BAB II
KAJIAN PUSTAKA

2.1 Penelitian Terkait

Banyak penelitian mengenai deteksi objek. Terdapat beberapa pendekatan metode yang dipakai untuk mendeteksi objek. Penelitian pendeteksian objek terbagi ke dalam dua jenis pendekatan yaitu metode deteksi berdasarkan pendekatan konvensional dan *deep learning*.

Pada tahun 2016, terdapat salah satu penelitian yang meneliti pengenalan objek dengan studi kasus deteksi penyakit kulit sederhana menggunakan *Scale Invariant Feature Transform* (SIFT). SIFT merupakan algoritma deteksi objek dengan pendekatan konvensional. Selain itu terdapat algoritma dengan pendekatan konvensional yang lain seperti *Speed up Robust Feature* (SURF) dan *Oriented FAST and Rotated BRIEF* (ORB). Ketiga algoritma diatas menggunakan teknik *matching*. Algoritma tersebut dinilai merupakan algoritma yang memiliki kinerja yang bagus dan efisien dalam pengenalan objek. Selain itu, terdapat pendekatan lain yang sedang banyak diteliti oleh para peneliti yaitu pendekatan *deep learning*.

Salah satu metode *deep learning* yang banyak digunakan yaitu metode CNN. Pada penelitian mengenai pendeteksian pejalan kaki yang menggunakan metode CNN, peneliti menggunakan metode tersebut untuk mengotomatisasi proses ekstraksi fitur dari gambar yang terdapat pada CCTV. Peneliti tersebut mendapatkan hasil yang cukup memuaskan dengan nilai akurasi yang mencapai 71,13% (Acharya et al., 2017).

Terdapat penelitian yang merupakan inovasi baru dari metode *Convolutional Neural Network* (CNN). Metode ini mengombinasikan wilayah deteksi dengan CNN yaitu *Region-based Convolutional Neural Network* (R-CNN). Model ini dikembangkan oleh Girshick et al., pada jurnal berjudul *rich feature hierarchies for accurate object detection and semantic segmentation*. Girshick et al., (2014) mengusulkan algoritma deteksi sederhana yang menggabungkan dua metode yaitu wilayah proposal dan fitur CNN. Penelitian ini menggunakan set data PASCAL VOC 2010. Algoritma ini meningkatkan presisi rata-rata (mAP) jika dibandingkan dengan hasil terbaik sebelumnya yang menggunakan set data VOC 2012. Dari hasil

penelitian tersebut didapatkan mAP sebesar 53,7%, lebih tinggi dibandingkan dengan wilayah proposal yang sama yang menggunakan *spatial pyramid* dan pendekatan *bag-of-visual-words* dengan mAP sebesar 35,1%. Selain itu, RCNN juga mengungguli OverFeat (Sermanet et al., 2014) pada set data yang besar seperti ILSVRC2013. RCNN memperoleh nilai mAP 31,4%, lebih besar 7,1% daripada menggunakan OverFeat.

Terdapat penelitian lain yang menggunakan *selective* search dan RPN, di mana peneliti tersebut melakukan penelitian mengenai deteksi objek yang dilakukan pada gambar manga. Hasil yang didapatkan pada penelitian yang dilakukan yaitu, metode RPN dapat mendeteksi gambar yang lebih ambigu pada gambar karakter maupun teks dibandingkan dengan *selective search* (Yanagisawa et al., 2018). Selain itu, terdapat penelitian pendeteksian teks dengan menggunakan RPN. Penelitian tersebut menggunakan dua tahap pendekatan yaitu proposal wilayah dengan metode RPN dan klasifikasi wilayah dengan RCNN. Pendekatan yang digunakan dapat mencapai hasil yang kompetitif pada set data ICDAR2015 dan ICDAR2013 (Jiang et al., 2017).

Penelitian lain mengenai deteksi objek pada video rekaman CCTV yaitu pendeteksian manusia. Penelitian dilakukan dengan mengambil informasi dari karakteristik wanita dan pria yang terdapat pada video pengawasan CCTV. Pada penelitian tersebut, digunakan metode *Faster* RCNN untuk mendeteksi. Setelah proses pendeteksian, dilakukan perbandingan menggunakan *Euclidean distance* dan Siamese. Hasil dari penelitian tersebut, didapati bahwa perhitungan yang dilakukan dengan *Euclidean distance* memberikan hasil yang menjanjikan sebagai metode asosiasi objek dalam menemukan kesamaan antara dua hal (Chahyati et al., 2017).

Pada tahun 2016, terdapat penelitian terkait menggunakan metode Siamese untuk pelacakan objek (Bertinetto et al., 2016). Bertinetto mengatakan jaringan Siamese biasanya menangani permasalahan pembelajaran kesamaan (*similarity learning*) menggunakan jaringan konvolusi. Siamese dilatih untuk menemukan gambar berdasarkan contoh atau referensi dalam gambar penelusuran yang lebih besar. Pada penelitiannya, Bertinetto, dkk menggunakan algoritma pelacakan dasar dengan jaringan Siamese konvolusi sepenuhnya (*fully convolutional)* yang dilatih

secara *end-to-end* pada set data ILSVRC15 untuk deteksi objek dalam video. Menurutnya, meskipun sederhana, algoritma yang diajukan Bertinetto, dkk dapat bekerja secara *real-time*. Selain itu, Siamese juga diterapkan pada beberapa penelitian lainnya diantaranya mempelajari matriks kesamaan antara dua kalimat (Shi et al., 2020), melakukan verifikasi tanda tangan yang ditulis pada tablet menggunakan pena tablet (Bromley et al., 1993). Selain itu, terdapat penelitian mengenai pembuatan desain penggabungan antara objek referensi dengan metode SVM yang bernama *exemplar*-SVM untuk mendeteksi objek pada set data PASCAL VOC 2007 (Malisiewicz, Gupta, & Efros, 2011). Penelitian ini menyajikan metode yang sederhana namun kuat. Kinerja dari penelitian ini dikatakan setara dengan metode yang paling baik untuk deteksi objek tetapi memiliki keunggulan antara deteksi dengan *training* objek referensi.

2.2 Kecerdasan Buatan

Kecerdasan buatan merupakan studi mengenai bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan manusia (Sri Kusumadewi, 2003). Sedangkan pengertian lain mengenai kecerdasan buatan (*artificial intelligence*) yaitu penelitian, aplikasi dan instruksi yang berkaitan dengan pemrograman komputer untuk melakukan sesuatu yang cerdas (Simon, 1995). Dengan adanya kecerdasan buatan, sangat membantu dalam memecahkan permasalahan dalam kehidupan bermasyarakat.

Pada awal diciptakannya, komputer hanya digunakan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, peran komputer melebihi daripada sebagai alat hitung. Komputer diharapkan dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia. Manusia dapat dengan cepat menyelesaikan masalah-masalah yang muncul karena manusia memiliki pengetahuan dan pengalaman yang dapat membantu dalam memecahkan masalahnya. Supaya komputer dapat mengerjakan sesuatu seperti dan sebaik manusia, maka komputer harus diberi bekal pengetahuan dan mempunyai kemampuan untuk menalar agar dapat mendapatkan pengalaman seperti layaknya manusia.

Kecerdasan buatan berbeda dengan program konvensional. Program konvensional berbasis pada algoritma yang mendefinisikan setiap langkah dalam

penyelesaian sebuah masalah. Selain itu juga dapat menggunakan rumus matematika untuk menghasilkan solusi. Sedangkan program kecerdasan buatan, sebuah simbol dapat berupa kalimat, kata atau angka yang digunakan untuk merepresentasikan objek hingga proses. Objek tersebut dapat berupa manusia, ide, benda, kegiatan, konsep atau pernyataan dari sebuah fakta. Proses digunakan untuk memanipulasi simbol sehingga menghasilkan sebuah pemecahan masalah.

Tujuan kecerdasan buatan yaitu untuk membuat komputer lebih cerdas, mengerti mengenai kecerdasan dan membuat mesin yang lebih berguna. Yang dimaksud dengan kecerdasan adalah kemampuan untuk belajar dan mengerti dari pengalaman, memahami dan menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah serta menyelesaikannya dengan efektif.

Teknik yang digunakan dalam kecerdasan buatan dapat memungkinkan pembuatan sebuah program yang setiap bagiannya berisi langkah penyelesaian masalah dan dapat mengidentifikasi dengan baik untuk memecahkan sejumlah persoalan. Sepotong informasi dalam pikiran manusia dapat digambarkan ke dalam setiap potongan bagian program. Jika informasi diabaikan, dapat secara otomatis mengatur cara kerjanya untuk menyesuaikan diri dengan fakta atau informasi baru. Tidak perlu selalu mengingat setiap potong informasi yang telah dipelajari, cukup informasi yang berkaitan dengan persoalan yang dihadapi dan yang digunakan. Setiap bagian program kecerdasan buatan dapat dimodifikasi tanpa mempengaruhi struktur seluruh programnya. Hal tersebut dapat menghasilkan program yang efisien dan mudah untuk dipahami (Wijaya & Farqi, 2015). Terdapat 2 basis utama yang dibutuhkan untuk aplikasi kecerdasan buatan (Krose & Smagt, 1996):

1) Basis pengetahuan (*knowledge base*): berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lain.
2) Motor inferensi (*inference engine*): kemampuan menarik kesimpulan berdasarkan pengalaman.

