptron* merupakan salah satu jaringan *feedforward* yang terdiri dari sebuah retina yang digunakan untuk akuisisi data yang mempunyai *fixed-weighted connection* dengan *neuron layer* yang pertama. *Fixed weight layer* diikuti dengan minimal satu *weight layer*. SLP merupakan sebuah perceptron yang memiliki satu variable *weight* dan satu variable layer dari output neuron  $\Omega$ . Teknik dari SLP dapat ditunjukkan pada gambar dibawah, sebuah SLP dengan dua *input* dan satu *output neuron*.

Sebuah *perceptron* dengan beberapa *output neuron* dapat juga dianggap seperti beberapa *perceptron* berbeda dengan input yang sama. Fungsi Boolean AND dan OR adalah salah satu contoh yang dengan mudah di susun.



Gambar 1.3. 8 teknik SLP

*Learning* algoritma pada perceptron dengan fungsi aktivasi binary neuron *Learning* algoritma pada perceptron mengurangi weight ke output neuron yang mengembalikan 1 daripada 0 dan di kasus yang berbeda weight nya meningkat Contohnya kita memiliki single layer perceptron dengan set weight yang random untuk training sample. Set dari training sample yang di panggil P, yang berisi pasangan dari training sampe p dan input t.

$$\forall p : y \approx t \quad \text{or} \quad \forall p : E_p \approx 0.$$

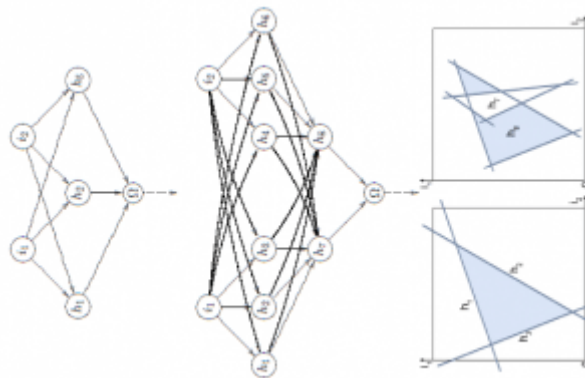
Dalam persamaan tersebut kita harus mengetahui tentang total error err sebagai fungsi dari weight. Total error naik atau berkurang tergantung pada bagaimana kita mengubah weight. Persamaan fungsi error pada SLP sebagai berikut :

$$Err : W \rightarrow \mathbb{R}$$

- **Multi Layer Perceptron (MLP)**

*Multi layer perceptron* adalah sebuah perceptron dengan dua atau lebih *trainable weight layer*. Pada SLP dapat membagi *input space* dengan sebuah *hyperlane* sedangkan MLP dapat

mengklasifikasi *convex polygon* dari proses *hyperlane* dengan mengenali *pattern* yang terletak di atas *hyperlane*. MLP merupakan representasi dari fungsi pendekatan universal. Sebuah *n-layer perceptron* adalah *n-variable weight layer* dan *n+1 neuron layer* dengan *neuron layer 1* sebagai *input layer*. Ilustrasi dari MLP dapat dilihat pada gambar dibawah.

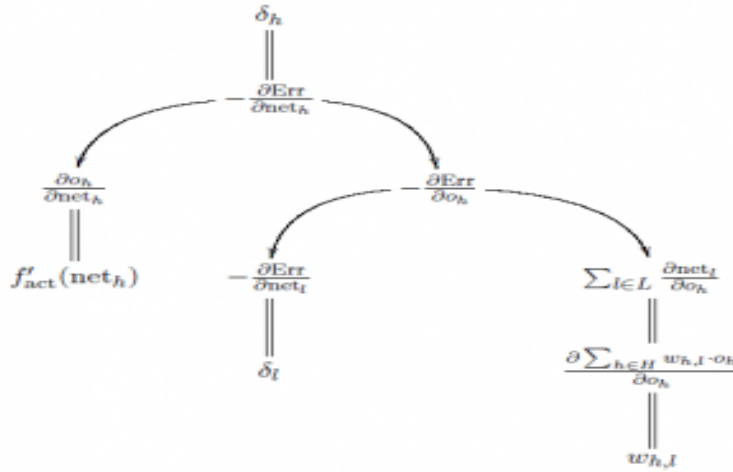


Gambar 1.3. 9 ilustrasi MLP

Penanganan fungsi error pada MLP menggunakan *backpropagation error*, yang dapat diaplikasikan pada *multi perceptron* dengan fungsi aktivasi semi linear. Walaupun fungsi *binary threshold* dan fungsi yang lain tidak lagi mendukung tetapi kita bisa menggunakan fungsi Fermi

atau *hyperbolic tangent*. *Backpropagation* merupakan prosedur *gradient* dengan fungsi error  $Err(W)$  menerima semua *n-weight* sebagai argument dan menugaskannya sebagai *output error n dimensional*. Ilustrasi fungsi error yang digunakan ditunjukkan oleh gambar berikut:





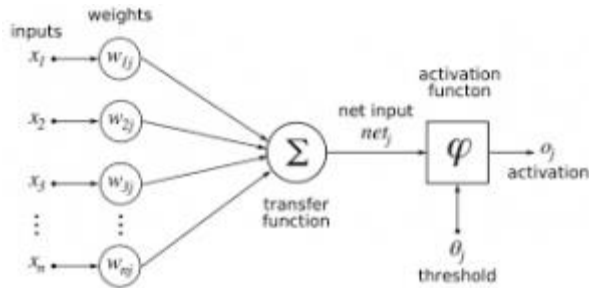
Gambar 1.3. 10 fungsi error pada MLP

Setelah dilakukan pembuktian persamaan yang fungsi yang dihasilkan :

$$\Delta w_{k,h} = \eta o_k \delta_h = \eta o_k \cdot (t_h - o_h)$$

### 1.3.5 Cara Kerja Neureal Network

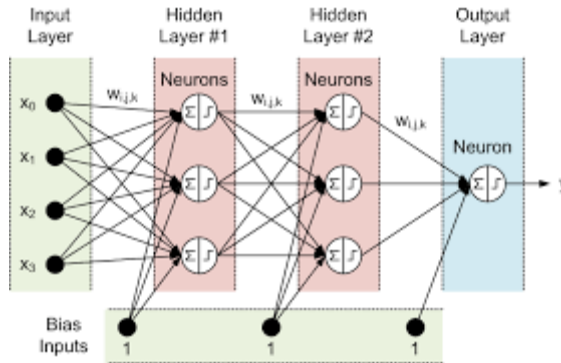
Cara kerja Neural Network dapat dianalogikan sebagaimana halnya manusia belajar dengan menggunakan contoh atau yang disebut sebagai supervised learning. Sebuah Neural Network dikonfigurasi untuk aplikasi tertentu, seperti pengenalan pola atau klasifikasi data, dan kemudian disempurnakan melalui proses pembelajaran. Proses belajar yang terjadi dalam sistem biologis melibatkan penyesuaian koneksi sinaptik yang ada antara neuron, dalam halnya pada Neural Network penyesuaian koneksi sinaptik antar neuron dilakukan dengan menyesuaikan nilai bobot yang ada pada tiap konektivitas baik dari input, neuron maupun output.



*Gambar 1.3. 11 Cara Kerja NN*

Neural Network memproses informasi berdasarkan cara kerja otak manusia. Dalam hal ini Neural Network terdiri dari sejumlah besar elemen pemrosesan yang saling terhubung dan bekerja secara paralel untuk memecahkan suatu masalah tertentu. Di sisi lain, komputer konvensional menggunakan pendekatan kognitif untuk memecahkan masalah; dimana cara pemecahan masalah haruslah sudah diketahui sebelumnya untuk kemudian dibuat menjadi beberapa instruksi kecil yang terstruktur. Instruksi ini kemudian dikonversi menjadi program komputer dan kemudian ke dalam kode mesin yang dapat dijalankan oleh komputer.

Neural Network, dengan kemampuannya dapat digunakan untuk memperoleh pengetahuan dari data yang rumit atau tidak tepat, serta juga dapat digunakan untuk mengekstrak pola dan mendeteksi tren yang terlalu kompleks untuk diperhatikan baik oleh manusia atau teknik komputer lainnya. Sebuah Neural Network yang telah terlatih dapat dianggap sebagai “ahli” dalam kategori pemrosesan informasi yang telah diberikan untuk dianalisa. Ahli ini kemudian dapat digunakan untuk menyediakan proyeksi terkait kemungkinan kondisi di masa mendatang serta menjawab pertanyaan “bagaimana jika?”



*Gambar 1.3. 12 Proses NN*

Keuntungan lainnya dari penggunaan Neural Network termasuk:

- Pembelajaran adaptif: Kemampuan untuk belajar dalam melakukan tugas-tugas berdasarkan data yang diberikan
- Self-Organization: Sebuah Neural Network dapat membangun representasi dari informasi yang diterimanya selama proses pembelajaran secara mandiri
- Operasi Real-Time: Penghitungan Neural Network dapat dilakukan secara paralel, sehingga proses komputasi menjadi lebih cepat.

Neural Network dan algoritma komputer konvensional tidaklah saling bersaing tetapi saling melengkapi. Beberapa tugas atau masalah lebih cocok diselesaikan dengan pendekatan algoritmik seperti halnya operasi aritmatika, di sisi lain ada tugas-tugas yang lebih cocok untuk jaringan saraf, misalnya prediksi pergerakan data time-series. Bahkan, sejumlah besar tugas lainnya memerlukan sistem yang menggunakan kombinasi dari kedua pendekatan tersebut, dimana biasanya komputer konvensional digunakan untuk mengawasi Neural Network agar dapat memberikan kinerja maksimum.

## 1.4 PENGANTAR DEEP LEARNING

### 1.4.1 Pengertian Deep Learning

Dalam istilah praktis, *deep learning* merupakan bagian dari *machine learning*. Sebuah model *machine learning* perlu 'diberitahu' untuk bagaimana ia menciptakan prediksi akurat, dengan terus diberikan data. Sementara model *deep learning* dapat mempelajari metode komputasinya sendiri, dengan 'otaknya' sendiri, apabila diibaratkan. Sebuah model *deep learning* dirancang untuk terus menganalisis data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan. Untuk dapat mencapai kemampuan itu, *deep learning* menggunakan struktur algoritma berlapis yang disebut *artificial neural network* (ANN).

Desain ANN terinspirasi dari jaringan neural biologis dari otak manusia. Hal ini membuat mesin kecerdasannya menjadi jauh lebih tangguh dibandingkan model *machine learning* standar.

Data-data yang digunakan dalam sebuah *deep learning* sangatlah penting, karena semakin banyak datanya, maka semakin banyak yang bisa dipahami model *deep learning* tersebut.

Contoh dari penggunaan model *deep learning* bisa dilihat dari AlphaGo-nya Google. Google menciptakan program komputer yang belajar bermain sebuah game sejenis catur dari China bernama Go. Tentunya, game ini membutuhkan pemikiran dan intuisi yang tajam untuk menang.

Dengan bermain melawan pemain Go profesional, *deep learning* AlphaGo mempelajari bagaimana ia bermain di tingkat yang belum terjamah sebelumnya dalam kecerdasan buatan. Hebatnya, apa yang dilakukannya tanpa instruksi apapun ketika melancarkan gerakan-gerakan spesifik.



*Gambar 1.4. 1 AlphaGo*

Melansir dari *MIT Technology Review* (6/3/2019), *deep learning* artian harfiahnya pembelajaran mendalam, namu juga dikenal sebagai pembelajaran terstruktur dalam atau pembelajaran hierarkis. Istilah tersebut memiliki arti bagian dari keluarga yang lebih luas dari metode *machine learning* berdasarkan pada representasi data pembelajaran, yang bertentangan dengan algoritma tugas-spesifik. Singkatnya, *deep learning* merupakan sebuah model yang dapat mempelajari metode komputasinya sendiri dengan 'otaknya' sendiri. Dengan adanya Deep Learning, banyak bidang mampu berkembang lebih pesat. Mulai dari kesehatan, bercocok tanam yang lebih efektif, hingga mobil tanpa supir (self-driving car).

Deep Learning adalah bagian dari kepintaran buatan (*Artificial Intelligence*—AI). AI terbukti mampu membangun dengan pesat berbagai bidang jenis industri. AI seakan menjadi “listrik baru” yang mampu

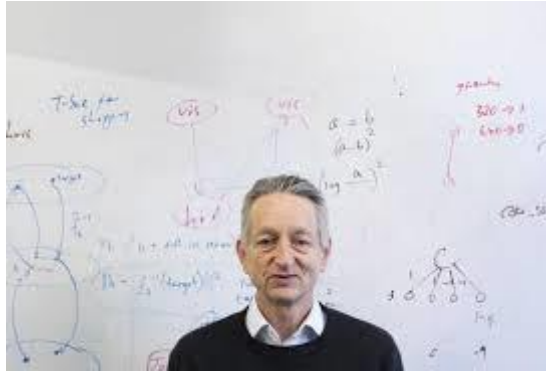
memberikan penemuan-penemuan baru dalam industri. Begitu juga dengan Deep Learning. Deep learning juga merupakan bagian dari *machine learning* yang merupakan pengembangan dari *neural network multiple layer* untuk memberikan ketepatan tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa dan lain-lain. *Deep Learning* berbeda dari teknik *machine learning* yang tradisional, karena *deep learning* secara otomatis melakukan representasi dari data seperti gambar, video atau text tanpa memperkenalkan aturan kode atau pengetahuan domain manusia.

