**BAB II**

**TINJAUAN PUSTAKA**

1. **Pengertian Citra**

Citra (*image*) adalah salah satu komponen multimedia yang memegang peranan yang sangat penting sebagai bentuk informasi visual. Citra sebagai kumpulan dari titik (*pixel*) yang mempunyai identitas tertentu untuk membentuk satu k esatuan perpaduan yang mempunyai arti, baik secara “artistik” maupun “intrinsik”. Citra yang dapat menampilkan keindahan gambar (aristik) dan kejelasan gambar untuk penganalisaan dan maksud-maksud lainnya (intrinsik) adalah karakteristik citra yang baik (Munir, 2004).

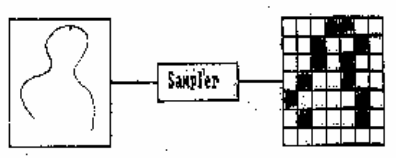
*Pixel*  menunjukkan satuan dari sebuah citra. *Pixel* merupakan singkatan dari *picture element*, yaitu unsur gambar atau representasi sebuah titik terkecil dalam sebuah gambar grafis yang dihitung per-inci. Setiap *pixel* mewakili tidak hanya satu titik dalam sebuah citra melainkan sebuah bagian berupa kotak yang merupakan bagian terkecil yang biasa disebut sel (Putra, 2010).

Ada dua macam jenis citra yaitu analog dan citra digital. Citra *analog* adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar x, hasil CT *Scan* dan lain-lain. Sedangkan pada citra *digital* adalah citra yang dapat diolah oleh computer (Sutoyo, et al., 2009).

1. **Citra Analog**

Citra *analog*  adalah citra yang bersifat *continue,* seperti citra pada monitor, telelvisim foto sinar X, foto yang tercetak dikertas foto, lukisan, hasil *CT scan,* citra - citra yang terekam pada pita kaset, dan lain sebagainya. Citra *analog* tidak dapat direpresentasikan dalam computer, sehingga tidak bias diproses di komputer secara langsung (Munir, 2004).

Oleh sebab itu , agar bias diproses di komputer, proses konversi analog ke digital harus dilakukan terlebih dahulu. . Citra analog dihasilkan dari alat-alat analog, video kamera analog, kamera foto analog, Web Cam, CT *scan*, sensor *ultrasound* pada sistem USG, dan lain-lain (Sutoyo, et al., 2009).



Gambar 2.1 Operasi erosi

(Sutoyo, et al., 2009)

1. **Citra Digital**

Secara matematis, sebuah citra dapat didefinisikan sebagai fungsi dua dimensi *f(x,y),* dimana *x* dan *y* adalah koordinat spasial *(plane)* dan *f(x,y)* adalah nilai intensitas warna pada koordinat *x* dan *y*. Nilai *f, x* dan *y* semuanya adalah nilai berhingga. Citra digital umumnya dua dimensi (2D) yang dinyatakan dalam bentuk matriks dengan jumlah elemen berhingga. Setiap elemen matriks citra memiliki posisi koordinat *x* dan *y* tertentu dan juga memiliki nilai (Madenda, 2015).

Secara umum citra digital merupakan representasi *pixel-pixel* dalam ruang 2D yang dinyatakan dalam *matrix* berukuran N baris dan M kolom. Setiap elemen *matrix* citra disebut *pixel* *(picture element, image element atau pel)*. Nilai setiap *pixel* *f* pada posisi koordinat *x* dan *y* merepresentasikan intensitas warna dan dapat dikodekan dalam 24 bit untuk citra berwarna (dengan tiga komponen warna RGB: R = *red*, G = *green*, dan B = *blue*), 8 bit untuk citra gray-level atau 1 bit untuk citra biner (Madenda, 2015). Pada umumnya citra digital berbentuk empat persegi panjang dan dimensi ukurannya dinyatakan sebagai tinggi x lebar (Munir, 2004). Citra digital yang tingginya N, lebarnya M dan memiliki L derajat keabuan dapat dianggap sebagai fungsi dapat dilihat pada rumus (1) (Munir, 2004) :

(1)

Citra digital yang berukuran N x M lazim dinyatakan dengan *matrix* yang berukuran N baris dan M kolom. Rumus (2) memperlihatkan representasi citra digital ke dalam *matrix* dua dimensi.