   Kecerdasan buatan memiliki ruang lingkup atau bidang. Adapun 7 ruang lingkup utama kecerdasan buatan (Nasri, 2014):

a. Sistem pakar (*expert system*). Komputer digunakan sebagai sarana untuk menampung pengetahuan para pakar. Komputer akan memiliki keahlian untuk

berpikir dalam pengambilan keputusan untuk menyelesaikan permasalahan dengan meniru keahlian yang dimiliki oleh pakar atau tenaga ahli dalam bidang yang bersangkutan. Biasanya program sistem pakar dibuat berdasarkan analisis informasi mengenai suatu masalah spesifik serta analisis matematis dari masalah tersebut.

b. Pengolahan bahasa alami (*Natural Language Processing*). *User* dapat berkomunikasi dengan komputer menggunakan bahasa sehari-hari seperti bahasa Indonesia, bahasa inggris dan lain-lain. Tujuannya yaitu melakukan proses pembuatan model komputasi dari Bahasa sehingga dapat terjadi suatu interaksi antara manusia dengan komputer dengan perantara bahasa alami manusia.
c. Pengenalan ucapan (*Speech Recognition*). *User* dapat berkomunikasi dengan komputer menggunakan suara. Hal ini memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan.
d. *Computer vision*. Menduplikasi kemampuan penglihatan manusia ke dalam benda elektronik sehingga benda tersebut dapat memahami dan mengerti arti dari sebuah gambar (Prabowo & Abdullah, 2018).
e. Robotika dan sistem sensor
f. *Intelligence computer-aided instruction* (ICAI). Komputer dapat digunakan sebagai pengajar yang dapat melatih dan mengajarkan sesuatu.
g. *Game playing*

2.3 *Computer Vision*

Computer vision sebagai salah satu disiplin ilmu yaitu berkaitan dengan teori di balik sistem kecerdasan buatan yang dapat mengekstraksi informasi dari gambar. Data gambar dapat berupa banyak bentuk seperti urutan video, sudut pandang dari beberapa kamera hingga data multidimensi dari *scanner* medis (Prabowo & Abdullah, 2018). Sebagian besar dalam tugas *computer vision* terkait dengan proses memperoleh informasi mengenai peristiwa atau sebuah deskripsi berdasarkan gambar digital dan ekstraksi fitur.

Computer vision merupakan kombinasi dari pemrosesan gambar dan pengenalan pola. *Computer vision* memiliki kombinasi konsep, teknik, ide dari

pengolahan citra digital, pengenalan pola, kecerdasan buatan dan grafik komputer (Cosido, 2015). Keluaran dari *computer vision* yaitu pemahaman citra. Pengembangan bidang ini terinspirasi dari penglihatan manusia dalam mengambil sebuah informasi (Wiley & Lucas, 2018).

Menurut Matiacevich,dkk, *Computer vision* bekerja menggunakan algoritma untuk mensimulasikan visualisasi objek untuk secara otomatis mendapatkan informasi berharga dari suatu objek tersebut (Matiacevich, Cofré, Silva, Enrione, & Osorio, 2013). Untuk mendapatkan beberapa informasi yang spesifik, biasanya pada metode *computer vision* memerlukan pemrosesan data yang bertujuan untuk memastikan data yang akan dipakai memenuhi asumsi tertentu yang berhubungan dengan metode tersebut. Misalnya seperti re-sampling, pengurangan *noise*, perbaikan kontras, representasi skala-ruang dan lain-lain (Gautama & Hendrik, n.d.).

C*omputer vision* juga dapat didefinisikan sebagai suatu disiplin teknologi, *computer vision* menerapkan teori dan model untuk membangun sistem *computer vision*. Contohnya seperti deteksi objek, pengenalan objek, pelacakan video, pemulihan citra dan lain-lain (Prabowo & Abdullah, 2018).

Computer vision merupakan ilmu pengolahan citra untuk membuat keputusan berdasarkan citra yang didapatkan dari sensor. Ilmu ini dapat membuat komputer melihat selayaknya mata manusia. Untuk membuat komputer dapat melakukan penglihatan seperti penglihatan manusia, komputer memerlukan sensor yang dapat berfungsi layaknya mata pada manusia dan sebuah program yang dapat melakukan pemrosesan data dari sensor tersebut. Dengan kata lain, *computer vision* bertujuan untuk membangun sebuah sistem pandai yang dapat "melihat". Cara kerja umum yang biasa dilakukan *computer vision* adalah akuisisi citra, ekstraksi fitur, deteksi / segmentasi pemrosesan tingkat tinggi dan pengambilan keputusan (Szeliski, 2010).

2.4 Video Rekaman CCTV

Menurut Asror,dkk, *Close Circuit Television* (CCTV) merupakan sebuah perangkat kamera video digital yang digunakan untuk mengirim sinyal ke layar monitor di suatu tempat tertentu. Hal tersebut digunakan untuk memantau situasi dan kondisi di tempat tertentu. Teknologi CCTV digunakan untuk mengawasi area

publik dan biasanya digunakan untuk keamanan. Pada saat ini, teknologi CCTV dapat diakses melalui komputer maupun telepon pintar (Asror & Siradj, 2016).

Umumnya CCTV dibagi menjadi 2 jenis yaitu analog dan digital. Sistem CCTV analog menggunakan kabel coaxial. CCTV jenis ini membutuhkan DVR, yang mana beberapa DVR mampu terhubung hingga 32 kamera. DVR memungkinkan penggunaan IP tetapi untuk tiap perangkat DVR. Sedangkan Sistem CCTV *digital* menggunakan kabel LAN. Selain itu, pada tiap kamera terdapat IP nya masing-masing (Asror & Siradj, 2016).

CCTV (*Close Circuit Television*) adalah teknologi yang sering dipakai sebagai alat keamanan fisik yang paling banyak digunakan. CCTV merupakan sebuah perangkat untuk mengumpulkan video yang dipasang di lokasi tertentu dan digunakan untuk berbagai keperluan (Muthusenthil & Kim, 2018).

Dalam konteks keamanan, CCTV berfokus pada masalah umum seperti pada pengawasan publik atau untuk tujuan tertentu seperti mendeteksi hingga mengidentifikasi. Sistem CCTV memiliki banyak peran dan fungsi berbeda dengan konsensus terbatas. Pengawasan CCTV menjadi teknologi umum yang dapat ditemukan di berbagai macam tempat seperti tempat umum, lingkungan sosial, hingga tempat kerja. Saat ini, pengawasan CCTV berarti pengamatan atau pemantauan seseorang atau apapun yang biasanya dapat diakses di berbagai macam elektronik (Brooks, 2016).

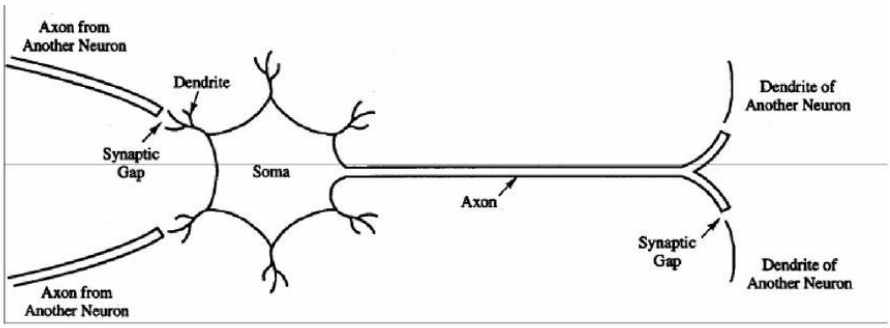
Menurut Sagala,dkk, salah satu penggunaan CCTV adalah untuk membantu penyelidikan dalam kasus tindak pidana. Dalam kasus tertentu data video rekaman CCTV disimpan untuk keperluan di masa mendatang (Sagala, Candradewi, & Harjoko, 2020).

Beberapa fungsi lain yang dimiliki CCTV selain fungsi keamanan yaitu seperti penggunaan CCTV untuk memantau kondisi kemacetan pada jalan raya yang ditempatkan di titik-titik persimpangan. Kegunaan lainnya yaitu ditempatkan didalam mobil untuk memantau kejadian di jalan raya. Selain itu ada juga beberapa CCTV yang digunakan untuk memantau keadaan kegiatan belajar mengajar yang ditempatkan di masing-masing ruang kelas (Asror & Siradj, 2016).

2.5 Jaringan Syaraf Tiruan

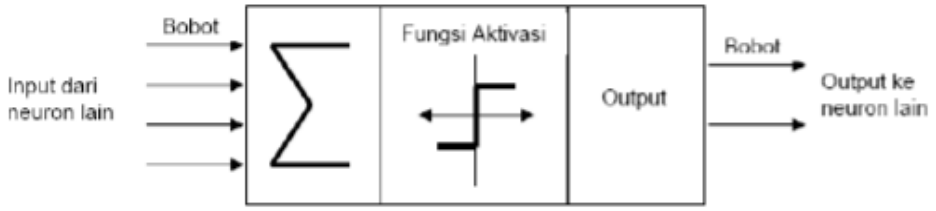
Jaringan Syaraf Tiruan (JST) merupakan suatu sistem pemrosesan informasi yang mempunyai karakteristik menyerupai jaringan syaraf biologis seperti pada Gambar 1 (Sudarsono, 2016). JST tercipta sebagai suatu generalisasi model sistematis dari pemahaman manusia yang didasari asumsi-asumsi sebagai berikut (Wuryandari & Afrianto, 2012):

1. Pemrosesan informasi terjadi pada elemen sederhana yang disebut neuron.
2. Sinyal yang mengalir antara sel neuron melalui suatu sambungan penghubung.
3. Setiap sambungan penghubung memiliki bobot yang bersesuaian. Bobot ini digunakan untuk menggandakan sinyal yang dikirim melaluinya.
4. Setiap sel saraf akan menerapkan fungsi aktivasi terhadap sinyal hasil penjumlahan berbobot yang masuk untuk menentukan sinyal keluarannya.