#### 1.4.2 Sejarah Deep Learning

*Deep Learning* dikembangkan pada tahun 1950 namun baru tahun 1990 dapat di aplikasikan dengan sukses. *Learning* algoritma yang digunakan sekarang pada task yang kompleks hampir sama seperti *learning* algoritma yang digunakan untuk menyelesaikan masalah permainan pada tahun 1980, meskipun model algoritma yang digunakan berubah menjadi *training* yang sederhana dari arsitektur *deep learning*. Hal yang penting pada pengembangan model yang sekarang adalah kita dapat mendukung dengan *resource* yang dibutuhkan agar menjadi sukses. Pengembangan data set yang semakin meningkat menyebabkan data set tersentralisasi yang memudahkan dalam pengelolaannya.

Pada tahun 2006, Geoffrey Hinton memperkenalkan salah satu varian jaringan saraf tiruan yang disebut *deep belief nets*, ide untuk *train* model jaringan saraf tiruan ini adalah dengan *train* dua *layer* kemudian tambahkan satu *layer* diatasnya, kemudian *train* hanya *layer* teratas dan begitu seterusnya. Dengan strategi ini kita dapat *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*.



Gambar 1.4. 2 Geoffrey Hinton

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 dan rekan rekannya 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 pengenalan wajah, *self-driving car*, pengenalan suara, dan lain lain.



*Gambar 1.4. 3 Andrew ng*

Sebuah model *deep learning* dirancang untuk terus menganalisis data dengan struktur logika yang mirip dengan bagaimana manusia mengambil keputusan. Untuk dapat mencapai kemampuan itu, *deep learning* menggunakan struktur algoritma berlapis yang disebut *artificial neural network* (ANN).

Istilah ini pertama kali diperkenalkan ke komunitas pembelajaran mesin oleh Rina Dechter pada tahun 1986. Perkembangan terus terjadi. Pada tahun 2009, Nvidia, perusahaan teknologi asal Amerika terlibat dalam "big bang" dari *deep learning*, Nvidia graphics processing units (GPUs) terus melatih *deep learning* dengan neural training. Masih di tahun yang sama, Google Brain juga menggunakan Nvidia GPU untuk membuat deep neuron network (DNN). Seiring berjalannya waktu, *deep learning* telah dikembangkan melalui beberapa pelatihan yang terus dilakukan oleh masing-masing ahli. Setelah melewati proses yang panjang, barulah optimalisasi perangkat keras dan algoritma khusus dapat digunakan untuk pemrosesan yang efisien.



*Deep learning* bisa menjadi kepingan *puzzle* utama yang dapat membawa manusia pada penciptaan AI yang lebih cerdas dan manusiawi. Ia dapat meningkatkan semua bagian AI, mulai dari pemrosesan bahasa alami hingga *machine vision*.

### **1.4.3 Contoh Pemanfaatan Deep Learning**

*Deep learning* adalah metode pembelajaran yang dilakukan oleh mesin dengan cara meniru bagaimana sistem dasar otak manusia bekerja. Sistem dasar otak manusia bekerja ini disebut *neural networks*. Itulah kenapa *deep learning* disebut menggunakan *artificial neural networks* yang dengan kata lain menggunakan ‘*neural networks* buatan’.

Penerapan *deep learning* telah banyak digunakan untuk melakukan sebuah penelitian dan berikut beberapa penerapan *deep learning* yang di gunakan.

- **Pengenalan wajah**

Teknologi pengenalan wajah sepertinya telah menyebar luas dalam beberapa tahun terakhir, hal ini bisa kita lihat pada kamera smartphone yang juga memiliki fungsi pengenalan wajah.

Sehubungan dengan fitur pengenalan wajah, masyarakat Tiongkok bahkan telah lama menggunakan teknologi *cardless ID* dalam kegiatan sehari-hari mereka.

Fungsi pengenalan gambar yang diluncurkan oleh Anto Financial, afiliasi dari Alibaba ini, memungkinkan kita untuk melakukan otentikasi yang diperlukan untuk pembayaran digital dimana user hanya perlu mengambil foto selfienya.

Di negara ini, pengguna (orang-orang) dapat dengan mudah memasuki ruang universitas, naik pesawat dan memasuki kantor karyawan hanya dengan menggunakan otentikasi wajah tanpa menggunakan kartu ID.

Algoritma otentikasi wajah yang akurat didasarkan pada deep learning membutuhkan kumpulan data yang besar untuk melatih sistem, tetapi pada tahun 2017 China berhasil mengidentifikasi individu hanya dalam beberapa detik dari 1,3 miliar orang penduduknya. Mereka telah berhasil membangun database otentikasi wajah besar yang dapat mengidentifikasi secara akurat setiap orang dari sejumlah besar data gambar wajah.

- **Aplikasi di Bidang Medis**

Sebuah perusahaan bernama Enlitic sedang mengembangkan teknologi deep learning yang mengkhususkan diri dalam bidang kesehatan.

Enlitic menggunakan teknik deep learning untuk mendeteksi tumor ganas dan penyakit kronis lainnya berdasarkan hasil diagnosis gambar (X-ray, CT scan, MRI dll.).

Keakuratan tingkat deteksi kanker paru-paru dan menemukan penyakit yang tidak dapat ditemukan bahkan oleh dokter menjadi tujuan dari dikembangkannya teknologi ini, karena ahli radiologi sendiri melebihi akurasi mendeteksi kanker paru-paru oleh hanya satu orang dengan lebih dari 50%.

Dengan teknologi pengenalan gambar deep learning, diharapkan tidak akan sulit untuk mengidentifikasi penyebab penyakit, ini menjadi bidang yang menarik perhatian karena dapat melakukan identifikasi dengan hasil yang lebih akurat daripada dokter.

- **Pencari produk**

Saat ini membeli produk melalui internet bukan lagi suatu hal yang asing bagi kita, dan konstruksi dari situs pembelian tersebut juga beragam.

Bagi penyedia layanan, sangat penting untuk secara otomatis menampilkan contoh gambar yang cocok dengan preferensi pengguna di dalam situs sehingga memudahkan mereka untuk memilih barang, yang secara langsung terkait dengan penjualan.

Teknologi pengenalan gambar sangat efektif untuk sistem ini karena memudahkan pengguna untuk mengidentifikasi preferensi mereka.

Deep learning sangat berguna terutama untuk mengidentifikasi preferensi barang-barang mode seperti pakaian dan sepatu.

Hal ini karena preferensi warna dan bentuk adalah bagian yang sulit untuk diucapkan.

Bahkan jika Anda mengatakan bahwa Anda menyukai warna merah, deep learning akan membantu Anda untuk mengidentifikasikan, apakah itu merah terang, merah sedikit gelap, atau merah mendekati pink.

- **Kombinasi dengan pemrosesan Bahasa**

Contoh keempat adalah kombinasi dari dua jenis deep learning, yaitu pemrosesan gambar dan pemrosesan bahasa.

Neural network yang sering digunakan dalam pengenalan gambar (CNN) dan neural network yang sering digunakan dalam pemrosesan bahasa alami (RNN), gabungan keduanya adalah sistem yang secara otomatis menghasilkan teks gambar deskriptif.

Karena memungkinkan untuk mengambil gambar dari kata tertentu, penyediaan oleh layanan sistem media sosial dianggap paling efektif.

Memang sangat mudah untuk mengunggah foto dan gambar di internet, tetapi agar orang lain menemukannya, kita perlu memberikan semacam informasi tag.

Dengan sistem ini, Anda tidak perlu menandainya lagi nanti, karena deep learning secara otomatis akan menandai gambar tersebut.

Hal ini tidak hanya bermanfaat untuk pengguna tetapi juga untuk penyedia layanan tentu saja menguntungkan, karena Anda juga dapat memeriksa tren favorit pengguna melalui foto dan gambar yang dipilih oleh setiap pengguna, dan berdasarkan informasi itu, dimungkinkan juga untuk memberikan informasi gambar kepada pengguna.

- **Kecerdasan Buatan dan Pengembangnya**

Terakhir kita akan membahas sebuah kasus di mana pengenalan gambar dengan deep learning yang mengarah pada pengembangan kecerdasan buatan yang kelima.

Di sini Anda akan diperkenalkan berbagai hasil di bidang pengenalan gambar dan di mana saja layanan itu sudah digunakan, dalam hal ini teknologi pengenalan gambar telah membuat kemajuan lagi.

Di sini, computer sudah bisa mengidentifikasi “kucing” dengan sendirinya dengan pengenalan gambar dari deep learning.

Anda mungkin bertanya-tanya memangnya mengidentifikasi kucing itu sangat keren? tetapi di bidang pembelajaran mesin, hal ini menjadi berita yang sangat mengejutkan.

Sebelumnya kita sudah membahas bahwa alam pembelajaran mesin, manusia yang perlu membaca data untuk mendefinisikan karakteristik, bukan.

Untuk mengajari computer, terutama dalam pengenalan gambar, kita perlu membuat dan memasukkan definisi kucing itu sendiri, seperti seberapa besar mata dan hidungnya, kumisnya ada berapa, atau seberapa panjang bulunya.

Namun, dalam pembelajaran mendalam, kita tidak perlu lagi memasukkan data seperti itu, karena deep learning sudah berhasil mengekstraksi hal-hal yang berbeda dari banyak video yang diposting di YouTube, yang akhirnya berhasil mengidentifikasi objek yang disebut kucing itu.

Hal ini tidak meminta manusia membaca data satu per satu atau menyiapkan banyak gambar hanya untuk mengidentifikasi kucing lebih jauh, namun deep learning mengidentifikasi kucing dari data video yang tidak terbatas.

Prestasi ini adalah penemuan bahwa kecerdasan buatan telah berkembang sedemikian rupa, bahkan bisa dikatakan sebagai pencapaian besar dalam sejarah peradaban manusia.

## **BAB II**

---

### **Pengenalan CNN, R-CNN, Fast R-CNN, Faster R-CNN, YOLO dan SSD**

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

##### **2.1.1 Pengertian CNN(Convolutional Neural Network)**

Convolutional Neural Network merupakan salah satu jenis algoritma Deep Learning yang dapat menerima *input* berupa gambar, menentukan aspek atau obyek apa saja dalam sebuah gambar yang bisa digunakan mesin untuk “belajar” mengenali gambar, dan membedakan antara satu gambar dengan yang lainnya.

Secara garis besar Convolutional Neural Network (CNN) tidak jauh beda dengan neural network biasanya. CNN terdiri dari neuron yang memiliki weight, bias dan activation function. Convolutional layer juga terdiri dari neuron yang tersusun sedemikian rupa sehingga membentuk sebuah filter dengan panjang dan tinggi (pixels) serta memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan.

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.

### **2.1.2 Sejarah CNN(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 Wiesel 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 Wiesel menemukan 2 tipe visual cortex, yaitu simple cell dan complex cell. Berdasarkan hasil pengamatan ini, pada tahun 1980-an Kuniyiko 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.

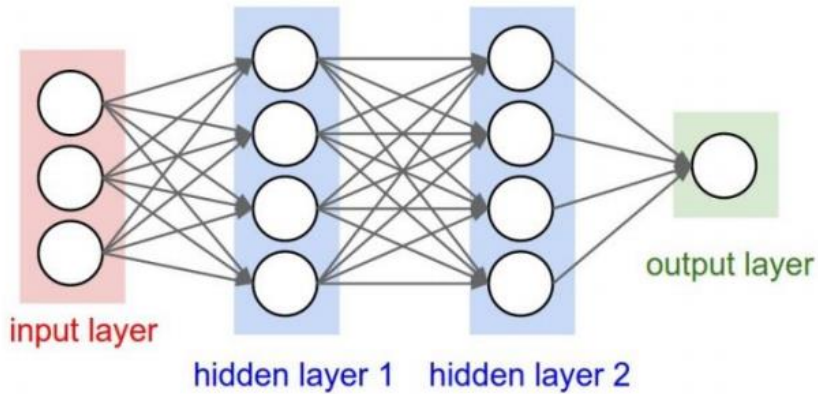
CNN memiliki kesamaan struktur dengan artificial neural network. Pada kasus klasifikasi citra, CNN menerima citra input atau masukan kemudian diproses dan diklasifikasi ke kategori tertentu (mis. pesawat, kapal, burung, kucing, sapi).

CNN dikembangkan dengan nama NeoCognitron oleh Kuniyuki Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeCun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeCun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode Deep Learning, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode Machine Learning lainnya seperti SVM pada kasus klasifikasi objek pada citra.

### **2.1.3 Cara Kerja CNN(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 atau gambar. 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.



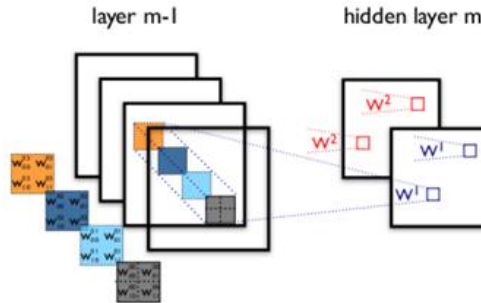


*Gambar 2.1. 1 Arsitektur MLP Sederhana*

Sebuah MLP memiliki layer (kotak merah dan biru) dengan masing-masing layer berisi neuron (lingkaran putih). MLP menerima input data satu dimensi dan mempropagasikan 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.