(2)

1. **Elemen-elemen Citra Digital**

Di dalam citra digital terdapat elemen-elemen sebagai berikut (Putra, 2010) :

1. Kecerahan (*brightness*)

Kecerahan merupakan intensitas cahaya yang dipancarkan pada sebuah titik (*pixel*) melalui citra yang ditangkap oleh indera penglihatan. Sedangkan intensitas rata-rata dari area yang melingkupinya meliputi kecerahan pada sebuah titik di dalam sebuah citra.

1. Kontras (*contrast*)

Kontras merupakan penyebaran warna terang (*lightness*) dan warna gelap (*darkness*) pada suatu citra. Citra dengan kontras rendah didirikan oleg sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dalam kontra yang baik, komposisi gelap dan terang tersebar secara merata.

1. Kontur (*countour*)

Kontur merupakan proses terjadinya perubahan intensitas cahaya pada *pixel*-*pixel* yang bertetangga. Karena adanya perubahan intensitas maka kita mampu mendeteksi tepi-tepi (*edge*) objek di dalam citra.Warna (*colour*)

1. Warna (colour)

Warna merupakan presepsi panjang gelombang cahaya yang dipantulkan oleh objek dan ditangkap oleh system penglihatan.

1. Bentuk (*shape*)

Bentuk merupakan property intrinsic utama dari objek tiga dimensi yang digunakan untuk *system* penglihatan manusia.

1. Tekstur (*texture*)

Tekstur merupakan keteraturan pola-pola tertentu yang berbentuk dari susunan piksel-piksel didalam citra digital yang dicirikan sebagai distribusi spasial dari derajat keabuan didalm sekumpulan piksel-piksel yang berdekatan sehingga dapat digunakan untk membedakan sifat-sifat spesifik dari kekasaran maupun kehalusan permukaan objek didalam suatu citra.

1. **Format File Citra**

Format *file* menentukan bagaimana informasi data dipresesnntasikan dalam suatu file. Informasi tersebut meliputi ada tidaknya kompresi, program aplikasi (*feature*) yang didukung (*support*), penggunaan enkripsi, dan lain-lain. Tiap format *file* memiliki kelebihan dan kelemahan masing-masing. Dalam sistem operasi *Windows*, format *file* dapat dibedakan dari namanya yaitu diakhiri titik dan diikuti tiga atau empat huruf terakhir sebagai penanda format. Format *file* citra ada yang terkompresi dengan sifat *lossless* dan ada yang bersifat *lossy*. ***Lossless*** adalah jenis kompresi data citra atau data lainnya yang tidak satu pun mengalami perubahan atau kehilangan data/informasi saat dilakukan rekonstruksi, atau dengan kata lain bahwa data hasil rekonstruksi adalah tepat sama dengan data aslinya. ***Lossy*** adalah jenis kompresi data citra atau data lainnya yang dapat mengalami perubahan atau kehilangan data/informasi saat dilakukan rekonstruksi, atau dengan kata lain bahwa data hasil rekonstruksi adalah tidak selalu tepat sama dengan data aslinya. Untuk *file* citra (*image*), format yang umum digunakan adalah (Madenda, 2015) :

1. BMP (*Windows bitmap*)

BMP (*Windows bitmap*) merupakan format *file* grafik dari sistem operasi windows, umumnya *file* BMP yang terkompresi dengan sifat *lossless* dan menggunakan algoritma RLE (*Run-Length Encoding*) dimana dapat menyimpan citra biner, *grayscale,* citra berwarna *true colour,*  dan citra berwarna *indexed* yang digunakan dalam program *Windows*. Pada umumnya, format *file* ini yang berukuran besar tidak dikomporesi.

1. TIF (*Tagged Image Format*)

TIF (*Tagged Image Format*) merupakan format penyimpanan citra yang terkompresi maupun tidak terkompresi seperti citra bitmap hingga berwarna.

1. GIF (*Graphics Interchange Format*)

GIF (*Graphics Interchange Format*) merupakan format *file* bitmap (grafik) yang didasarkan pada jenis data raster 2D dengan berbagai resolusi. Format GIF memiliki kelebihan yaitu dapat menyimpan citra sekuensial (beberapa citra) dalam satu *file* sehingga dapat memberikan efek animasi saat ditampilkan pada layar komputer.