Gambar 1. Syaraf secara biologis (Salim & Jauhari, 2016)

Struktur dari sistem pengolahan informasi yang terdiri dari sejumlah besar elemen pemrosesan yang saling berhubungan (*neuron*), bekerja serentak untuk menyelesaikan masalah tertentu. Cara kerja jaringan syaraf tiruan yaitu sama seperti cara kerja manusia (belajar dari contoh) (Salim & Jauhari, 2016).



Gambar 2. Model struktur JST (Wuryandari & Afrianto, 2012)

Jaringan syaraf tiruan dapat belajar dari pengalaman, melakukan generalisasi berdasarkan contoh yang didapatnya dan mengekstraksi karakteristik esensial masukan bahkan untuk data yang tidak relevan. Algoritma ini beroperasi secara langsung dengan angka sehingga data yang tidak numerik harus diubah menjadi data numerik terlebih dahulu. JST tidak diprogram untuk menghasilkan keluaran tertentu. Semua keluaran yang diambil oleh jaringan, didasari berdasarkan pada pengalamannya. Pada proses pembelajaran, ke dalam JST dimasukkan pola-pola masukkan (dan keluaran) lalu jaringan akan diajari untuk memberikan jawaban yang dapat diterima seperti pada Gambar 2. Pada dasarnya karakteristik JST ditentukan oleh (Wuryandari & Afrianto, 2012):

1. Pola hubungan antar neuron yang disebut arsitektur jaringan.
2. Metode penentuan bobot-bobot sambungan yang disebut pelatihan atau proses belajar jaringan.
3. Fungsi aktivasi.

A. Arsitektur Jaringan Syaraf Tiruan

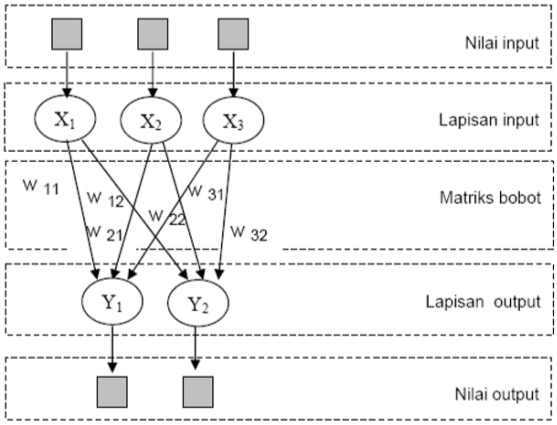
Pada jaringan syaraf tiruan, neuron akan dikumpulkan dalam lapisan-lapisan (*layer*) yang disebut dengan lapisan neuron (neuron *layers*). Neuron-neuron pada satu lapisan akan dihubungkan dengan lapisan sebelum dan sesudahnya. Informasi yang diberikan pada jaringan syaraf akan dirambatkan lapisan ke lapisan, mulai dari lapisan masukan sampai lapisan keluaran melalui lapisan tersembunyi (*hidden layer*).

Terdapat faktor penting dalam menentukan sifat suatu neuron yaitu penggunaan fungsi aktivasi dan pola bobotnya dari neuron tersebut. Umumnya neuron yang terletak pada lapisan yang sama akan memiliki keadaan yang sama sehingga pada setiap lapisan yang sama neuron-neuron akan memiliki fungsi

aktivasi yang sama. Bila neuron pada suatu lapisan akan dihubungkan dengan neuron pada lapisan lain maka setiap neuron pada lapisan tersebut juga harus dihubungkan dengan setiap neuron pada lapisan lainnya. Terdapat tiga macam arsitektur jaringan syaraf tiruan berdasarkan jumlah lapisannya (Wuryandari & Afrianto, 2012):

1. Jaringan dengan lapisan tunggal (*single layer net*)

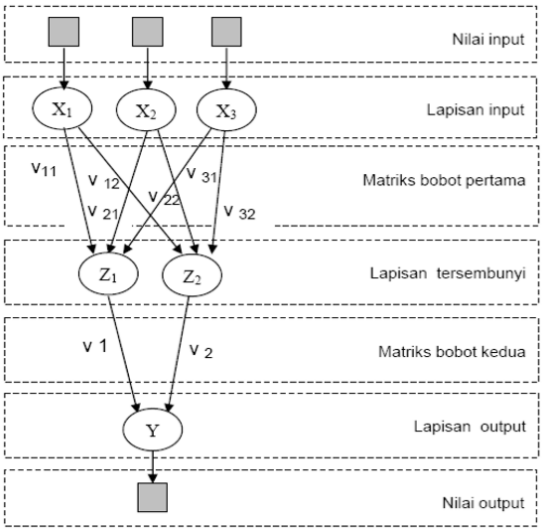
Jaringan ini hanya memiliki satu lapisan dengan bobot-bobot terhubung. Jaringan ini hanya menerima masukan kemudian secara langsung akan mengolahnya menjadi keluaran tanpa harus melalui *hidden layers* seperti pada Gambar 3. Jaringan ini terdiri dari satu lapisan masukan dan satu jaringan keluaran.



Gambar 3. Struktur *single layer net*
(Wuryandari & Afrianto, 2012)

2. Jaringan dengan banyak lapisan (*multilayer net*)

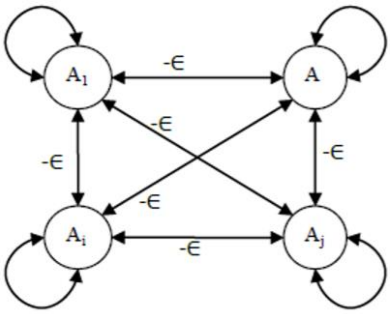
Jaringan ini memiliki satu atau lebih lapisan yang terletak diantara lapisan masukan dan lapisan keluaran. Jaringan ini juga memiliki lapisan tersembunyi (*hidden layer*). Umumnya terdapat lapisan bobot-bobot yang terletak antara dua lapisan yang bersebelahan seperti Gambar 4. Jaringan dengan banyak lapisan ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan lapisan tunggal. Pembelajaran pada jaringan dengan banyak lapisan ini lebih sukses dalam menyelesaikan masalah tetapi proses pelatihan lebih membutuhkan waktu yang cenderung lama.



Gambar 4. Struktur *multilayer net*
(Wuryandari & Afrianto, 2012)

3. Jaringan dengan lapisan kompetitif (*competitive layer net*)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Berikut contoh struktur yang menggunakan jaringan kompetitif yang terlihat pada Gambar 5.



Gambar 5. Struktur *competitive layer net*
(Wuryandari & Afrianto, 2012)

2.6 Deteksi Objek

Deteksi objek merupakan pekerjaan yang penting dalam *computer vision* dalam mendeteksi sebuah objek visual dari kelas tertentu dalam gambar digital. Hal ini bertujuan untuk mengembangkan model dan teknik komputasi yang menyediakan salah satu dari sebagian besar informasi dasar yang dibutuhkan (Zou, Shi, Guo, & Ye, 2019).

Deteksi objek merupakan suatu pendekatan *computer vision* yang memiliki proses yang cukup kompleks untuk dilakukan. Hal ini berguna untuk mengenali bagian objek yang diinginkan dengan akurat. Proses deteksi objek akan mengolah hasil dari proses segmentasi sehingga dapat diketahui seperti banyaknya objek yang terdeteksi, luas area dan titik pusat tiap objek (Budi Putranto, Hapsari, & Wijana, 2011).

Untuk mengidentifikasi objek dalam sebuah video dan untuk mengelompokkan piksel dari objek-objek tersebut disebut juga deteksi objek. Objek yang terdeteksi dapat diklasifikasikan dalam berbagai kategori seperti manusia, kendaraan, burung, pohon, dan benda lainnya. Deteksi objek dapat dilakukan dengan berbagai teknik seperti perbedaan frame, *optical flow*, dan *background substraction* (Parekh, Thakore, & Jaliya, 2014).

Deteksi objek yang kuat dan cepat merupakan tantangan besar dalam bidang *computer vision*. Deteksi yang kuat dan cepat memiliki dua fitur utama. Fitur tersebut yaitu eksekusi paralel hibrida dan metode skala gambar. Eksekusi paralel hibrida mengeksploitasi penggolongan struktur *cascade*, sedangkan untuk pengklasifikasian yang terletak pada *cascade* lebih sering digunakan daripada pengklasifikasian berikutnya (Saubari, Ansari, & Gazali, 2019).

Menurut Mushawwir, dkk, deteksi objek bertujuan untuk memisahkan objek atau *foreground* dari citra latar. Untuk melakukan deteksi objek, biasanya dibuat model latar pada sebuah citra bergerak berdasarkan nilai pixel yang masuk seiring waktu (Piccardi, 2004). Metode atau teknik untuk mendapatkan nilai pixel tersebut dapat dikelompokkan menjadi *temporal frame difference, background substraction* dan metode statistik (Mushawwir & Supriana, 2015).