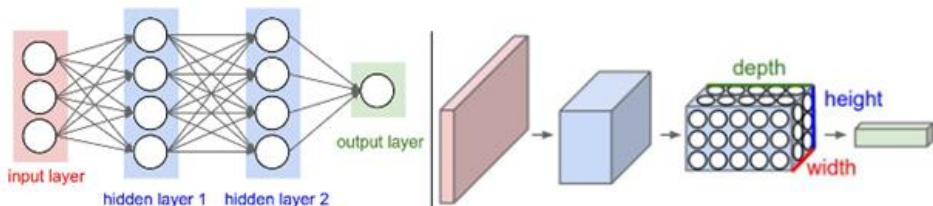
Disetiap data input pada layer dilakukan operasi linear dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi non linear yang disebut sebagai fungsi aktivasi. Pada CNN, data yang dipropagasikan pada jaringan adalah data dua dimensi, sehingga operasi linear dan parameter bobot pada CNN berbeda. Pada CNN operasi linear menggunakan operasi konvolusi, sedangkan bobot tidak lagi satu dimensi saja, namun berbentuk empat dimensi yang merupakan kumpulan kernel konvolusi.

*neuron input  $\times$  neuron output  $\times$  tinggi  $\times$  lebar*



*Gambar 2.1. 2 Proses Konvolusi pada CNN*

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.



*Gambar 2.1. 3 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 diatas , 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.

Secara garis besarnya, CNN memanfaatkan proses konvolusi dengan menggerakan sebuah kernel konvolusi (filter) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan filter yang digunakan.

Berikut ilustrasi langkah Kerja CNN:

1. Memecah gambar menjadi gambar yang lebih kecil yang tumpang tindih

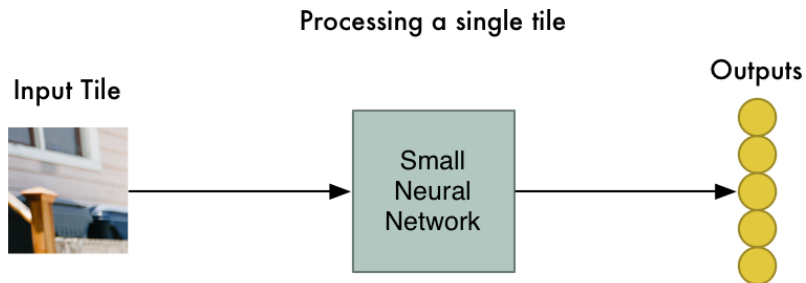


*Gambar 2.1. 4 Memecah gambar*

Dengan ini, gambar asli dari seorang anak kecil diatas menjadi 77 gambar yang lebih kecil dengan konvolusi yang sama.

2. Memasukkan setiap gambar yang lebih kecil ke *small neural network*

Setiap gambar kecil dari hasil konvolusi tersebut kemudian dijadikan input untuk menghasilkan sebuah representasi fitur. Hal ini memberikan CNN kemampuan mengenali sebuah objek, dimanapun posisi objek tersebut muncul pada sebuah gambar.

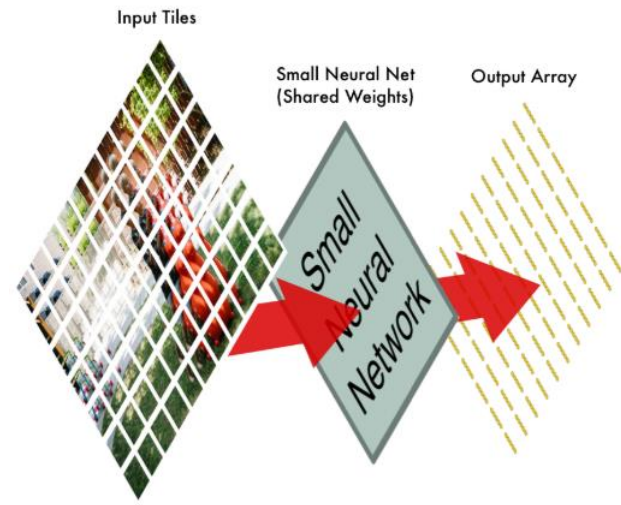


*Gambar 2.1. 5 representasi fitur*

Diulang sebanyak 77 kali pada masing-masing gambar kecilnya. Proses ini dilakukan untuk semua bagian dari masing-masing gambar kecilnya, dengan menggunakan filter yang sama. Dengan kata lain, setiap bagian gambar akan memiliki faktor pengali yang sama, atau dalam konteks *neural network* disebut sebagai *weights sharing*. Jika ada sesuatu yang tampak menarik di setiap gambarnya, maka akan ditandai bagian itu sebagai *object of interest*.

3. Menyimpan hasil dari masing-masing gambar kecil ke dalam array baru.

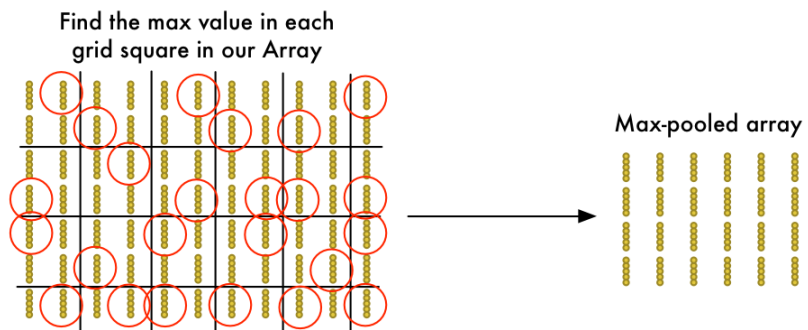
Maka ilustrasinya akan seperti gambar 2.1.6



Gambar 2.1. 6 Simpan kedalam array baru

#### 4. *Downsampling*

Pada langkah 3, array masih terlalu besar, maka untuk mengecilkan ukuran array nya digunakan *downsampling* yang penggunaannya dinamakan max pooling atau mengambil nilai *pixel* terbesar di setiap *pooling kernel*. Dengan begitu, sekalipun mengurangi jumlah parameter, informasi terpenting dari bagian tersebut tetap diambil.

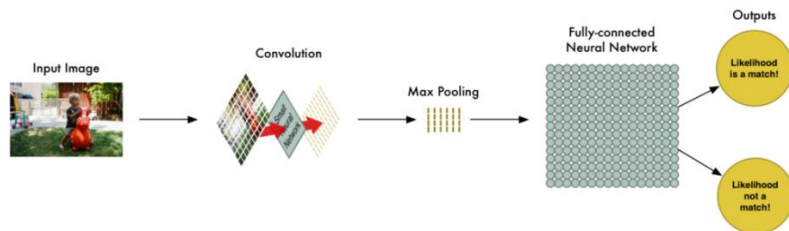


Gambar 2.1. 7 Ilustrasi Max Pooling

## 5. Membuat prediksi

Sejauh ini, kita telah merubah dari gambar yang berukuran besar menjadi array yang cukup kecil. Nah, array merupakan sekelompok angka, jadi dengan menggunakan array kecil itu kita bisa inputkan ke dalam jaringan saraf lain. Jaringan saraf yang paling terakhir akan memutuskan apakah gambarnya cocok atau tidak. Untuk memberikan perbedaan dari langkah konvolusi, maka bisa kita sebut dengan “fully connected” network.

Secara garis besarnya, langkah-langkah diatas tampak seperti gambar berikut ini :

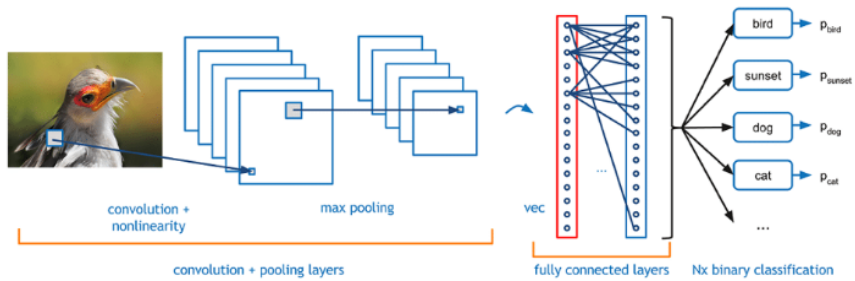


*Gambar 2.1. 8 Ilustrasi dari setiap langkah*

### 2.1.4 Arsitektur CNN(Convolutional Neural Network)

Struktur CNN terdiri dari input, proses ekstraksi fitur, proses klasifikasi dan output. Proses ekstraksi dalam CNN terdiri dari beberapa lapisan tersembunyi atau hidden layer, yaitu lapisan konvolusi, fungsi aktivasi (ReLU), dan pooling. CNN bekerja secara hierarki, sehingga output pada lapisan konvolusi pertama digunakan sebagai input pada lapisan konvolusi selanjutnya. Pada proses klasifikasi terdiri dari fully-connected dan fungsi aktivasi (softmax) yang outputnya berupa hasil klasifikasi. Hal yang membedakan CNN dengan ANN adalah CNN memiliki arsitektur tambahan yang dioptimisasi untuk fitur yang ada

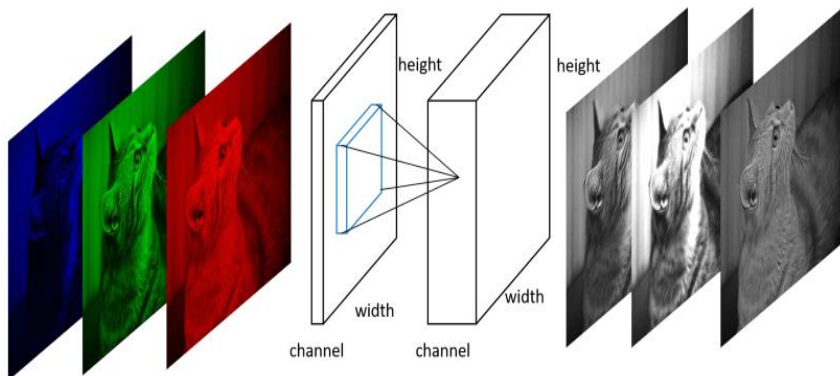
pada citra input. komponen-komponen utama yang ada di dalam Convolutional Neural Networks adalah:



*Gambar 2.1. 9 Arsitektur CNN*

- **Input Layer**

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



*Gambar 2.1. 10 Input layer RGB*

- **Convolution Layer**

Convolution Layer adalah inti dari CNN. Convolution Layer menghasilkan citra baru yang menunjukkan fitur dari citra input. Dalam proses tersebut, Convolution Layer menggunakan filter pada setiap citra yang menjadi masukan. Filter pada layer ini berupa array 2 dimensi bisa berukuran  $5 \times 5$ ,  $3 \times 3$  atau  $1 \times 1$ . Proses convolution dengan menggunakan filter pada layer ini akan menghasilkan feature map yang akan digunakan pada activation layer.

Convolution Layer melakukan operasi konvolusi pada output dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari sebuah CNN. Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel pada citra disemua offset yang memungkinkan. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Konvolusi akan menghasilkan transformasi linear dari data input sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan input pada CNN.



1	1	1	0	0
0	1	1	1	0
0	0	1x1	1x0	1x1
0	0	1x0	1x1	0x0
0	1	1x1	0x0	0x1

4	3	4
2	4	3
2	3	4

*Gambar 2.1. 11 Convolution Layer*

Convolutional Layer akan menghitung output dari neuron yang terhubung ke daerah lokal dalam input, masing-masing menghitung produk titik antara bobot mereka dan wilayah kecil yang terhubung ke dalam volume input.

Ada proses dalam hal melakukan pergeseran suatu convolution layer berikut penjelasannya:

#### Stride

Stride adalah parameter yang menentukan berapa jumlah pergeseran filter. Jika nilai stride adalah 1, maka conv. filter akan bergeser sebanyak 1 pixel secara horizontal lalu vertical. Pada ilustrasi diatas, stride yang digunakan adalah 2. 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. Namun perlu diperhatikan bahwa dengan menggunakan *stride* yang kecil kita tidak selalu akan mendapatkan performa yang bagus.

## Padding

Padding atau zero padding adalah parameter menentukan jumlah pixel (berisi nilai 0) yang akan ditambahkan di setiap sisi dari input. Hal ini digunakan dengan tujuan untuk memanipulasi dimensi output dari conv. layer (feature map).

Dengan menggunakan padding, kita akan dapat mengukur dimensi output agar tetap sama seperti dimensi input atau setidaknya tidak berkurang secara drastis. Sehingga kita bisa menggunakan conv. layer yang lebih dalam sehingga lebih banyak feature yang berhasil di-extract. Meningkatkan performa model karena conv. layer akan fokus pada informasi yang sebenarnya yaitu yang berada diantara zero padding tersebut. Pada ilustrasi diatas, dimensi dari input sebenarnya adalah  $5 \times 5$ , jika dilakukan convolution dengan filter  $3 \times 3$  dan stride sebesar 2, maka akan didapatkan feature map dengan ukuran  $2 \times 2$ . Namun jika ditambahkan zero padding sebanyak 1, maka feature map yang dihasilkan berukuran  $3 \times 3$  (lebih banyak informasi yang dihasilkan). Untuk menghitung dimensi dari feature map kita bisa gunakan rumus sebagai berikut.