1. PNG (*Portable Network Graphics*)

PNG (*Portable Network Graphics*) merupakan format *file* citra bitmap yang menggunakan algortima kompresi *lossless* untuk menyimpan data citra. PNG merupakan format *file* citra *open source* (dapat digunakan tanpa lisensi hak cipta) dan diciptakan untuk memperbaiki kekurangan format GIF. Format PNG dapat menyimpan citra berwarna dalam ruang warna RGB dengan 24 bit per-*pixel* atau ruang warna RGBA (A adalah *alpha channel*) dengan 32 bit per *pixel*, dan citra *grayscale* baik dengan atau tanpa *alpa channel*.

1. JPEG (*Joint Photographic Experts Group*)

JPEG (*Joint Photographic Experts Group*) merupakan format *file* citra yang digunakan untuk menyimpan dan mengirimkan gambar foto di *World Wide Web*. Namun, nama ini sering tidak digunakan dan tetap hanya disebut JPEG. Format JPEG menggunakan algoritma kompresi yang bersifat *lossy*, dimana kualitas informasi citra sangat bergantung pada banyaknya data atau informasi yang hilang atau berubah. Rasio kompresi JPEG relatif besar, dengan kualitas citra rekonstruksi yang tetap baik.

1. JPEG 2000

JPEG 2000 merupakan format kompresi citra dan sistem pengkodean standar yang dibuat oleh *Joint Photographic Experts Group* pada tahun 2000. Dengan maksud menggantikan metode *Discrete Cosine Transform* (DCT) yang digunakan pada standar JPEG dengan metode terbaru *Discrete Wavelet Transform* (DWT) *multiscale* atau multi-resolusi. Format ini memiliki ekstensi nama *file* standar .jp2 untuk ISO/IEC 15444-1 dan .jpx untuk ISO/IEC 15444-2. Format kompresi JPEG 2000 memiliki rasio dan kualitas kompresi yang lebih baik dibandingkan dengan format kompresi JPEG. Keuntungan utama JPEG 2000 adalah *codestream* (*bitstream*) hasil kompresinya memiliki fleksibilitas resolusi yang signifikan. *Codestream* ini dapat dipenggal-penggal, dimana setiap penggalannya menghasilkan resolusi atau ukuran citra yang berbeda-beda. Namun, format ini memiliki kelemahan dibandingkan JPEG yaitu algoritma kompresi dan dekompresi (*encoder & decoder*) yang lebih kompleks dan waktu komputasinya pun lebih lama.

1. **Jenis-Jenis Citra**

Untuk menyimpan citra dan foto digunakan format citra layar *kuadratis* (berbentuk kotak) yang terdiri atas titik-titik citra kecil yang disebut dengan *pixel. Pixel*  berbentuk bujur sangkar dengan ukuran relatif kecil. Banyaknya *Pixel* tiap satuan luas tergantung pada resolusi yang digunakan (Putra, 2010).

Suatu pixel memiliki nilai rentang tertentu, dari nilai minimum hingga nilai maksimum. Jangkauan 0-255 yang digunakan berbeda-beda tergantung dari jenis warna dari citra tersebut (Madenda, 2015). Beberapa jenis citra yang umum digunakan dalam pemrosesan citra yaitu (Kadir, et al., 2013) :

1. Citra Warna (RGB)

Citra warna (RGB) adalah jenis citra yang memilki 3 buah *channel* warna didalamnya yaitu *channel red, channel green, channel blue.* Setiap *channel* memiliki 8 bit (nilai dari 0-255) yang menghasilkan kemungkinan warna 16.777.216 warna (Kadir, et al., 2013).



Gambar 2.2 citra warna

(Hidayatullah, 2017)

1. Citra Grayscale (Skala Keabuan)

Citra *grayscale*  adalah citra yang memiliki piksel nilai *luminance* 8 bit (28 -1) dimana bervariasi dari nilai 0 sebagai warna hitam sampai 255 sebagai warna putih, dan nilai diantaranya mempresentasikan warna keabuan yang bervariasi dan warna gelap (hitam) menuju warna cerah (putih) (Madenda, 2015).