Deteksi objek banyak digunakan dalam berbagai hal. Contohnya pada pendeteksian keberadaan objek lubang dalam sebuah citra (Yusuf Budiarto & Sutikno, 2017). Salah satu penerapan konsep deteksi objek lainnya yaitu pada pembuatan rumah pintar untuk menerapkan konsep deteksi objek untuk manusia (Robby Yuli Endra, Cucus, Affandi, & Syahputra, 2013). Selain itu juga pada sistem robot dan sistem pengawasan pun sering diterapkan pendeteksian objek.

Deteksi objek dapat memproses gambar bagian demi bagian untuk secara otomatis mendeteksi sebuah wilayah dengan konten visual tertentu dengan

mengandalkan klasifikasi gambar. Contohnya pada citra udara dan satelit. Pada citra udara, deteksi objek memiliki tugas yang penting. Pemrosesan citra yang dilakukan cepat dan akurat merupakan hal yang sangat penting karena teknologi tersebut dapat memberikan sejumlah besar citra udara dan satelit dengan resolusi tinggi (Sevo & Avramovic, 2016).

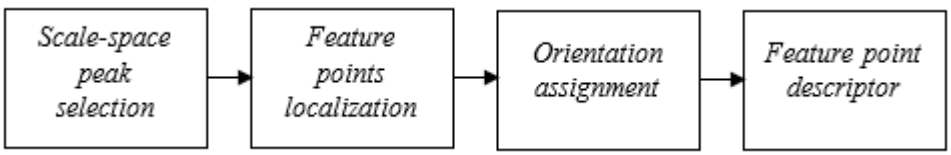
Adapun beberapa tantangan utama dalam mendeteksi sebuah objek, diantaranya:

1) Oklusi atau penumpukan sebuah objek sehingga objek hanya terlihat sebagian (Fulari, 2018; Mushawwir & Supriana, 2015).
2) Pencahayaan yang buruk, perbedaan intensitas cahaya hingga keadaan yang dapat mengganggu saat mendeteksi seperti hujan, malam hari, sore hari dan lain-lain (Buch, Velastin, & Orwell, 2011; Fulari, 2018) .
3) Terdapat bayangan pada objek sehingga dapat mengganggu pendeteksian (Mushawwir & Supriana, 2015).
4) Sudut pandang kamera yang membuat ukuran dan dimensi objek menjadi tidak pasti (He, Wang, & Zhang, 2011).

Untuk mendeteksi sebuah objek diperlukan teknik atau metode. Metode yang digunakan dibagi menjadi dua yaitu metode konvensional dan *deep learning*. Berikut beberapa metode konvensional untuk mendeteksi sebuah objek:

1) *Scale-Invariant Feature Transform* (SIFT)

SIFT termasuk ke dalam metode konvensional. SIFT merupakan algoritma deteksi feature pada *computer vision* untuk deteksi dan deksripsi fitur lokal pada gambar. Metode ini invariant terhadap skala, rotasi gambar, pencahayaan, dan sudut pandang kamera 3D. Fitur yang yang didapat memungkinkan fitur tersebut untuk dicocokkan dengan tepat sehingga menghasilkan probabilitas yang tinggi. Secara umum, algoritma SIFT terdiri dari empat tahap yang terlihat pada Gambar 6.

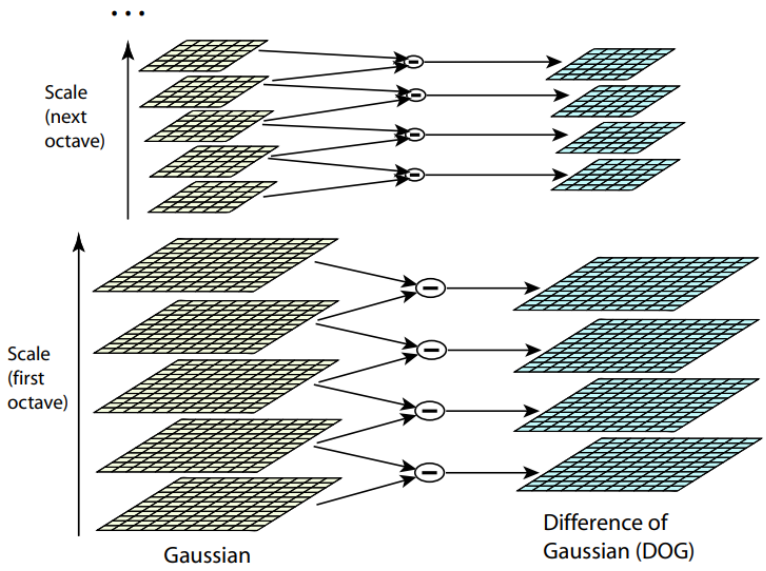


Gambar 6. Tahapan algoritma SIFT

a. *Scale-space peak selection*

Tahap pertama yaitu *scale-space peak selection.* Tahap ini mengidentifikasi semua lokasi dan skala gambar. Hal tersebut diimplementasikan secara efisien menggunakan perbedaan fungsi Gaussian untuk mengidentifikasi *interest points* yang invarian terhadap skala dan orientasi. Mendeteksi lokasi yang berbeda dengan perubahan skala gambar dapat dicapai dengan mencari feature stabil di semua kemungkinan skala, dengan menggunakan fungsi skala yang berkelanjutan yang dikenal sebagai *scale-space* (Lowe, 2004).

Gambar asli diperbesar dua kali, kemudian dengan dua skala pengambilan sampel hingga ukuran tertentu 64 x 64. Setelah itu dibuat piramida Gaussian. Puncak lokal pada *scale-space* dalam serangkaian *Difference of Gaussian* (DoG) yang dicari disebut juga oktaf. Pencarian puncak local pada *scale-space* terlihat pada Gambar **7**. Kemudian pilih *keypoints* pada setiap oktaf sebagai kandidat untuk menjadi *interest point* dari skala dan orientasi yang tidak berubah-ubah.



Gambar 7. Pencarian puncak lokal pada *scale-space* (Lowe, 2004)

Scale-space didefinisikan sebagai fungsi *L (x, y, σ)*, yang menghasilkan konvolusi skala dari variabel Gaussian *G (x, y, σ)*, dengan inputan gambar *I (x, y)*. Standar deviasi distribusi normal Gaussian adalah σ. Lokasi titik point stabil dalam skala dapat dihitung dari selisih Gaussian $D(x, y, \sigma)$, yang dipisahkan oleh faktor multiplikasi konstanta k.

b. *Feature points localization*

Pada setiap lokasi kandidat, model yang didapat cocok untuk menentukan lokasi dan skala. *Keypoints* dipilih berdasarkan ukuran kestabilannya. Penggunaan fungsi kuadrat 3D secara akurat dapat menentukan lokasi dan skala *keypoints* serta menghilangkan *keypoints* dengan kontras yang rendah dan *edge response points* yang tidak stabil untuk meningkatkan stabilitas pencocokan dan meningkatkan kemampuan anti-*noise* (Lu, Xu, Dai, & Zheng, 2012).

c. *Orientation assignment*

Satu atau lebih orientasi ditugaskan untuk setiap lokasi *keypoints* berdasarkan arah gradien gambar lokal. Semua operasi selanjutnya akan dilakukan pada data gambar yang telah ditransformasikan relative terhadap orientasi, skala, dan lokasi yang ditetapkan untuk setiap feature sehingga memberikan variasi pada transformasi ini (Lowe, 2004).

d. *Feature point descriptor*

Gradien gambar local akan diukur pada skala yang dipilih di wilayah sekitar masing-masing *keypoints*. Selanjutnya descriptor akan dinormalisasi untuk meningkatkan stabilitasnya terhadap perubahan proyeksi dan iluminasi (Lu et al., 2012).

2) *Histogram of Oriented Gradient* (HOG)

Histogram of Oriented Gradient termasuk ke dalam metode konvensional. HOG digunakan untuk mengekstraksi fitur pada objek gambar dengan menggunakan objek manusia (Robby Yuli Endra et al., 2013). HOG merupakan deskriptor wilayah citra berdasarkan gradien citra local. Secara khusus, informasi tampilan suatu objek ditangkap menggunakan HOG. HOG invarian terhadap transformasi geometrik dan fotometri serta tidak berubah terhadap perubahan iluminasi dan translasi objek. Selain itu, fitur HOG invarian terhadap rotasi objek (He et al., 2011).

Proses awal metode ini adalah citra RGB (*Red, Green, Blue*) dikonversi menjadi *grayscale*. Langkah selanjutnya, kemudian dengan menghitung nilai gradien pada setiap piksel. Setelah mendapatkan nilai gradien tersebut, maka proses selanjutnya yaitu menentukan jumlah bin orientasi yang akan digunakan dalam

pembuatan *histogram* yang disebut *spatial orientation binning* (Robby Yuli Endra et al., 2013).

Pada proses komputasi gradien, gambar pelatihan akan dibagi menjadi beberapa *cell* dan dikelompokkan menjadi ukuran yang lebih besar yang dinamakan *block*. Proses normalisasi *block* digunakan perhitungan geometri R-HOG. Proses ini dilakukan karena adanya *block* yang saling tumpang tindih. Berbeda dengan proses pembuatan *histogram* citra yang menggunakan nilai-nilai intensitas piksel dari suatu citra atau bagian tertentu dari citra untuk pembuatan *histogram* (Robby Yuli Endra et al., 2013).