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

*Gambar 2.1. 12 rumus padding*

Keterangan :

W = Panjang/Tinggi Input

N = Panjang/Tinggi Filter

P = Zero Padding

S = Stride

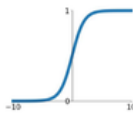
- **Activation Layer**

Activation Layer adalah 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. ini bertujuan untuk meneruskan nilai yang menampilkan fitur dominan dari citra yang masuk ke layer berikutnya. Fungsi aktivasi yang umum digunakan bisa dilihat pada gambar berikut.

## Activation Functions

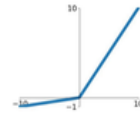
### Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



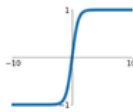
### Leaky ReLU

$$\max(0.1x, x)$$



### tanh

$$\tanh(x)$$

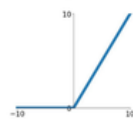


### Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

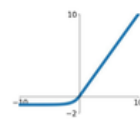
### ReLU

$$\max(0, x)$$



### ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Gambar 2.1. 13 Activation Functions

Fungsi aktivasi berada pada tahap sebelum melakukan *pooling layer* dan setelah melakukan proses konvolusi. Pada tahap ini, nilai hasil konvolusi dikenakan fungsi aktivasi atau *activation function*. Terdapat beberapa fungsi aktivasi yang sering digunakan pada *convolutional network*, diantaranya *tanh()* atau *reLU*. Aktivasi *reLU* menjadi pilihan bagi beberapa peneliti karena sifatnya yang lebih berfungsi dengan baik.

Fungsi yang digunakan untuk aktivasi pada reLU, fungsi reLU adalah nilai *output* dari *neuron* bisa dinyatakan sebagai 0 jika *inputnya* adalah *negatif*. Jika nilai *input* dari fungsi aktivasi adalah *positif*, maka *output* dari *neuron* adalah nilai *input* aktivasi itu sendiri.

- **Pooling Layer**

Pooling layer biasanya berada setelah conv. layer. Pada prinsipnya pooling layer terdiri dari sebuah filter dengan ukuran dan stride tertentu yang bergeser pada seluruh area feature map. Pooling yang biasa digunakan adalah Max Pooling dan Average Pooling. Tujuan dari penggunaan pooling layer adalah mengurangi dimensi dari feature map (downsampling), sehingga mempercepat komputasi karena parameter yang harus di update semakin sedikit dan mengatasi overfitting.

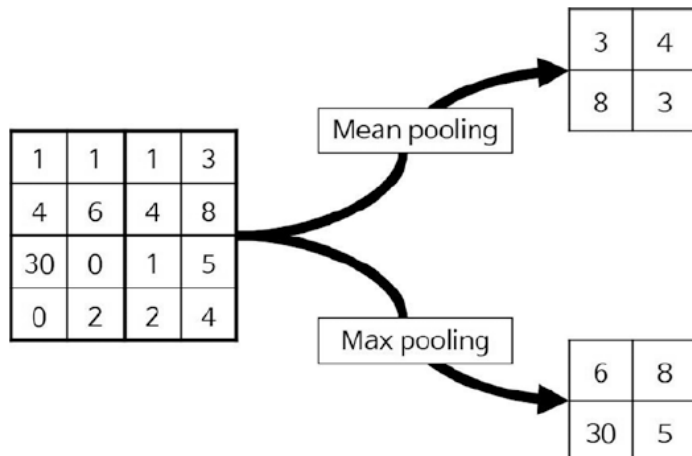
Pooling layer menerima input dari activation layer kemudian mengurangi jumlah parameternya. Pooling juga biasa disebut subsampling atau downsampling yang mengurangi dimensi dari feature map tanpa menghilangkan informasi penting di

dalamnya. Proses dalam pooling layer cukup sederhana. pertama-tama kita menentukan ukuran downsampling yang akan digunakan pada feature map, misalnya  $2 \times 2$ . Setelah itu kita akan melakukan proses pooling pada feature map, sebagai contoh kita akan menggunakan feature map berukuran  $4 \times 4$  berikut.

1	1	1	3
4	6	4	8
30	0	1	5
0	2	2	4

*Gambar 2.1. 14 sampel untuk pooling*

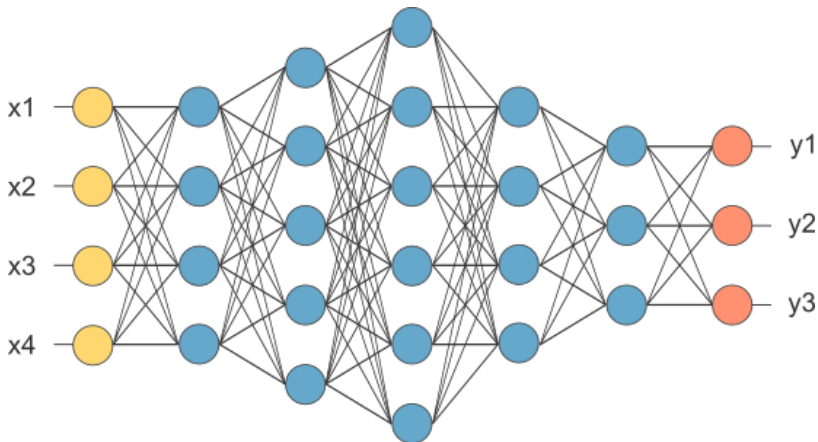
setelah itu kita akan menggunakan matrix  $2 \times 2$  untuk melakukan proses pooling. proses Pooling sendiri ada beberapa macam seperti Max pooling, Mean pooling Sum pooling.



*Gambar 2.1. 15 feature map 2x2*

- **Fully Connected Layer**

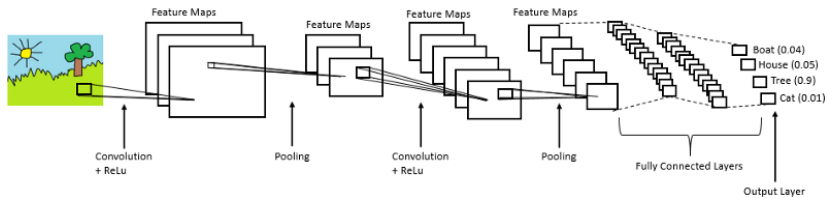
setelah melewati proses-proses diatas, hasil dari pooling layer digunakan menjadi masukan untuk Fully connected layer. Layer ini memiliki kesamaan struktur dengan Artificial Neural Network pada umumnya yaitu memiliki input layer, hidden layer dan output layer yang masing-masing memiliki neuron-neuron yang saling terhubung dengan neuron-neuron di layer tetangganya. gambar di bawah ini merupakan contoh Fully Connected Layer.



*Gambar 2.1. 16 Fully Connected Layer*

pada gambar diatas dapat dilihat sebelum hasil pooling digunakan sebagai input, hasil pooling terlebih dahulu diubah menjadi vektor (x1, x2, x3, dst) kemudian dari sini diproses ke dalam Fully Connected Layer. Pada layer terakhir di dalam Fully Connected layer akan digunakan fungsi aktivasi sigmoid atau

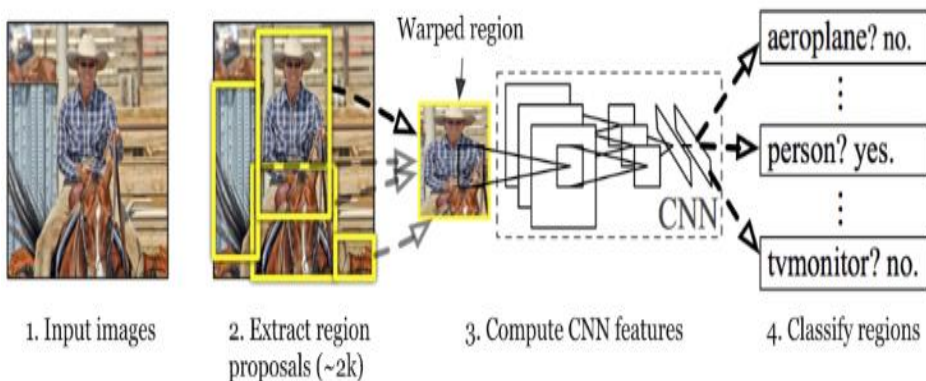
softmax untuk menentukan klasifikasi dari citra inputan atau masukan yang dari Input Layer CNN.



*Gambar 2.1. 17 langkah-langkah CNN*

## 2.2 PENGANTAR R-CNN (REGION CONVOLUTIONAL NEURAL NETWORKS)

R-CNN adalah singkatan dari "wilayah berbasis convolutional Neural Networks". Ide utama terdiri dari dua langkah. Pertama, menggunakan pencarian selektif, itu mengidentifikasi jumlah yang dikelola kotak daerah objek bounding kandidat ("wilayah yang menarik" atau "ROI"). Dan kemudian itu mengekstrak fitur CNN dari masing-masing daerah secara independen untuk klasifikasi.



*Gambar 2.2. 1 architecture of R-CNN*

### 2.1.3 Cara Kerja R-CNN(REGION-Convolutional Neural Network)

Cara kerja R-CNN dapat diringkas sebagai berikut:

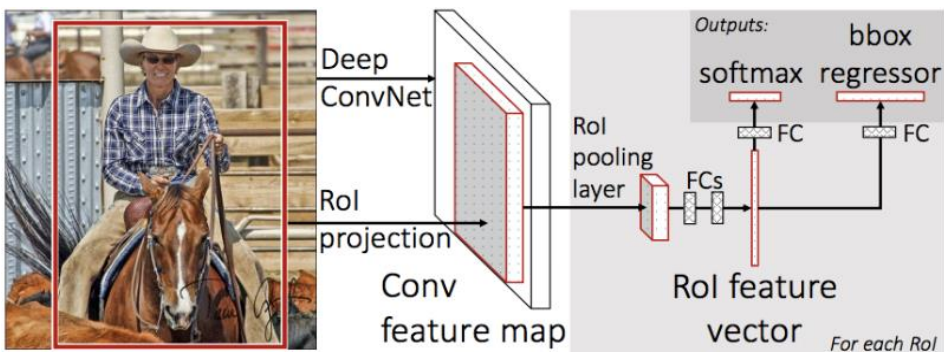
1. Pra-latih jaringan CNN pada tugas klasifikasi gambar; misalnya, VGG atau ResNet dilatih tentang dataset ImageNet. Tugas klasifikasi melibatkan kelas  $N$ .
2. Usulkan wilayah minat kategori-bebas dengan pencarian selektif ( $\sim 2k$  kandidat per gambar). Wilayah-wilayah itu mungkin berisi objek sasaran dan ukurannya berbeda.
3. Calon daerah dibengkokkan memiliki ukuran tetap seperti yang disyaratkan oleh CNN.
4. Terus menyempurnakan CNN pada daerah proposal yang melengkung untuk kelas  $K + 1$ ; Satu kelas tambahan mengacu pada latar belakang (tidak ada objek yang menarik). Pada tahap fine-tuning, kita harus menggunakan tingkat pembelajaran yang jauh lebih kecil dan mini-batch melebihi-lebihkan kasus positif karena sebagian besar wilayah yang diusulkan hanyalah latar belakang.
5. Mengingat setiap wilayah gambar, satu propagasi maju melalui CNN menghasilkan vektor fitur. Vektor fitur ini kemudian dikonsumsi oleh SVM biner yang dilatih untuk setiap kelas secara mandiri. Sampel positif diusulkan daerah dengan surat hutang (persimpangan atas Serikat) tumpang tindih ambang batas  $\geq 0,3$ , dan sampel negatif yang tidak relevan lainnya.



6. Untuk mengurangi kesalahan lokalisasi, model regresi dilatih untuk memperbaiki prediksi jendela deteksi pada kotak koreksi offset menggunakan fitur CNN.

## 2.3 PENGANTAR FAST REGION - CONVOLUTION NEURAL NETWORK (FATER R-CNN)

Untuk membuat R-CNN lebih cepat, para ilmuwan meningkatkan prosedur pelatihan dengan menyatukan tiga model independen menjadi satu kerangka kerja bersama yang terlatih dan meningkatkan hasil komputasi bersama, bernama Fast R-CNN. Alih-alih mengekstraksi vektor fitur CNN secara independen untuk setiap proposal kawasan, model ini mengumpulkannya menjadi satu CNN meneruskan seluruh gambar dan proposal kawasan berbagi matriks fitur ini. Kemudian matriks fitur yang sama bercabang untuk digunakan untuk mempelajari pengklasifikasi objek dan regresi kotak-terikat. Kesimpulannya, pembagian komputasi mempercepat R-CNN.