Gambar 2.3 citra grayscale

(Hidayatullah, 2017)

1. Citra Biner

Citra biner atau citra hitam putih (*black and white image*) adalah citra yang memiliki *pixel* nilai *luminance* 1 bit dimana nilai untuk setiap *pixel-*nya hanya ada 2 kemungkinan nilai yaitu 0 atau 1. Nilai 0 sebagai warna hitam sedangkan nilai 1 sebagai warna putih. Jenis citra ini sering digunakan untuk segmentasi citra (Hidayatullah, 2017). Citra biner diperoleh berdasarkan nilai ambang (*Threshold*) dimana jika nilai *pixel*  lebih kecil daripada nilai *threshold*  maka nilai *pixel* diubah menjadi 0 (hitam) dan sebaliknya jika nilai *pixel* lebih besar atau sama dengan nilai *threshold* maka nilai *pixel*  diubah menjadi 1 (Putih) (Madenda, 2015).

1. **Pengolahan Citra**

Banyaknya informasi yang tersirat di dalam sebuah citra tidak akan terhindar dari penurunan kualitas (degredasi) diantaranya terdapat derau (*noise*), warna yang terlalu kontras maupun kurang tajam, efek warna yang tidak merata mengakibatkan suatu citra sulit untuk dikenali dikarenakan adanya *blurring* (kabur), dan lain sebagainya. Oleh sebab itu, perlu dilakukan pengolahan citra sebagai proses untuk memanipulasi citra yang terdegrasi menjadi kualitas yang lebih baik. Pemrosesan citra dilakukan, dengan maksud mendapatkan kualitas citra yang diinginkan, terdapat berbagai operasi pengolahan citra yang dapat diklasifikasikan dalam berbagai jenis. Operasi-operasi yang dilakukan di dalam pengolahan citra banyak ragamnya. Namun, secara umum operasi pengolahan citra dapat diklasifikasikan dalam beberapa jenis sebagai berikut (Munir, 2004):

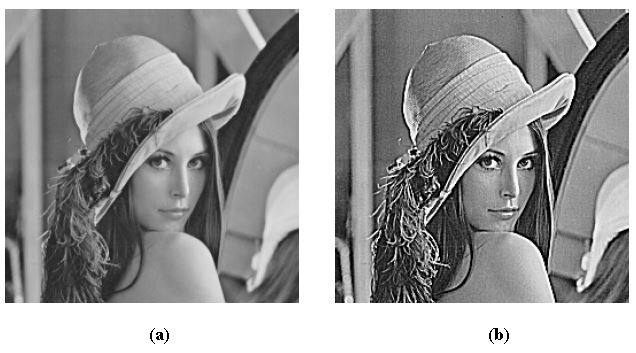
1. Perbaikan Kualitas Citra (*image enhancement*)
2. Pemugaran Citra (*image restoration*)
3. Pemampatan Citra (*image transformation*)
4. Segmentasi Citra (*image segmentation*)
5. Analisis Citra (*image analysis*)

Operasi-operasi pengolahan citra tersebut akan diuraikan ppad sub bab berikutnya .

1. **Perbaikan Kualitas Citra (*image enhancement*)**

Perbaikan Kualitas Citra (*image enhancement*) adalah proses perbaikan sudut pandang citra menjadi lebih baik (Kadir, et al., 2013). *Image Enhancement*  berfungsi untuk memperbaiki kualitas citra dengan cara memanipulasi parameter-parameter citra seperti perbaikan kontras pada citra yang gelap menjadi terang, perbaikan tepi objek (*edge* *enhancement*), penajaman (*sharpening*), pemberian warna semu (*pseudocoloring*), dan pelapisan derau (*noise filtering*) sehingga ciri-cir khusus yang terkandung didalam citra lebih terlihat jelas (Munir, 2004). Contoh-contoh operasi perbaikan citra:

1. Perbaikan kontras gelap/terang.
2. Perbaikan tepian objek (*edge enhancement*).
3. Penajaman (*sharpening*).
4. Pemberian warna semu (*pseudocoloring*).
5. *Noise filtering*.