Penerapan algoritma HOG untuk pelacakan dalam pengawasan video menggunakan kamera pengawasan, fokus pada aspek suatu tindakan yang tidak biasa, identifikasi orang, pengenalan aktivitas dan lain-lain. Algoritma HOG merupakan bagian dari *computer vision*. Sistem pengawasan video memainkan peran utama dalam *computer vision*. Penelitian ini terkonsentrasi pada deteksi objek manusia dan pelacakan untuk menghindari tantangan yang terlibat dalam kondisi sulit. Model yang diusulkan yaitu berdasarkan pendekatan segmentasi klaster. Masukan yang dipertimbangkan yaitu video akan dibagi menjadi beberapa *frame* diikuti dengan segmentasi kluster dan ekstraksi fitur. Ekstraksi fitur dilakukan berdasarkan *histogram of gradient*. Sedangkan klasifikasi menggunakan *support vector machine*. Setiap aktivitas objek akan dideteksi berdasarkan dengan hasil klasifikasinya. Model yang diusulkan menghitung akurasi deteksi setiap objek hingga 89,59% (Seemanthini & Manjunath, 2018).

Penelitian lain menggunakan algoritma *histogram of* gradient yaitu mengenai sistem pengenalan wajah untuk mengidentifikasi para pelaku kejahatan. Keberadaan sistem ini diharapkan dapat membantu aparat penegak hukum untuk mengidentifikasi wajah para pelaku kejahatan. Kemudian dapat tertangkap dan mengurangi angka kasus kejahatan (R. Y. Endra, Kurniawan, & Saputra, 2016).

2.7 *Deep Learning*

Deep learning adalah teknik *machine learning* yang mengajarkan komputer untuk mempelajari dan memahami sesuatu berdasarkan pengalaman atau contoh (*learn by example*) yang secara alami terjadi pada manusia. *Deep learning*

memungkinkan model komputasi untuk mempelajari data yang kompleks (L. Liu et al., 2018).

Deep learning merupakan salah satu bidang dari *machine learning* yang terdiri dari banyak lapisan (*hidden layer*) dan membentuk tumpukan. Lapisan tersebut adalah sebuah algoritma yang melakukan klasifikasi perintah yang dimasukan hingga menghasilkan sebuah keluaran (Nurfita & Ariyanto, 2018).

Deep learning merupakan cabang dari *machine learning* yang terinspirasi dari kortek manusia dengan menerapkan jaringan syaraf tiruan (JST) yang memiliki banyak *hidden layer*. *Deep learning* menjadi sorotan dalam pengembangan *machine learning*. Hal ini karena *deep learning* mencapai hasil yang luar biasa dalam *computer vision* (Santoso & Ariyanto, 2018).

Deep learning merupakan sebuah bidang keilmuan baru dalam bidang *machine learning*. *Deep learning* memiliki kemampuan yang sangat baik dalam *computer vision*. Salah satunya pada kasus klasifikasi objek pada citra. Dengan mengimplementasikan salah satu metode *machine learning* yang dapat digunakan untuk klasifikasi citra objek yaitu *Convolution Neural Network* (CNN) (Marifatul Azizah, Fadillah Umayah, & Fajar, 2018).

Deep learning mendapatkan perhatian yang banyak belakangan ini karena dapat mencapai hasil yang sebelumnya tidak memungkinkan. Sebuah kemajuan yang signifikan dalam berbagai masalah seperti deteksi objek, pengenalan objek, pengenalan bahasa, pemrosesan bahasa alami, analisis citra dan lain-lain.

Seiring cepatnya perkembangan riset mengenai *deep learning*, banyak *library* yang muncul dengan fokus mempelajari jaringan syaraf tiruan, salah satunya Keras. Keras merupakan *library* jaringan syaraf tiruan tingkat tinggi yang ditulis dengan bahasa python dan mampu berjalan di atas TensorFlow, CNTK dan Theano (Chollet, 2015). *Library* tersebut menyediakan fitur yang digunakan untuk mempermudah pengembangan lebih dalam mengenai *deep learning* (Santoso & Ariyanto, 2018).

2.8 Objek Referensi

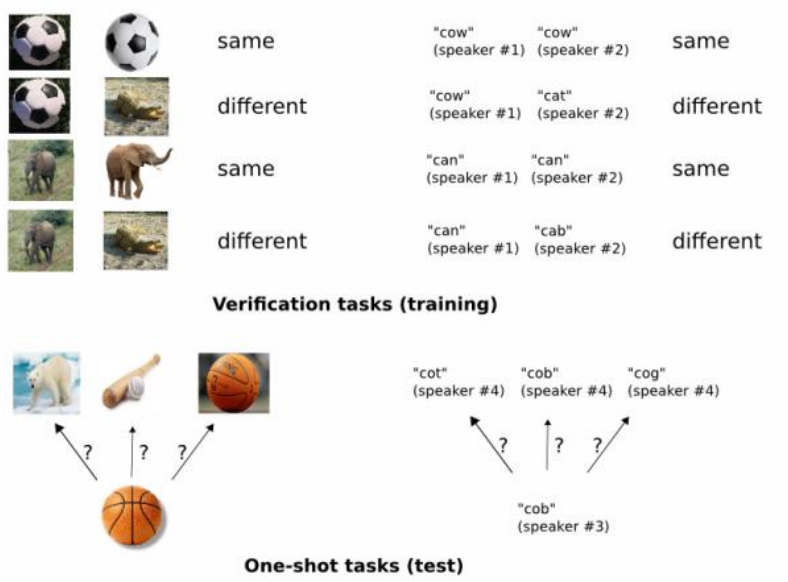
Pembelajaran objek referensi merupakan pendeteksian objek berdasarkan referensi yang ada. Pembelajaran ini merupakan paradigma yang kuat untuk menemukan kesamaan visual pada pembelajaran *unsupervised*. Bagian gambar dari

objek yang ingin dicari akan diambil untuk dijadikan referensi atau pembanding dalam melakukan pendeteksian. Dengan hanya satu sampel positif, ketidakseimbangan antara satu sampel positif dan banyak sampel negatif akan menjadi hal yang sulit dilakukan dalam pembelajaran objek referensi ini (Bautista, Sanakoyeu, Sutter, & Ommer, 2016).

Pembelajaran ini dapat disebut juga dengan *one-shot learning*, di mana diharuskan membuat prediksi dengan benar meski hanya satu contoh dari masing-masing kelasnya (Koch, Zemel, & Salakhutdinov, 2015). Fitur yang didapat dari hanya satu contoh tersebut akan diekstrak dan dicari nilai kesamaannya antara kumpulan sampel yang ada. Sampel dengan fitur yang jauh berbeda akan didistribusikan ke dalam kelompok yang berbeda dan sampel dengan fitur yang serupa akan dikelompokkan ke dalam kelompok yang sama (Bautista et al., 2016). Umumnya pelatihan *deep neural network*, akan digunakan untuk mempelajari fitur yang berguna untuk melakukan *one-shot learning*.

One-shot learning bertujuan untuk mengidentifikasi gambar yang menjadi referensi dalam semua instance objek dari kelas yang sama yang tersirat dalam gambar referensi. Kesulitan utamanya terletak pada situasi jika label kelas gambar referensi tidak tersedia dalam data pelatihan. Pada penelitian Chen, et al. untuk permasalahan objek referensi, dimanfaatkan sebuah konsep untuk meningkatkan metode deteksi berbasis pembelajaran metrik (Chen et al., 2020).

Terdapat dua tahap dalam pembelajaran *one-shot learning* yang terlihat pada Gambar 8. Tahap pertama yaitu melatih model untuk membedakan pasangan gambar yang mana akan diberi label ‘sama’ atau ‘beda’. Tahap kedua yaitu memprediksi untuk menentukan masukan tersebut merupakan citra dengan identitas yang berbeda atau sama, dengan menggunakan *one-shot learning* untuk mengidentifikasi gambar baru yang berbeda dengan gambar pada tahap pertama.



Gambar 8. *One-shot learning* (Koch et al., 2015)

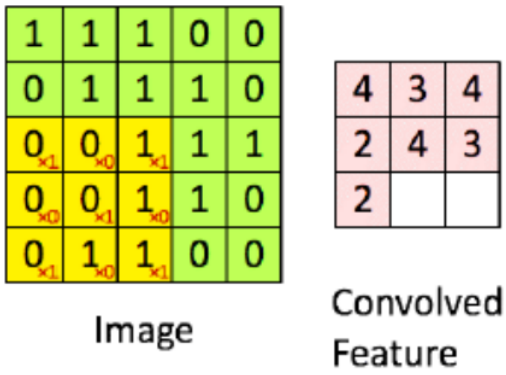
2.9 *Convolutional Neural Network*

Convolutional neural networks termasuk ke dalam metode *deep learning*, lebih tepatnya kelas *deep neural networks*. CNN telah banyak digunakan dalam permasalahan *computer vision* seperti klasifikasi dan deteksi gambar. CNN seperti *neural networks* pada umumnya hanya saja memiliki lapisan yang lebih dalam yang memiliki bobot, bias dan keluaran melalui aktivasi nonlinier (Galvez, Bandala, Dadios, Vicerra, & Maningo, 2019).

Convolutional neural networks adalah suatu *layer* yang memiliki bentuk neuron 3D yaitu lebar, tinggi dan kedalaman (*depth*). Lebar dan tinggi untuk bentuk *layer*. Sedangkan kedalaman mengacu pada jumlah *layer*. Secara umum CNN dibedakan menjadi dua jenis *layer* yaitu *layer* ekstraksi fitur gambar yang terdiri atas beberapa *layer* konvolusi dan *pooling layer*. Setiap *layer* ini memiliki fungsi aktivasi. *Layer-layer* ini menerima input gambar secara langsung dan memprosesnya menjadi sebuah vektor yang akan diproses pada *layer* berikutnya. Jenis *layer* berikutnya yaitu *layer* klasifikasi. *Layer* ini terdiri atas beberapa *layer* yang setiap *layer*-nya tersusun atas neuron yang terkoneksi sepenuhnya (*fully connected*) dengan *layer* lainnya (Zufar & Setiyono, 2016).