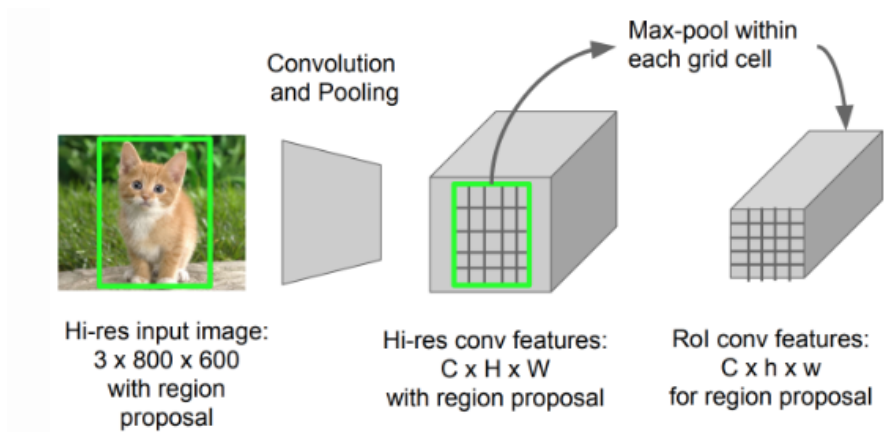
*Gambar 2.3. 1 Fast Region-Convolution neureal network*

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 mengumpankan 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.

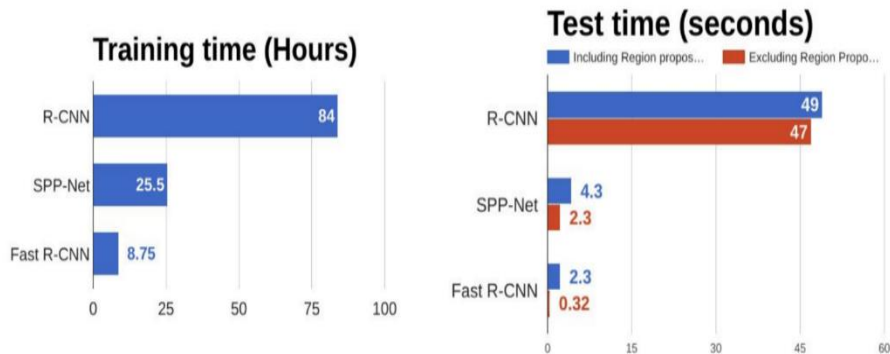
- **RoI Pooling**

Ini adalah jenis penyatuan maks untuk mengonversi fitur di wilayah yang diproyeksikan dari gambar dalam berbagai ukuran,  $h \times w$ , menjadi jendela kecil,  $H \times W$ . Wilayah input dibagi menjadi kisi  $H \times W$ , kira-kira setiap jendela di bawah ukuran  $h / H \times w / W$ . Kemudian menerapkan max-pooling di setiap kisi.



*Gambar 2.3. 2 RoI Pooling*

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 2.3. 3 grafik fast r-CNN*

Dari gambar grafik diatas 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.

### **2.3.1 Alur Kerja Model R-CNN**

Cara kerja Cepat R-CNN diringkas sebagai berikut; banyak langkah yang sama seperti di R-CNN:

1. Pertama, pra-kereta jaringan saraf convolutional pada tugas klasifikasi gambar.
2. Mengusulkan daerah dengan pencarian selektif (~ 2k calon per gambar).
3. Mengubah CNN pra-dilatih:

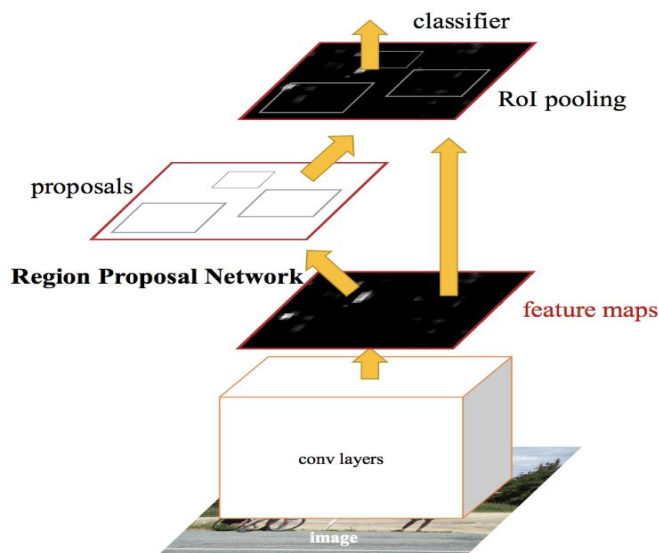
- Ganti layer maks penggabungan terakhir dari CNN yang telah dilatih sebelumnya dengan layer ROI penggabungan . ROI penggabungan layer output tetap panjang vektor fitur proposal wilayah. Berbagi komputasi CNN membuat banyak akal, karena banyak wilayah proposal gambar yang sama sangat tumpang tindih.
- Ganti layer terakhir yang terhubung sepenuhnya dan layer softmax terakhir (kelas  $K$ ) dengan layer yang terhubung sepenuhnya dan softmax pada kelas  $K + 1$ .

#### 4. Akhirnya model Branch menjadi dua layer output:

- Sebuah pengukur softmax  $K + 1$  kelas (sama seperti di R-CNN, + 1 adalah "latar belakang" kelas), keluaran probabilitas diskrit distribusi per ROI.
- Sebuah bounding-Box model regresi yang memprediksi offset relatif terhadap RoI asli untuk masing-masing kelas  $K$ .

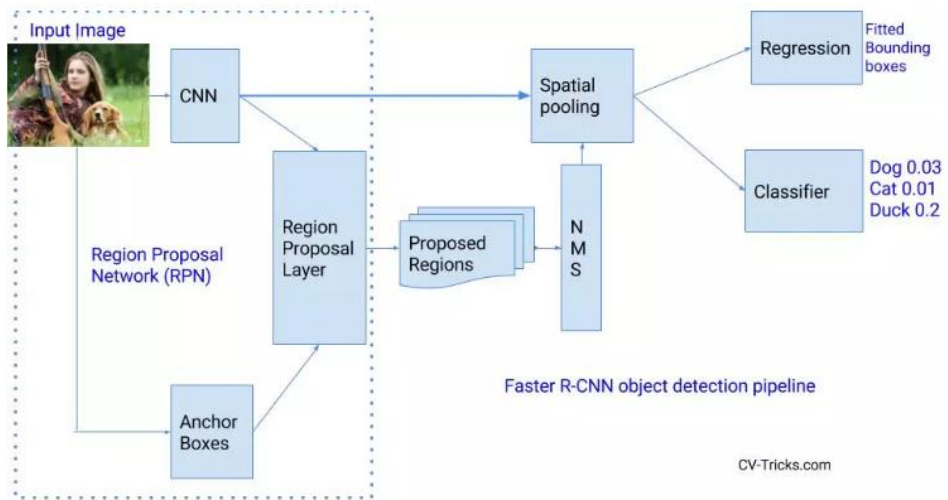
## 2.4 FASTER REGION - CONVOLUTION NEURAL NETWORK (FASTER R-CNN)

Solusi percepatan intuitif adalah mengintegrasikan algoritma proposal wilayah ke dalam model CNN. Lebih cepat R-CNN melakukan persis ini: membangun model tunggal, terpadu yang terdiri dari RPN (region proposal network) dan R-CNN cepat dengan lapisan fitur konvolusional bersama.



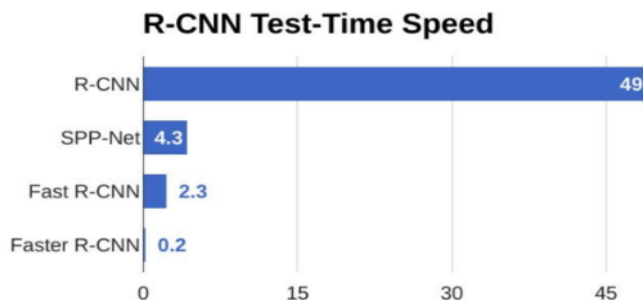
*Gambar 2.4. 1 FASTER REGION - CONVOLUTION NEURAL NETWORK*

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.



Gambar 2.4. 2 faster r-CNN

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 2.4. 3 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 mendeteksi objek. Sebelum faster R-CNN diperkenalkan, berikut perkembangan metode R-CNN seiring berjalannya waktu.



*Gambar 2.4. 4 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

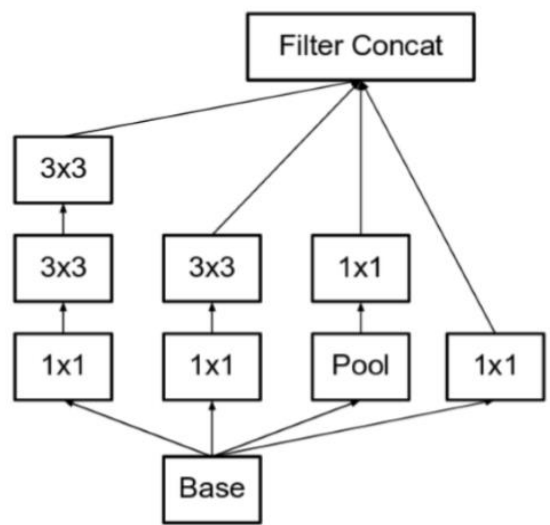


- Lalu dilakukan seleksi untuk setiap box yang dihasilkan sesuai dengan skor adanya objek pada box tersebut
- 2. Jalankan CNN untuk setiap bounding box yang dihasilkan dari RPN
- 3. Setiap output yang dihasilkan CNN menjadi input SVM (Support Vector Machine) untuk klasifikasi kelasnya dan juga sebagai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

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

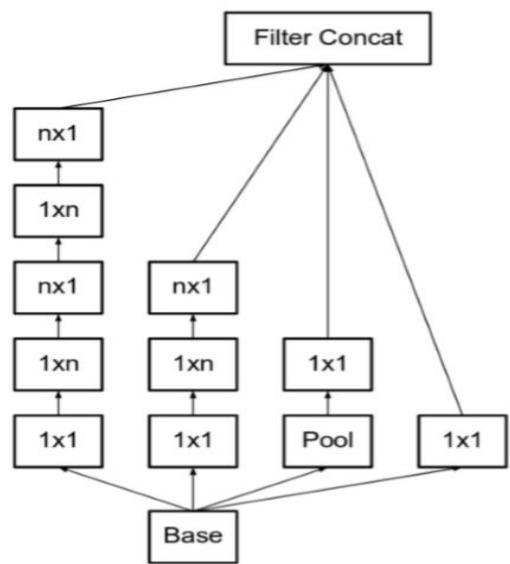
CNN memiliki berbagai arsitektur, salah satunya dalam penelitian ini adalah Inception V2. Arsitektur dari Inception V2 dirancang untuk mengurangi kompleksitas CNN, yang dilakukan dengan cara menyusun arsitektur yang lebih melebar dari pada mendalam. Inception V2 memiliki 3 modul yang ditunjukkan oleh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

a. Modul Pertama Inception V2



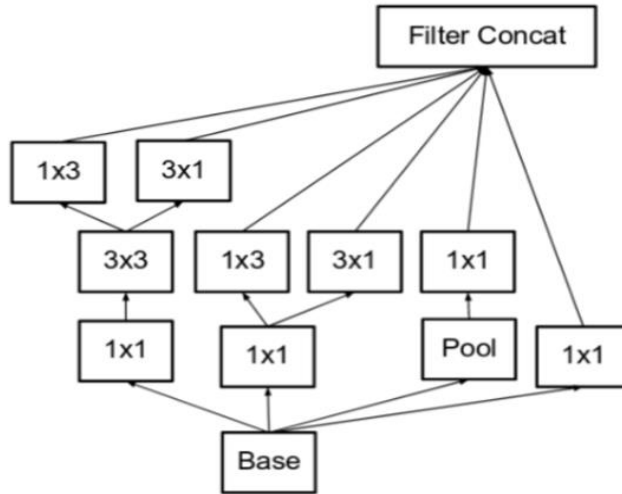
Gambar 2.4. 5 modul pertama inception v2

b. Modul kedua Inception V2



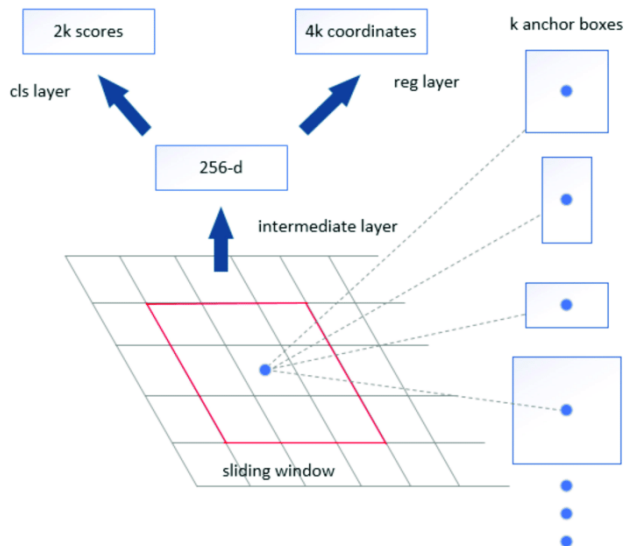
Gambar 2.4. 6 Modul kedua Inception V2

c. Modul Ketiga Inception V2



*Gambar 2.4. 7 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 2.4. 8 Region Proposal Network (RPN)*