1) *Layer* Konvolusi

Layer ini melakukan operasi konvolusi pada output dari *layer* sebelumnya. Konvolusi merupakan istilah matematis yang berarti mengaplikasikan sebuah fungsi yang melakukan perulangan pada keluaran fungsi lain. Operasi konvolusi terlihat pada Gambar **9**. Tujuannya adalah untuk mengekstraksi fitur dari citra inputan. Konvolusi akan menghasilkan sebuah transformasi linear sesuai dengan informasi spasial yang terdapat pada data. Bobot pada *layer* menspesifikasikan *kernel* konvolusi yang digunakan sehingga kernel tersebut dapat dilatih berdasarkan input pada CNN (Suartika E. P, 2016).

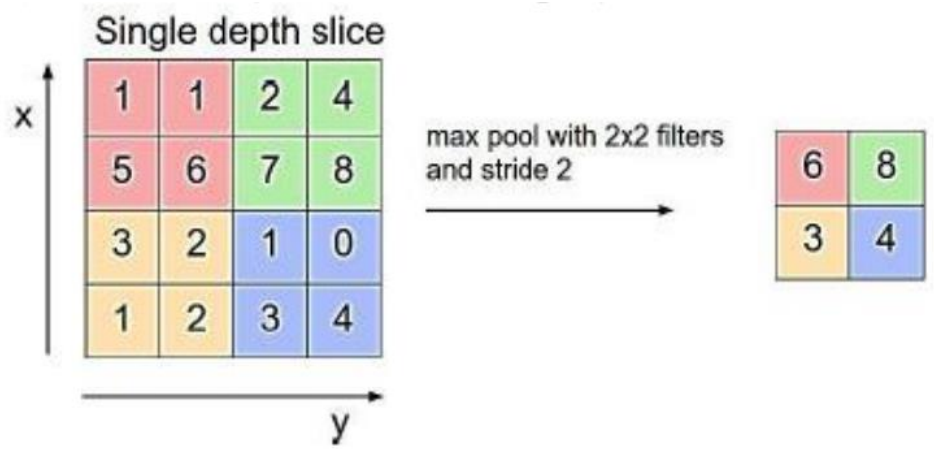


Gambar 9. Operasi konvolusi (Suartika E. P, 2016)

2) *Pooling layer*

Pooling layer atau biasa disebut juga subsampling. Subsampling merupakan proses mereduksi ukuran matriks sebuah data citra. Untuk pengolahan citra, subsampling juga memiliki tujuan untuk meningkatkan invariansi posisi dari fitur (Suartika E. P, 2016). *Layer* ini akan mereduksi jumlah parameter dan ukuran spasial pada jaringan. Selain itu, *pooling layer* juga dapat mempercepat komputasi dan dapat mengontrol terjadinya masalah *overfitting*. *Pooling layer* mempunyai beberapa macam tipe yaitu *max pooling*, *average pooling* dan Lp *pooling* (Zufar & Setiyono, 2016). Perhitungan *max pooling* terlihat pada Gambar **10**.

Menurut Springgenberg,dkk pada CNN, penggunaan *pooling layer* hanya bertujuan untuk mereduksi ukuran citra sehingga dapat digantikan dengan *convo layer* dengan *stride* yang sama dengan *pooling layer* yang bersangkutan (Springenberg, Dosovitskiy, Brox, & Riedmiller, 2015).



Gambar 10. Operasi *max pooling* (Suartika E. P, 2016)

3) *Fully connected layer*

Layer ini biasanya digunakan untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara linear. Untuk masuk kedalam *fully connected layer*, setiap neuron pada *convo layer* harus ditransformasikan menjadi data satu dimensi terlebih dahulu sehingga *layer* ini hanya dapat diimplementasikan di akhir jaringan. Hal tersebut dikarenakan dapat menyebabkan kehilangan informasi spasial yang terdapat pada data dan tidak reversible (Suartika E. P, 2016).

Keluaran yang dihasilkan pada *layer* ekstraksi fitur gambar berupa vektor, akan ditransformasikan dengan tambahan *hidden layer*. Hasil keluaran dari *layer* ini yaitu skoring kelas untuk klasifikasi (Zufar & Setiyono, 2016).

4) Fungsi aktivasi

Fungsi aktivasi atau disebut juga fungsi transfer merupakan fungsi non-linear yang memungkinkan sebuah jaringan dapat menyelesaikan permasalahan non trivial. *Layer* ini terdapat pada akhir perhitungan keluaran *feature map* atau sesudah proses konvolusi atau pooling untuk menghasilkan suatu pola fitur. Beberapa fungsi aktivasi yang sering digunakan dalam penelitian yaitu sigmoid, tanh, *Rectified Linear Unit* (ReLU), Leaky ReLU dan lain-lain (Zufar & Setiyono, 2016).

2.10 *Region Convolutional Neural Network*

Region Convolutional Neural Network termasuk ke dalam metode *deep learning*. RCNN merupakan perubahan penting dari CNN untuk deteksi objek. Model ini dikembangkan menggunakan pendekatan *sliding* window pada c*onvolutional neural network* (Abbas & Singh, 2018).

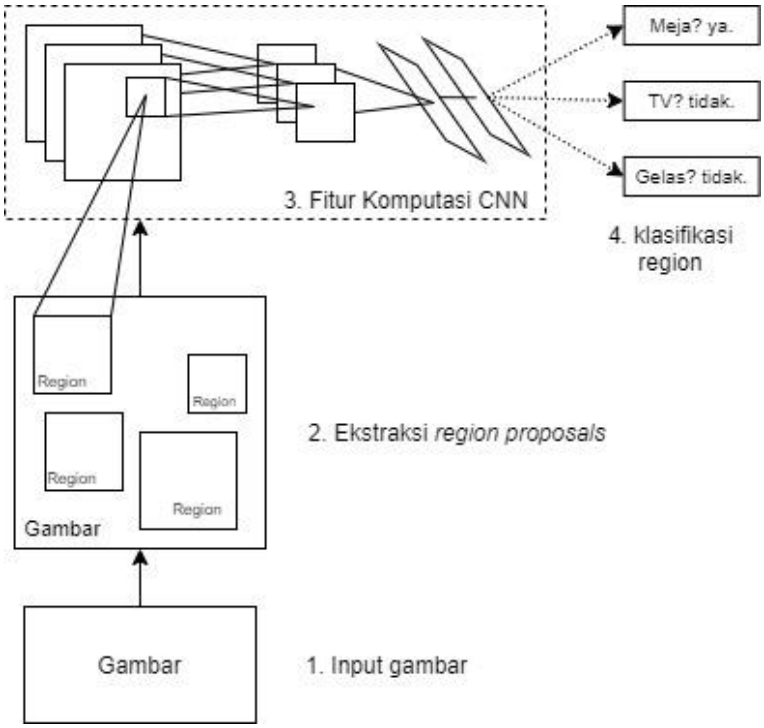
RCNN adalah jaringan saraf pertama yang mengajukan wilayah proposal untuk mendeteksi objek berdasarkan ekstraksi fitur dan klasifikasi CNN yang baik. Wilayah proposal akan memilih objek yang memiliki probabilitas tinggi untuk menjadi objek dengan menggeser proposal dengan tinggi dan lebar yang berbeda (Du, 2018).

Algoritma R-CNN mengusulkan memakai banyak kotak pada gambar dan memeriksa apakah terdapat objek pada setiap kotak. Kotak-kotak ini disebut *regions*. Untuk mendeteksi wilayah, metode ini menggunakan empat warna pengenal, warna, skala dan penutup yang bervariasi. Metode RCNN menggunakan *selective search* untuk mengekstrak kotak-kotak (*regions*) dari gambar. RCNN melakukan satu persatu ekstraksi fitur dari objek setelah itu akan dilakukan pengklasifikasian terhadap wilayah proposal (Abbas & Singh, 2018). Langkah – Langkah yang akan dilakukan pada metode RCNN adalah sebagai berikut (Wang, He, & Li, 2015):

1. Mengambil gambar sebagai input.
2. Menggunakan *selective search* atau metode wilayah proposal lain untuk mempertahankan RoI pada gambar (wilayah yang berisi objek).
3. Wilayah proposal yang telah didapat sebelumnya kemudian diteruskan ke CNN tetapi sebelum diteruskan, wilayah proposal akan diubah (*reshape*) untuk dijadikan sebagai input pada CNN. CNN akan mengekstrak fitur untuk setiap wilayah proposal.
4. Selanjutnya wilayah proposal yang didapat akan dibagi ke berbagai kelas dengan menggunakan SVM. Regresi kotak pembatas (*boundingbox*) digunakan untuk memprediksi kotak pembatas untuk setiap wilayah proposal yang diidentifikasi.

2.9.1 Arsitektur RCNN

RCNN terdiri dari tiga modul. Yang pertama menghasilkan wilayah proposal (*region proposals)* kategori-independent menggunakan *selective search*. Yang kedua yaitu *convolutional neural network* yang mengekstraksi vektor fitur dari masing-masing wilayah proposal yang telah didapatkan sebelumnya dengan ukuran panjang yang tetap. Yang ketiga yaitu SVM untuk klasifikasi objek (Girshick et al., 2014). Pada Gambar 11 terlihat aristektur RCNN.



Gambar 11. Arsitektur RCNN (Girshick et al., 2014)

a. *Region proposals*

Seperti yang telah dijelaskan pada paragraf sebelumnya, pada arsitektur RCNN menggunakan metode untuk menghasilkan wilayah proposal kategori-independen. Beberapa metode yang digunakan untuk menghasilkan wilayah proposal kategori-independen yaitu *objectness, selective search, category-independent object proposals, constrained parametric min-cuts* (CPMC), *multi-scale combinatorial grouping*. Pada RCNN metode yang digunakan adalah *selective search* (Girshick et al., 2014).

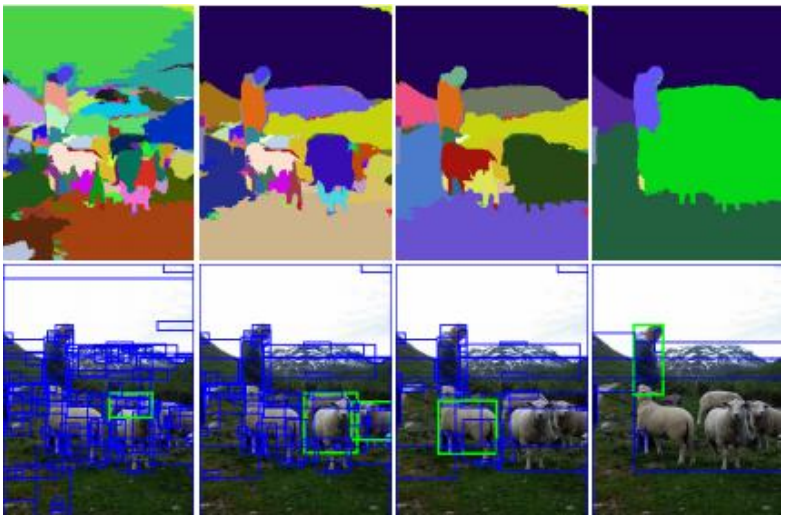
Selective search merupakan salah satu algoritma yang dapat menghasilkan wilayah proposal kategori-independen untuk masalah deteksi objek. Metode ini didasari pada komputasi pengelompokan hierarki dari kawasan atau daerah yang serupa berdasarkan warna, tekstur, ukuran dan bentuk. *Selective search* akan

mencari daftar wilayah (*region*) yang dianggap paling masuk akal yang memiliki objek di dalamnya (Chahal & Dey, 2018).

Hal pertama yang dilakukan pada metode *selective search* yaitu melakukan segmentasi berlebih pada gambar berdasarkan intensitas piksel menggunakan metode segmentasi berbasis grafik oleh Felzenszwalb dan Huttenlocher. *Selective search* menggunakan segmentasi berlebih ini sebagai masukkan awal seperti yang terdapat pada Gambar 12 (Uijlings, Van De Sande, Gevers, & Smeulders, 2013).

Gambar 12. Contoh segmentasi (Uijlings et al., 2013)

Setelah melakukan segmentasi, Langkah selanjutnya yaitu menambahkan kotak pembatas sesuai dengan bagian yang tersegmentasi ke daftar wilayah proposal. Kemudian dikelompokkan berdasarkan kesamaannya seperti pada Gambar **13**. *Selective search* dapat mengambil berbagai skala. Objek yang terdapat pada suatu gambar dapat muncul dengan berbagai skala. Oleh karena itu *selective search* perlu memperhitungkan semua skala objek yang muncul (Uijlings et al., 2013).



Gambar 13. Contoh *selective search* (Uijlings et al., 2013)

b. Fitur ekstraksi

Pada tahap ini, akan dilakukan ekstraksi fitur dengan ukuran yang tetap pada setiap wilayah proposal menggunakan CNN (Girshick, Donahue, Darrell, & Malik, 2016). Tahap ini terdiri dari serangkaian lapisan konvolusi, *max pooling* dan fungsi aktivasi. Wilayah proposal yang didapatkan sebelumnya akan menjadi masukkan pada CNN untuk menghasilkan peta fitur (*feature map*) (Chahal & Dey, 2018).

Untuk menghitung fitur wilayah proposal, diharuskan mengonversi data gambar pada wilayah proposal tersebut kedalam bentuk yang kompatibel dengan CNN. Pada arsitektur CNN dibutuhkan input yang memiliki ukuran tetap sebesar 227x227. Fitur dihitung dengan menyebarkan gambar 227x227 yang dikurangi rata-rata melalui lima lapisan konvolusi dan dua lapisan terhubung sepenuhnya (*fully connected layer*) (Girshick et al., 2014).

Setelah didapatkan peta fitur, peta fitur tersebut dimasukkan ke dalam dua lapisan terhubung sepenuhnya (*fully connected layer*). Hasil dari lapisan terhubung sepenuhnya yaitu vektor yang berukuran 4.079 dimensi. Kemudian vector tersebut akan menjadi masukkan dalam dua jaringan SVM yang terpisah yaitu untuk menjadi klasifikasi dan regresi. Klasifikasi digunakan untuk memprediksi kelas objek yang dimiliki wilayah proposal tersebut. Sedangkan regresi digunakan untuk memprediksi offset ke koordinat wilayah agar lebih sesuai dengan objek.

Terdapat dua komputasi *loss* pada RCNN. Loss klasifikasi dan loss regresi. Loss klasifikasi memakai *cross entropy loss,* sedangkan loss regresi memakai L2 atau *mean square error loss* (Chahal & Dey, 2018).

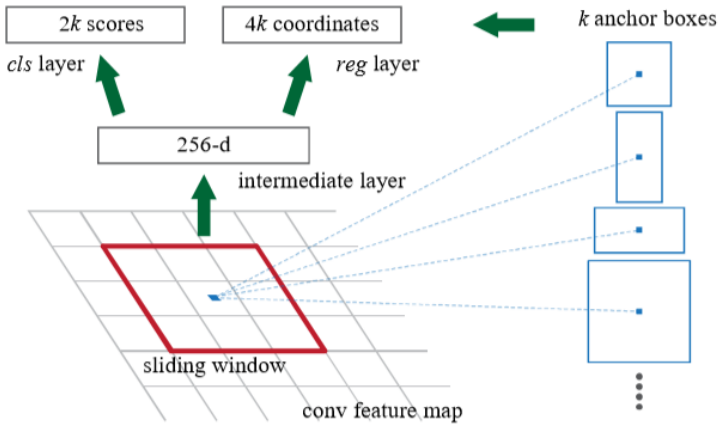
Model RCNN dilatih menggunakan dua langkah prosedur. Sebelum memulai pelatihan, akan digunakan *pretrained* ImageNet basis konvolusional. Langkah pertama yaitu melatih klasifikasi SVM menggunakan *cross entropy loss*. Bobot pada regresi tidak akan diperbaharui. Selanjutnya pada Langkah kedua, akan dilakukan pelatihan regresi menggunakan L2 *loss*. Bobot pada klasifikasi sudah ditetapkan. Proses ini akan membutuhkan waktu 84 jam karena fitur dihitung dan disimpan untuk setiap wilayah proposal. Jumlah wilayah proposal yang tinggi akan membutuhkan media penyimpanan yang besar dan operasi masukan dan keluaran menambah biaya overhead yang cukup besar (Chahal & Dey, 2018).

2.11 *Region Proposal Network*

Region proposal network atau dapat disingkat RPN dikemukakan oleh Ross Girschik pada tahun 2015 sebagai pendekatan deteksi objek menggunakan *deep learning* (Abbas & Singh, 2018). RPN merupakan jaringan konvolusional penuh (*fully connected network*) yang secara bersamaan dapat memprediksi batas objek dan skor objektivitas di setiap posisi. RPN dilatih secara menyeluruh guna menghasilkan proposal wilayah yang berkualitas tinggi. RPN dirancang agar dapat memprediksi proposal wilayah secara efisien dengan berbagai skala dan rasio (Ren, He, Girshick, & Sun, 2015).

RPN mengambil gambar dengan ukuran yang bermacam-macam dan menghasilkan sekumpulan proposal objek pada setiap posisi di peta fitur. Metode ini menggunakan jendela geser (*sliding window*) diatas peta fitur untuk menghasilkan vektor untuk setiap posisinya (X. Wu, Sahoo, & Hoi, 2020).

RPN dapat mencari kemungkinan lokasi atau wilayah proposal pada gambar yang menjadi masukan dengan cepat. Seperti yang sudah dijelaskan di paragraf sebelumnya, model ini menggunakan *sliding window* diatas peta fitur. Peta fitur akan diperoleh pada lapisan terakhir pada *convolutional layer*. *Sliding window* yang digunakan sebesar n x n seperti pada Gambar 14. Setiap jendela geser dipetakan ke fitur dimensi yang lebih rendah. Fitur tersebut dimasukkan ke dalam dua lapisan terhubung sepenuhnya (*fully connected layer*) yaitu lapisan klasifikasi (*cls*) yang akan mengklasifikasi apakah fitur tersebut objek atau bukan dan lapisan regresi kotak pembatas (*reg*). Wilayah deteksi akan ditandai dengan adanya *anchor*. Arsitektur model ini secara alami diimplementasikan dengan lapisan konvolusional n x n diikuti oleh dua lapisan konvolusional 1 x 1 masing-masing pada lapisan klasifikasi dan regresi (Ren et al., 2015).