## Alur Kerja Mode

1. Pra-melatih jaringan CNN pada tugas klasifikasi gambar.
2. Menyempurnakan RPN (region proposal network) end-to-end untuk tugas proposal wilayah, yang diinisialisasi oleh classifier gambar pra-kereta. Sampel positif memiliki surat hutang (persimpangan-over-Union)  $> 0,7$ , sementara sampel negatif memiliki surat hutang  $< 0,3$ .
  - Geser jendela spasial  $n \times n$  kecil di atas peta fitur Konv dari seluruh gambar.
  - Di tengah setiap jendela geser, kami memprediksi beberapa wilayah dengan berbagai skala dan rasio secara bersamaan. Sebuah jangkar adalah kombinasi dari (geser jendela pusat, skala, rasio). Misalnya, 3 timbangan + 3 rasio  $\Rightarrow k = 9$  jangkar di setiap posisi geser.
3. Latih cepat model deteksi objek R-CNN menggunakan proposal yang dihasilkan oleh RPN saat ini
4. Kemudian gunakan jaringan Fast R-CNN untuk menginisialisasi pelatihan RPN. Sambil menjaga lapisan convolutional bersama, hanya menyempurnakan lapisan RPN-spesifik. Pada tahap ini, Jaringan RPN dan deteksi telah berbagi convolutional Layers!
5. Akhirnya menyempurnakan lapisan unik Fast R-CNN
6. Langkah 4-5 dapat diulang untuk melatih RPN dan Fast R-CNN alternatif jika diperlukan

## **2.5 PENGANTAR YOLO (YOU ONLY LOOK ONCE)**

### **2.5.1 Pengertian YOLO (YOU ONLY LOOK ONCE)**

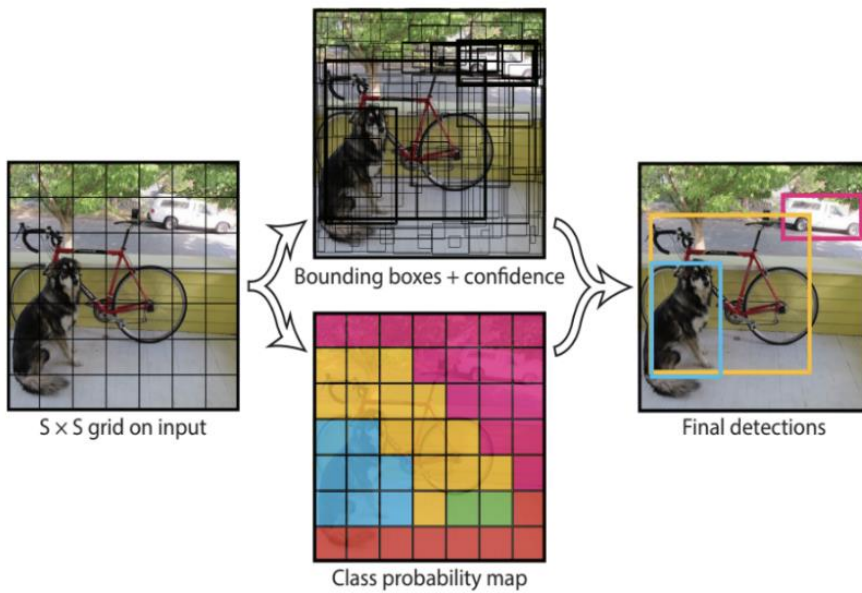
You Only Look Once (Yolo) adalah sebuah algoritma yang dikembangkan untuk mendeteksi sebuah objek secara real-time. Sistem pendeteksian yang dilakukan adalah dengan menggunakan repurpose classifier atau localizer untuk melakukan deteksi. Sebuah model diterapkan pada sebuah citra di beberapa lokasi dan skala. Daerah dengan citra yang diberi score paling tinggi akan dianggap sebagai sebuah pendeteksian.

Yolo menggunakan pendekatan jaringan saraf tiruan (JST) untuk mendeteksi objek pada sebuah citra. Jaringan ini membagi citra menjadi beberapa wilayah dan memprediksi setiap kotak pembatas dan probabilitas untuk setiap wilayah. Kotak-kotak pembatas ini kemudian dibandingkan dengan setiap probabilitas yang diprediksi. Yolo memiliki beberapa kelebihan dibandingkan dengan sistem yang berorientasi pada classifier, terlihat dari seluruh citra pada saat dilakukan test dengan prediksi yang diinformasikan secara global pada citra.

Hal tersebut juga membuat prediksi dengan sintesis jaringan saraf ini tidak seperti sistem Region Convolutional Neural Network (R-CNN) yang membutuhkan ribuan untuk sebuah citra sehingga membuat Yolo lebih cepat hingga beberapa kali daripada R-CNN.

YOLO populer karena mencapai akurasi tinggi dan juga dapat berjalan secara real-time. Algoritma "only looks once" atau hanya terlihat sekali, pada gambar dalam arti hanya membutuhkan satu

propagasi maju melewati jaringan saraf untuk membuat prediksi. Setelah penindasan non-max (yang memastikan algoritma deteksi objek hanya mendeteksi setiap objek sekali), kemudian output objek yang dikenali bersama dengan kotak pembatas.

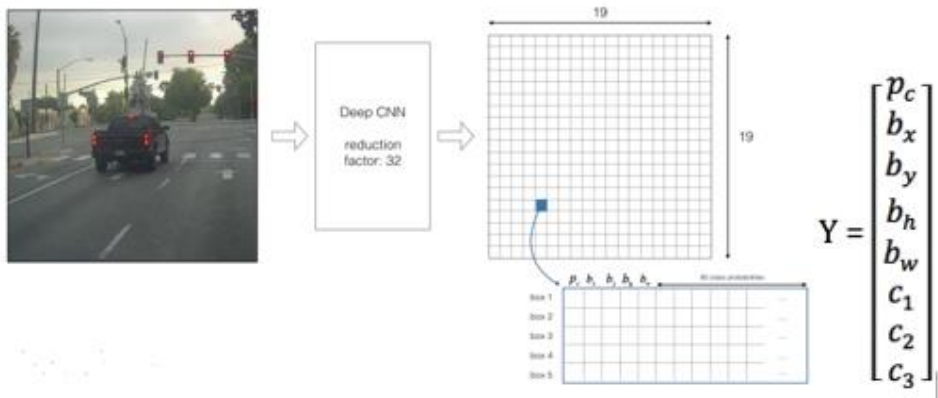


*Gambar 2.5. 1 You Only Look Once (Yolo)*

### 2.5.2 Cara kerja YOLO (YOU ONLY LOOK ONCE)

Klasifikasi secara umum adalah proses untuk mengidentifikasi label dari data yang diuji, sedangkan pada Yolo, klasifikasinya dengan localization, yaitu terdapat tambahan pemberian lokasi objek dalam bentuk bounding box (bx,by,bh,bw)

Kami mulai dengan menempatkan kisi di atas gambar input. Kemudian, untuk masing-masing sel kisi, kemudian menjalankan algoritma klasifikasi dan lokalisasi



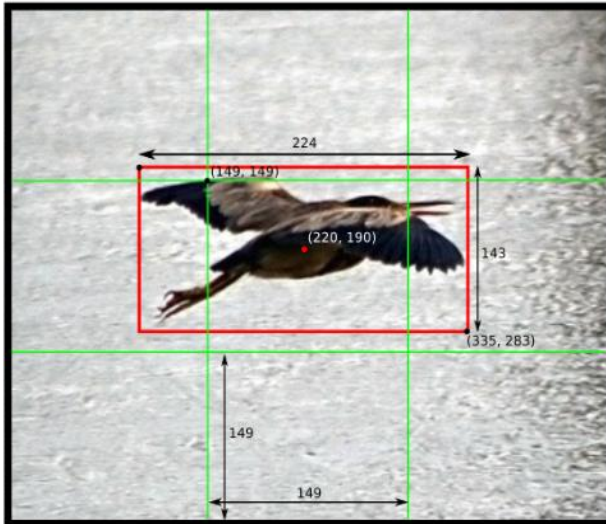
Gambar 2.5. 2 ilustrasi proses prediksi box

Untuk setiap sel, kita akan mendapatkan hasil apakah ada objek atau tidak. Sebagai contoh:

$$\text{Object of class 2 (i.e. car) detected: } \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} ; \quad \text{No object detected: } \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

Gambar 2.5. 3 perbedaan ada objek dan tidak ada objek

(0, 0)



$$\begin{aligned}x &= (220-149) / 149 = 0.48 \\y &= (190-149) / 149 = 0.28 \\w &= 224 / 448 = 0.50 \\h &= 143 / 448 = 0.32\end{aligned}$$

(447, 447)

*Gambar 2.5. 4 contoh bentuk bounding box*

Untuk memprediksi kelas objek dan kotak pembatas yang menentukan lokasi objek. Setiap kotak pembatas dapat dideskripsikan menggunakan empat descriptor:

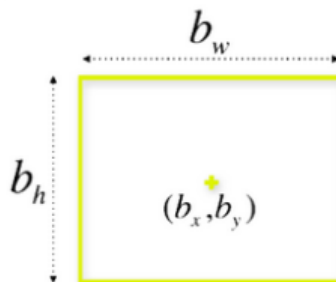
Titik tengah bounding box ( $\mathbf{b_x \ b_y}$ )

Lebar ( $\mathbf{b_w}$ )

Tinggi ( $\mathbf{b_h}$ )

Kelas objek (mobil, orang, lampu jalan,.....)

$$y = (p_c, b_x, b_y, b_h, b_w, c)$$



*Gambar 2.5. 5 kotak pembatas objek*



## Loss function

YOLO memprediksi beberapa kotak pembatas per sel kotak. Untuk menghitung kerugian untuk positif sejati, kami hanya ingin salah satu dari mereka **bertanggung jawab** atas objek tersebut. Untuk tujuan ini, kami memilih yang memiliki IoU tertinggi (persimpangan atas persatuan) dengan kebenaran dasar. Strategi ini mengarah ke spesialisasi di antara prediksi kotak terikat. Setiap prediksi menjadi lebih baik dalam memprediksi ukuran dan rasio aspek tertentu.

YOLO menggunakan jumlah kesalahan kuadrat antara prediksi dan kebenaran dasar untuk menghitung kerugian. Fungsi kerugian terdiri dari:

- klasifikasi kerugian (classification loss)  
Jika *suatu objek terdeteksi*, kehilangan klasifikasi di setiap sel adalah kesalahan kuadrat dari probabilitas bersyarat kelas untuk setiap kelas
- lokalisasi loss (kesalahan antara kotak batas diprediksi dan kebenaran tanah). “localization loss”.

Kehilangan lokalisasi mengukur kesalahan dalam prediksi lokasi dan ukuran kotak batas YOLO memprediksi dari lebar dan tinggi kotak pembatas alih-alih lebar dan tinggi. Selain itu, untuk lebih menekankan akurasi kotak batas

- percaya diri loss (objektivitas kotak)” confidence loss”

Sebagian besar kotak tidak mengandung benda apa pun. Ini menyebabkan masalah ketidakseimbangan kelas, yaitu kami melatih model untuk mendeteksi latar belakang lebih sering daripada mendeteksi objek.

Semua algoritme deteksi objek sebelumnya menggunakan wilayah untuk melokalkan objek dalam gambar. Jaringan tidak melihat gambar lengkap. Sebagai gantinya, bagian dari gambar yang memiliki probabilitas tinggi mengandung objek. Di YOLO, satu jaringan konvolusional memprediksi kotak pembatas dan probabilitas kelas untuk kotak-kotak ini.

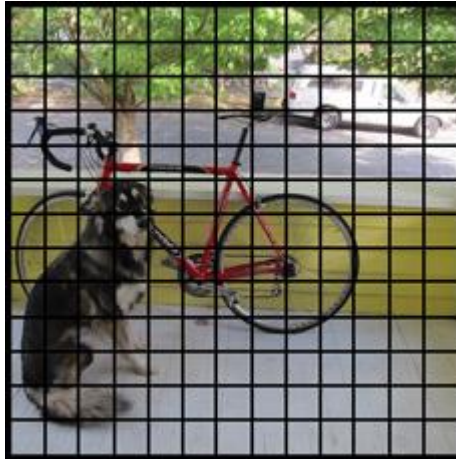
Cara kerja YOLO adalah mengambil gambar dan membaginya menjadi kisi  $S \times S$ , di dalam masing-masing kisi tersebut kemudian mengambil kotak pembatas. Untuk setiap kotak pembatas, jaringan mengeluarkan probabilitas kelas dan mengimbangi nilai untuk kotak pembatas. Kotak pembatas yang memiliki probabilitas kelas di atas nilai ambang dipilih dan digunakan untuk menemukan objek di dalam gambar.

Keterbatasan algoritma YOLO adalah bahwa ia berjuang dengan benda-benda kecil di dalam gambar, misalnya mungkin mengalami kesulitan dalam mendeteksi kawanan burung. Ini karena kendala spasial dari algoritma.

Secara umum metode YOLO mempunyai tahapan yang paling simpel dibanding SSD maupun faster R-CNN. Sebagian besar tahapan YOLO seperti halnya klasifikasi gambar menggunakan CNN biasa. YOLO melakukan kalkulasi bounding box dengan satu skala fitur map.