Gambar 14. A*nchor RPN* (Ren et al., 2015)

Pada setiap lokasi *sliding window*, secara bersamaan akan dilakukan prediksi dibeberapa proposal wilayah, dimana jumlah maksimum proposal yang mungkin untuk setiap lokasi akan dilambangkan dengan *k*. Lapisan regresi memiliki keluaran 4*k* untuk koordinat *k* kotak dan lapisan klasifikasi menghasilkan keluaran skor 2*k* yang memperkirakan probabilitas objek atau bukan objek untuk setiap proposal. *Anchor* yaitu proposal *k* yang diberi parameter sesuai dengan kotak referensi *k*. *Anchor* dipusatkan pada *sliding windows* yang terdapat pada gambar 2.11 dan dikaitkan dengan skala dan aspek rasio. RPN menggunakan 3 skala dan 3 aspek rasio, yang menghasilkan *k* = 9 *anchor* pada setiap posisi *sliding window*. Untuk peta fitur konvolusi dengan ukuran lebar x tinggi, terdapat total lebar x tinggi x *k* *anchor* (Ren et al., 2015).

Untuk melatih RPN, akan ditetapkan label kelas biner (objek atau bukan) untuk setiap *anchor*. Untuk memilih *anchor*, *anchor* akan dibagi menjadi dua kategori yaitu positif dan negatif. Terdapat dua kondisi untuk menetapkan label positif yaitu jika *anchor* dengan *intersection over union* (IoU) tertinggi tumpang tindih (*overlap*) dengan kotak *ground-truth* dan jika *anchor* memiliki IoU *overlap* lebih tinggi dari 0.7 dengan kotak *ground-truth* apapun (Ren et al., 2015). Menurut (Y. Liu, 2019), aturan atau kondisi untuk melakukan klasifikasi adalah sebagai berikut:

1. Jika nilai IoU *anchor* lebih besar dari *ground truth*, maka anchor dilabeli sebagai sampel positif.

2. Jika nilai IoU *anchor* lebih besar dari 0.7, maka *anchor* dilabeli sebagai sampel positif. Biasanya kita dapat menemukan cukup sampel positif pada aturan kedua ini.

   $p* = 1$ jika IoU > 0.7
3. Jika nilai IoU *anchor* lebih rendah 0.3, maka *anchor* diberi label sebagai sampel negatif.

   $p* = -1$ jika IoU > 0.3
4. Sisa *anchor* yang bukan sampel positif maupun negatif dan tidak dipakai untuk pelatihan.

   $p* = 0$
5. Kotak *anchor* yang melintasi batas gambar dibuang

IoU digunakan untuk menghitung rasio tumpang tindih (*overlap*) pada *anchor*. Rasio tersebut yang akan menentukan *anchor* ke dalam dua kategori. Rasio tumpang tindih IoU terlihat pada Gambar **15**.

$$IoU = \frac{S_{anchorBox} \cap S_{groundTruth}}{S_{anchorBox} \cup S_{groundTruth}}$$

Gambar 15. *Intersection Over Union* (Y. Liu, 2019)

Tujuan dari RPN yaitu untuk menghasilkan kotak wilayah proposal deteksi objek yang baik. Untuk melakukannya, RPN harus belajar untuk mengklasifikasikan kotak *anchor* sebagai positif dan negatif. Kemudian menghitung koefisien regresi untuk mengubah posisi, lebar, dan tinggi kotak *anchor* positif yang lebih baik. RPN *loss function* diformulasikan sedemikian rupa untuk suatu gambar yang didefinisikan pada persamaan (1).

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i*) + \lambda \frac{1}{N_{reg}} \sum_i p_i * L_{reg}(t_i, t_i*) \quad (1)$$

Berdasarkan persamaan diatas, i adalah indeks *anchors* dan p_i adalah kemungkinan indeks *anchor i* yang diprediksi sebagai objek. Sedangkan p_i* adalah *ground truth*. t_i adalah vektor yang mewakili empat koordinat parameter dari *anchor* yang diprediksi dan t_i* adalah *anchor* dari *ground truth* yang memiliki *anchor* positif. L_{cls} adalah log dari dua kelas kategori yaitu objek dan bukan objek, didefinisikan sebagai berikut (Y. Liu, 2019).

$$L_{cls}(p_i, p_i*) = -\log\left[p_i * p_i + (1 - p_i*)\,(1 - p_i)\right] \quad (2)$$

Untuk fungsi L_{reg} didefinisikan sebagai berikut.

$$L_{reg}(t_i, t_i*) = R(t_i - t_i*) \quad (3)$$

Di mana R adalah *robust loss function* (*smooth* L1). $p_i * L_{reg}$ adalah fungsi *loss* regresi diaktifkan hanya untuk *anchor* positif ($p_i * = 1$) dan dinonaktifkan jika ($p_i * = 0$). Keluaran dari *layer* cls dan reg ini masing-masing terdiri dari $\{p_i\}$ dan $\{t_i\}$ (Ren et al., 2015).

Untuk kotak regresi diambil empat koordinat parameter yang didefinisikan pada persamaan (4) dan (5) (Abbas & Singh, 2018).

$$t = \left[\frac{x - x_\alpha}{w_\alpha}, \frac{y - y_\alpha}{h_\alpha}, \log(w/\,w_\alpha), \log(h/\,h_\alpha)\right] \quad (4)$$

$$t^* = \left[\frac{x^* - x_\alpha}{w_\alpha}, \frac{y^* - y_\alpha}{h_\alpha}, \log(w^*/\,w_\alpha), \log(h^*/\,h_\alpha)\right] \quad (5)$$

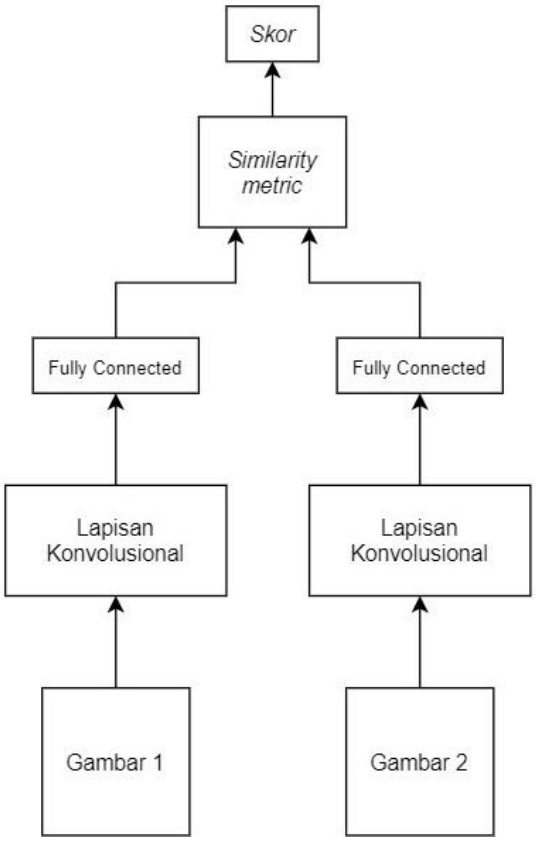
Di mana rumus diatas menunjukkan koordinat titik pusat kotak dan lebar serta tinggi secara berurut $[x, y, w, h]$. Variabel x, x_α, x^* masing-masing untuk kotak prediksi, kotak *anchor* dan kotak *ground truth* dan begitu pula untuk y, w, h.

2.12 Jaringan Siamese

Jaringan Siamese digunakan untuk mempelajari matriks kemiripan (*similarity metric*) antara dua masukan gambar. Siamese merupakan struktur jaringan berbasis pasangan (*pair-based*). Siamese dilatih untuk menemukan gambar berdasarkan contoh atau referensi dalam gambar penelusuran yang lebih besar (Bertinetto et al., 2016). Jaringan Siamese biasanya menangani permasalahan pembelajaran kesamaan (*similarity learning*) menggunakan jaringan konvolusi (Bertinetto et al., 2016).

Struktur jaringan Siamese terdiri dari dua jaringan yang identik yang berbagi bobot dan parameter. Tujuan utamanya adalah untuk mempelajari representasi fitur yang optimal dari pasangan input, yang mana jika gambar input pasangan lebih cocok dengan input contoh akan ditarik lebih dekat (memiliki nilai kesamaan yang

relatif tidak jauh berbeda) dan jika gambar input pasangan tidak cocok akan dijauhkan (Melekhov, Kannala, & Rahtu, 2016).



Gambar 16. Arsitektur *Siamese network*

Masukkan dari jaringan Siamese yaitu dua masukkan gambar. Jaringan ini memproses dua input tersebut secara terpisah melalui jaringan individual berbentuk jaringan saraf konvolusional yang identik (Tao, Gavves, & Smeulders, 2016). Jaringan yang identik yang berbagi bobot dan parameter yang terlihat pada Gambar **16**. Setiap jaringan memiliki jaringan saraf dalam yang mencakup satu set lapisan konvolusional, *rectified linear units* (ReLU) dan lapisan terhubung sepenuhnya (*fully connected layer*). Keluaran dari setiap jaringan saraf konvolusional tersebut adalah representasi dari gambar. Kemudian akan digunakan matriks kesamaan untuk membandingkan keluaran dari kedua struktur jaringan saraf konvolusional. Kesamaan yang dihitung merupakan hasil akhir dari arsitektur *siamese* (Shi et al., 2020).