Tahapan Yolo adalah sebagai berikut :

- Input gambar ditandai kedalam sejumlah  $S \times S$  grid box



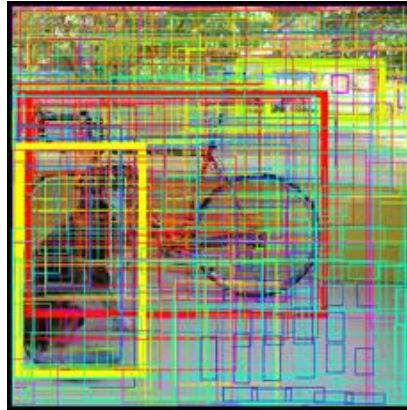
*Gambar 2.5. 6 grid sell pada gambar SXS*

- Setiap grid box akan membentuk sejumlah B “bounding box” yang tetap, dengan berbagai ukuran yang bervariasi. Variasi bounding box sama untuk setiap grid box. Setiap bounding box yang terbentuk dari grid box tertentu berada tepat ditengah grid box tersebut. Ukuran bounding box bisa lebih besar dari ukuran grid box nya.



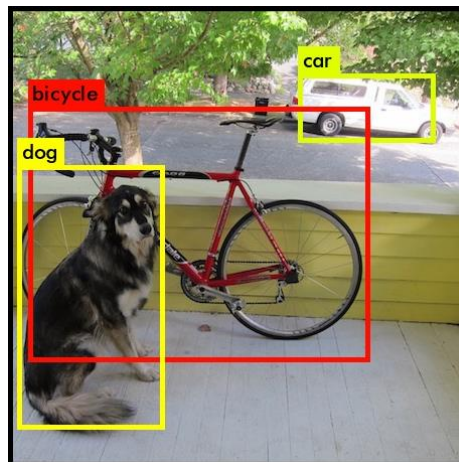
*Gambar 2.5. 7 prediksi box untuk objek*

- Jika pada bounding box terdapat objek, maka grid box yang membentuk bounding box tersebut bertanggung jawab untuk mendeteksi objek tersebut.



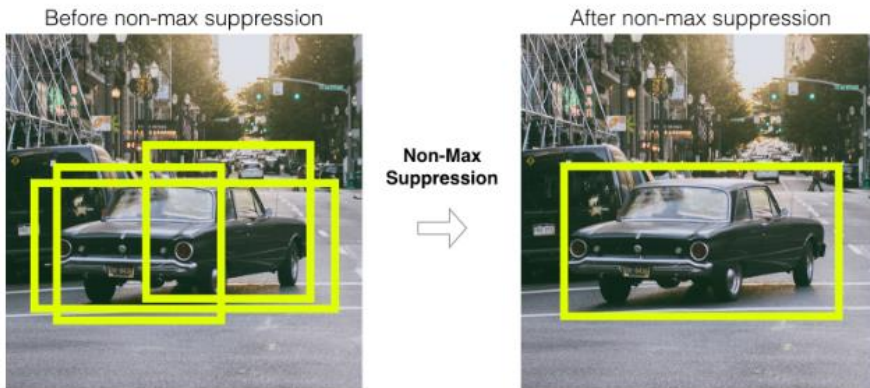
*Gambar 2.5. 8 pembentukan bounding box*

- Satu grid box hanya boleh mendeteksi satu objek saja. Sehingga jika lebih dari satu bounding box untuk satu grid box tertentu mendeteksi adanya objek, maka hanya diperbolehkan untuk memilih satu objek dengan satu bounding box saja yang memiliki tingkat keyakinan paling tinggi.



*Gambar 2.5. 9 box yang mempunyai keyakinan tinggi*

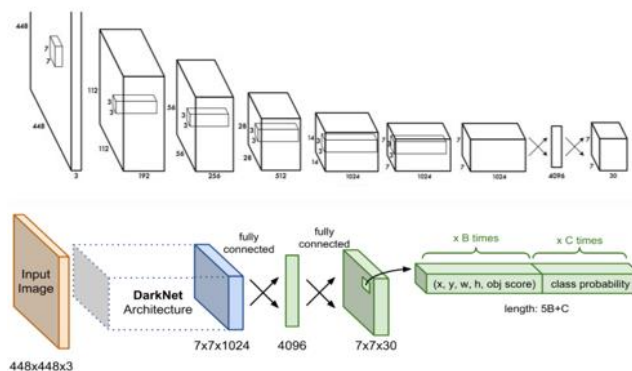
- “Non max suppression” digunakan untuk menentukan saat terdapat banyak bounding box terdeteksi untuk satu objek yang sama.



*Gambar 2.5. 10 proses non max suppression*

YOLO menggunakan jaringan konvolusional tunggal untuk secara bersamaan memprediksi beberapa kotak pembatas dan probabilitas kelas untuk kotak-kotak itu.

Jaringan ini terinspirasi oleh model GoogleNet untuk klasifikasi gambar, dengan modul awal digantikan oleh 1x1 dan 3x3 lapisan Konv. Prediksi akhir bentuk  $s \times s \times (5B+k) \times s \times s \times (5B+k)$  diproduksi oleh dua lapisan yang sepenuhnya terhubung di seluruh peta fitur Konv.



*Gambar 2.5. 11 arsitektur yolo terinspirasi oleh googleNet*

### 2.5.3 Macam-macam YOLO (YOU ONLY LOOK ONCE)

#### • YOLO

YOLO membagi gambar input ke dalam grid  $S \times S$ . Jika pusat suatu objek jatuh ke dalam sel grid, sel grid bertanggung jawab untuk mendeteksi objek tersebut (kami tetapkan objek ke sel kisi tempat pusat objek ada).

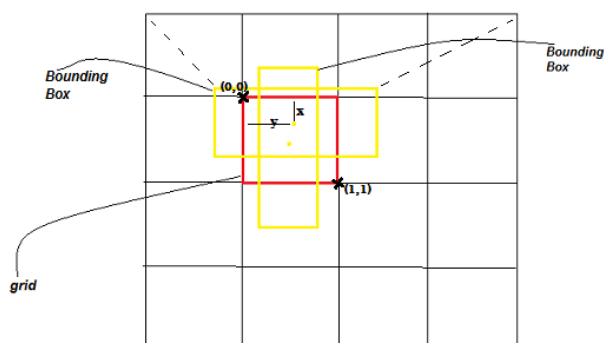


*Gambar 2.5. 12 pembagian grid cell*

YOLO menjalankan masalah klasifikasi dan lokalisasi untuk masing-masing sel kisi secara bersamaan. Karena jaringan klasifikasi dan lokalisasi hanya dapat mendeteksi satu objek yang berarti sel jaringan apa pun dapat mendeteksi hanya satu objek. Karena ide **grid** ini, YOLO menghadapi beberapa masalah:

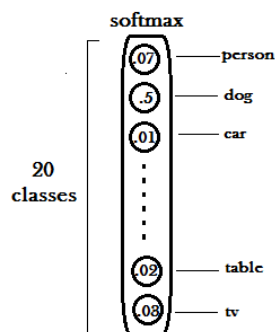
- kisi apa pun dapat mendeteksi hanya satu objek
- Jika sel kotak berisi lebih dari satu objek, model tidak akan dapat mendeteksi semuanya.
- Objek mungkin terletak di lebih dari satu kisi sehingga model dapat mendeteksi taksi lebih dari satu kali (dalam lebih dari satu kisi) dan masalah ini diselesaikan menggunakan penindasan non-max.

Setiap sel grid memprediksi kotak pembatas B , dan untuk setiap kotak model menghasilkan skor kepercayaan . Skor kepercayaan ini mencerminkan seberapa yakin model yang kotak itu berisi objek. Selain itu untuk skor kepercayaan C model menghasilkan 4 angka ((x, y), w, h) untuk mewakili lokasi dan dimensi dari kotak terikat yang diprediksi. Koordinat (x, y) mewakili pusat kotak relatif terhadap batas sel kisi . Lebar dan tinggi diprediksi relatif terhadap keseluruhan gambar , jadi  $0 < (x, y, w, h) < 1$ .



Gambar 2.5. 13 menentukan box dalam grid

Untuk setiap sel jaringan, model akan menghasilkan 20 probabilitas kelas bersyarat satu untuk setiap kelas.



Gambar 2.5. 14 probabilitas kelas bersyarat

## • YOLO V2

YOLO membuat sejumlah besar kesalahan lokalisasi. Selain itu, YOLO memiliki daya ingat yang relatif rendah. Dengan demikian dalam versi kedua YOLO mereka berfokus terutama pada peningkatan daya ingat dan lokalisasi sambil mempertahankan akurasi klasifikasi. Untuk mencapai kinerja yang lebih baik mereka menggunakan beberapa ide:

1. BatchNormalisasi : Dengan menambahkan normalisasi batch pada semua lapisan konvolusional di YOLO, mereka mendapatkan lebih dari 2% peningkatan dalam pemetaan.
2. Tinggi Klasifikasi Resolusi: YOLO melatih jaringan klasifikasi untuk meningkatkan resolusi untuk deteksi .

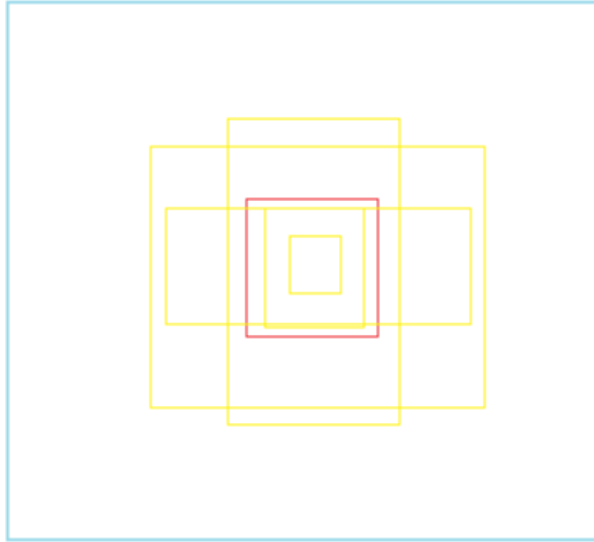
YOLOv2 mereka awalnya melatih model pada gambar, kemudian mereka menyesuaikan jaringan klasifikasi. Ini memberi jaringan waktu untuk menyesuaikan filternya agar bekerja lebih baik pada input resolusi yang lebih tinggi. Jaringan klasifikasi resolusi tinggi ini memberi mereka peningkatan hampir 4% mAP.

3. Konvolusional Dengan Kotak Jangkar(anchor box) ( prediksi multi-objek per sel kotak) :

YOLO memprediksi koordinat kotak pembatas secara langsung menggunakan lapisan yang terhubung sepenuhnya di atas fitur ekstraktor konvolusional. Kotak jangkar adalah lebar dan tinggi, yang kita dapat memprediksi kotak pembatas relatif terhadapnya daripada memprediksi kotak relatif terhadap semua gambar. Menggunakan ide ini, akan lebih mudah bagi jaringan untuk belajar. Menggunakan hanya



lapisan convolutional (tanpa lapisan yang sepenuhnya terhubung) Lebih cepat R-CNN memprediksi offset dan kepercayaan untuk kotak jangkar. Dalam gambar dibawah memiliki sel kotak (merah) dan 5 kotak jangkar (kuning) dengan bentuk yang berbeda.

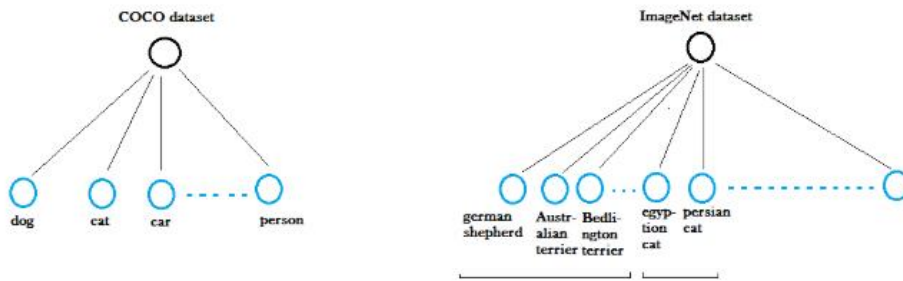


*Gambar 2.5. 15 contoh anchor box*

YOLOv2 mencoba menggunakan ide kotak jangkar untuk menemukan bentuk kotak jangkar terbaik untuk membuatnya lebih mudah bagi jaringan untuk mempelajari deteksi.

## • YOLO 9000

Beberapa kali kita memerlukan model yang dapat mendeteksi lebih dari 20 kelas, dan itulah yang dilakukan YOLO9000. Ini adalah kerangka kerja waktu nyata untuk mendeteksi lebih dari 9000 kategori objek dengan secara bersama-sama mengoptimalkan deteksi dan klasifikasi.

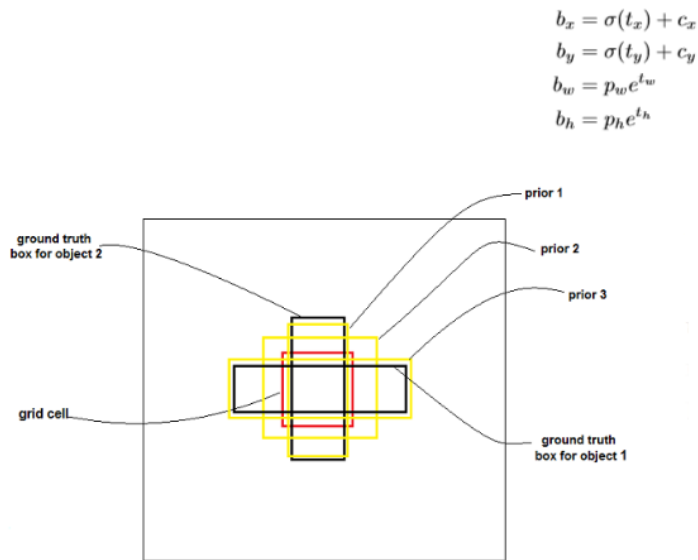


Gambar 2.5. 16 contoh klasifikasi

Model ini mengeluarkan softmax untuk setiap tingkat cabang. Kami memilih simpul dengan probabilitas tertinggi (jika lebih tinggi dari nilai ambang) saat kami bergerak dari atas ke bawah. Prediksi akan menjadi simpul tempat kita berhenti. Melakukan klasifikasi dengan cara ini juga memiliki beberapa manfaat. Kinerja menurun dengan anggun pada kategori objek baru atau tidak dikenal. Misalnya, jika jaringan melihat gambar seekor anjing tetapi tidak yakin jenis anjingnya, ia akan berhenti pada anjing dengan kepercayaan diri yang tinggi dan hasilnya adalah (anjing ). YOLO9000 berjuang untuk memodelkan beberapa kategori seperti "kacamata hitam" atau "celana renang"

### • YOLO V3

Sama seperti YOLO9000 jaringan memprediksi 4 koordinat untuk setiap kotak pembatas, tx, ty, tw, th. Jika sel diimbangi dari sudut kiri atas gambar dengan (cx, cy) dan kotak pembatas sebelumnya memiliki lebar dan tinggi Pw, Ph, maka prediksi sesuai dengan



*Gambar 2.5. 17 prediksi yolov3*

YOLOv3 juga memprediksi skor objektivitas (kepercayaan) untuk setiap kotak pembatas menggunakan regresi logistik. Ini harus 1 jika kotak pembatas sebelumnya tumpang tindih dengan objek ground truth lebih dari kotak pembatas lainnya sebelumnya. Misalnya (sebelum 1) tumpang tindih objek kebenaran tanah pertama dengan lebih dari kotak ikatan lainnya sebelumnya (memiliki IOU tertinggi) dan sebelum 2 tumpang tindih objek kebenaran tanah kedua oleh lebih dari kotak ikatan lainnya sebelumnya. Sistem hanya menetapkan satu ikatan kotak sebelum untuk setiap objek kebenaran tanah. Jika kotak sebelum melompat tidak ditugaskan ke objek kebenaran tanah itu tidak menimbulkan kerugian untuk koordinat atau prediksi kelas, hanya objek.

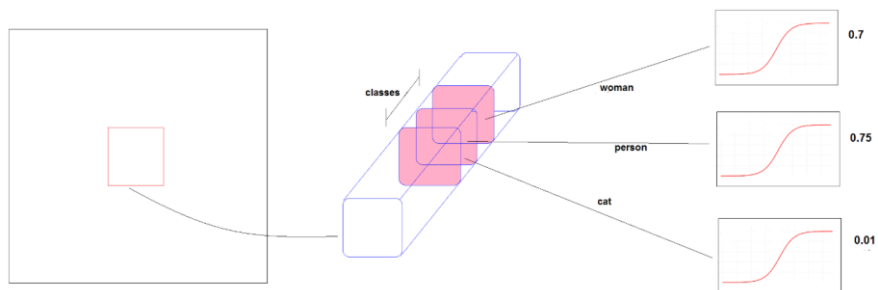
Jika kotak itu memang memiliki IOU tertinggi tetapi tumpang tindih dengan objek kebenaran tanah lebih dari beberapa ambang batas, kita mengabaikan prediksi (Mereka menggunakan ambang batas 0,5).

#### Predeksi multi label:

Dataset suatu objek mungkin memiliki multilabel, misalnya objek dapat dilabeli sebagai wanita dan sebagai orang.

Dalam dataset ini ada banyak label yang tumpang tindih. Menggunakan softmax untuk prediksi kelas memaksakan asumsi bahwa setiap kotak memiliki tepat satu kelas yang sering tidak demikian Pendekatan multilabel memodelkan data dengan lebih baik. Untuk alasan ini YOLOv3 tidak menggunakan softmax, melainkan hanya menggunakan klasifikasi logistik independen untuk kelas apa pun. Selama pelatihan itu menggunakan kerugian lintas-entropi biner untuk prediksi kelas.

Menggunakan klasifikasi logistik independen suatu objek dapat dideteksi sebagai wanita dan sebagai manusia pada saat yang bersamaan.



*Gambar 2.5. 18 klasifikasi logistik independen*

#### Deteksi benda kecil:

YOLO berjuang dengan benda-benda kecil. Namun, dengan YOLOv3 kami melihat kinerja yang lebih baik untuk objek kecil, dan itu karena

menggunakan **koneksi** pintasan. Menggunakan metode koneksi ini memungkinkan kami untuk mendapatkan informasi yang lebih berbutir dari peta fitur sebelumnya. Namun dibandingkan dengan versi sebelumnya, YOLOv3 memiliki kinerja yang lebih buruk pada objek ukuran sedang dan lebih besar.

Untuk windows Anda juga dapat menggunakan darkflow yang merupakan implementasi dari darknet tensorflow, tetapi Darkflow belum menawarkan implementasi untuk YOLOv3.

## **2.6 PENGANTAR SSD (Single Shot Detector)**

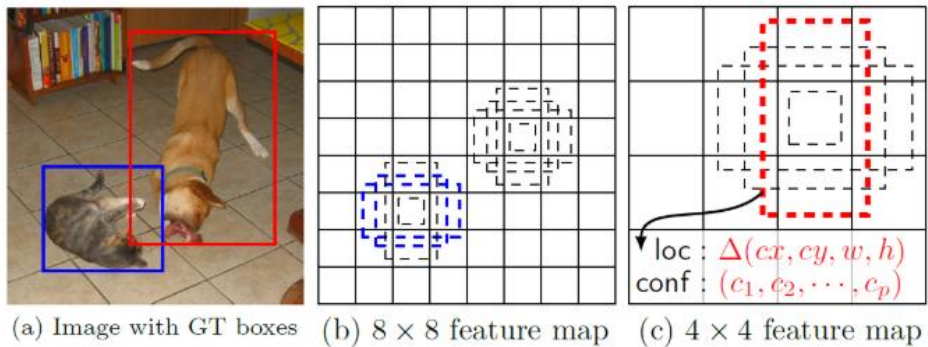
SSD adalah detektor sekali pakai. Ia tidak memiliki jaringan proposal wilayah yang didelegasikan dan memprediksi kotak batas dan kelas langsung dari peta fitur dalam satu pass tunggal.

SSD hanya membutuhkan gambar input dan kebenaran kotak untuk setiap objek selama pelatihan. Untuk setiap kotak default, kita memprediksi kedua bentuk offset dan kepercayaan untuk semua kategori objek ((C1, C2, ..., CP)).

Dengan menggunakan SSD, kita hanya perlu mengambil satu bidikan tunggal untuk mendeteksi beberapa objek di dalam gambar, sedangkan pendekatan jaringan proposal regional (RPN) seperti seri R-CNN yang membutuhkan dua bidikan, satu untuk menghasilkan proposal wilayah, satu untuk mendeteksi objek dari setiap proposal. Dengan demikian, SSD jauh lebih cepat dibandingkan dengan pendekatan dua-shot berbasis RPN.

## Detektor MultiBox

Setelah melalui konvolusi tertentu untuk ekstraksi fitur, kami memperoleh lapisan fitur ukuran  $m \times n$  (jumlah lokasi) dengan saluran  $p$ , seperti  $8 \times 8$  atau  $4 \times 4$  di atas. Dan konv  $3 \times 3$  diterapkan pada layer fitur  $m \times n \times p$  ini.



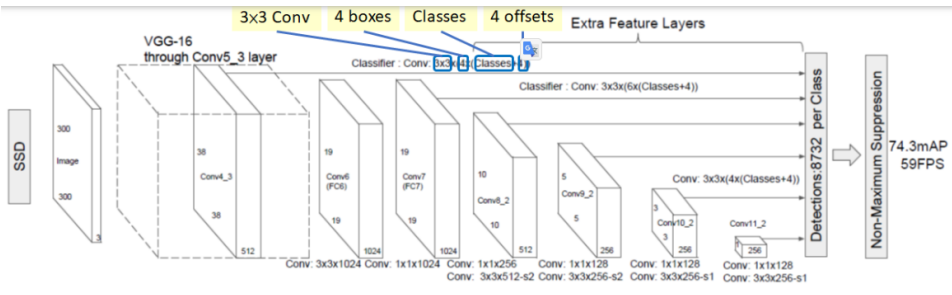
*Gambar 2.6. 1 Beberapa Kotak Batas untuk Pelokalan (loc) dan Keyakinan (conf)*

Untuk setiap lokasi, kami mendapat  $k$  kotak pembatas. Kotak pembatas  $k$  ini memiliki ukuran dan rasio aspek yang berbeda. Konsepnya adalah, mungkin persegi panjang vertikal lebih cocok untuk manusia, dan persegi panjang horizontal lebih cocok untuk mobil.

Untuk setiap kotak pembatas, kami akan menghitung  $c$  skor kelas dan 4 offset relatif terhadap bentuk kotak pembatas default asli.

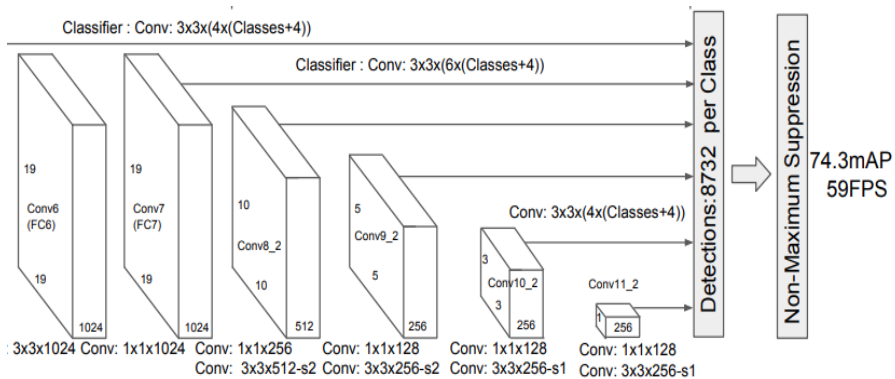
Jadi, kami mendapat  $(c + 4) kmn$  output.

## Arsitektur Jaringan SSD



Gambar 2.6. 2 arsitektur SSD

SSD menggunakan **penindasan non-maksimum** untuk menghapus prediksi duplikat yang menunjuk ke objek yang sama. SSD memilah prediksi berdasarkan skor kepercayaan. Mulai dari prediksi keyakinan teratas, SSD mengevaluasi apakah kotak batas yang diprediksi sebelumnya memiliki IoU lebih tinggi dari 0,45 dengan prediksi saat ini untuk kelas yang sama. Jika ditemukan, prediksi saat ini akan diabaikan. Paling-paling, kami menyimpan 200 prediksi teratas per gambar.



Gambar 2.6. 3 ssd non-maximum suppression

SSD berkinerja lebih buruk daripada F-R-CNN lebih cepat untuk objek skala kecil. Dalam SSD, objek kecil hanya dapat dideteksi pada lapisan dengan resolusi lebih tinggi (lapisan paling kiri). Tetapi lapisan-lapisan itu mengandung fitur tingkat rendah, seperti tepi atau tambalan warna, yang kurang informatif untuk klasifikasi.

Akurasi meningkat dengan jumlah kotak batas default dengan mengorbankan kecepatan.

Peta fitur multi-skala meningkatkan deteksi objek pada skala yang berbeda.

Desain kotak batas default yang lebih baik akan membantu akurasi.

Untuk meningkatkan akurasi, gunakan kotak standar yang lebih kecil. SSD memiliki kesalahan lokalisasi yang lebih rendah dibandingkan dengan R-CNN tetapi lebih banyak kesalahan klasifikasi berkaitan dengan kategori yang sama. Kesalahan klasifikasi yang lebih tinggi kemungkinan karena kami menggunakan kotak batas yang sama untuk membuat prediksi beberapa kelas. Untuk meningkatkan akurasi, SSD memperkenalkan:

- filter konvolusional kecil untuk memprediksi kelas objek dan offset ke kotak batas default.
- filter terpisah untuk kotak standar untuk menangani perbedaan dalam rasio aspek.
- peta fitur multi-skala untuk deteksi objek.

SSD dapat dilatih secara end-to-end untuk akurasi yang lebih baik. SSD membuat prediksi lebih banyak dan memiliki cakupan yang lebih baik pada lokasi, skala, dan rasio aspek. Dengan



peningkatan di atas, SSD dapat menurunkan resolusi gambar input ke  $300 \times 300$  dengan kinerja akurasi komparatif. Dengan menghapus proposal kawasan yang didelegasikan dan menggunakan gambar beresolusi lebih rendah, model ini dapat berjalan pada kecepatan waktu nyata dan masih mengalahkan ketepatan R-CNN yang lebih canggih.

## **BAB III**

---

## **BAB IV**

---

## **BAB V**

---