**

Gambar (a) Citra Wanita asli, (b) Citra Wanita setelah ditajamkan

(Statistic For Image Sharpening, 2006)

1. **Pemugaran Citra (*image restoration*)**

Pemugaran citra (*image restoration*) adalah proses merekontruksi atau perbaikan penurunan kualitas citra yang disebabkan oleh derau (*noise*) sehingga kualitas citra menjadi keadaan asli (Kadir, et al., 2013). *Image restoration* berfungsi untuk meminimumkan atau menghilangkan kerusakan (degredrasi) pada citra seperti menghilangkan kesamaran (*deblurring*) dan derau (*noise*) (Munir, 2004). Contoh-contoh operasi pemugaran citra:

1. Penghilangan kesamaran (*deblurring*).
2. Penghilau derau (*noise*).



Gambar 2. 6 (a) Citra Wanita yang kabur (blur), (b) Citra Wanita setelah debluring (Seetharaman, 2007)

1. **Pemampatan Citra (*image compression*)**

Jenis operasi ini dilakukan agar citra dapat dipresentasikan dalam bentuk yang lebih kompak sehingga memerlukan memori yang lebih sedikit. Hal penting yang harus diperhatikan dalam pemampatan adalah citra yang telah dimampatkan harus tetap mempunyai kualitas gambar yang bagus. Contoh metode pemampatan citra adalah metode JPEG.

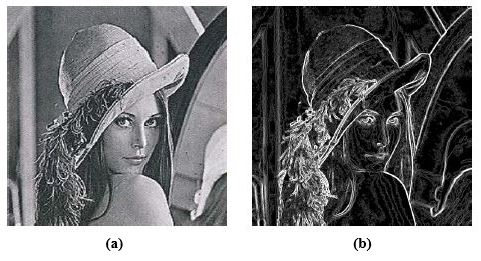
1. **Segmentasi Citra (*image segmentation*)**

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa segmen dengan suatu kriteria tertentu. Jenis operasi ini berkaitan erat dengan pengenalan pola. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya.

1. **Analisis Citra (*image analysis*)**

Analisis citra berfungsi untuk menghitung nilai kuantitatif dari citra untuk menghasilkan deskripsinya dengan cara mengekstraksi ciri-ciri tertentu yang digunakan untuk mengidentifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya (Munir, 2004). Contoh-contoh operasi analisis citra:

1. Pendeteksian tepi objek (*edge detection*).
2. Ektraksi batas (*boundary*).
3. Representasi daerah (*region*).



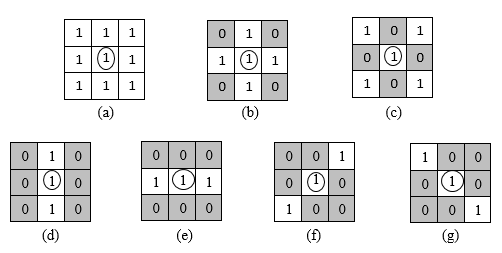
Gambar 2. 7 (a) Citra Wanita asli, (b) Citra hasil pendeteksian seluruh tepi

(Segmentasi Citra Digital Menggunakan Thresholding, 2016)

1. **Morphologi**

Morphologi(*morphology*) dapat diartikan sebagai bentuk dan struktur suatu objek atau dalam deskripsi lainnya disebutkan bahwa morphologi adalah susunan dan hubungan antar bagian pada suatu objek (Putra, 2010). Pada pengolahan citra, operasi morphologi merupakan metode dalam mendeteksi suatu objek gambar, dimana metode ini merepresentasikan citra dua objek dua dimensi sebagai suatu himpunan matematika, misal citra biner digambarkan sebagai suatu himpunanan titik-titik *pixels* dalam bidang biner Z, terbentuk dari titik-titik himpunan A dan titik-titik himpunan B yang nantinya membentuk objek (Nainggolan, et al., 2018) **.**

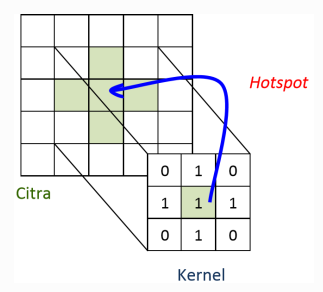
Morphologi cocok digunakan untuk ektraksi pada gambar dengan pemilihan ekstraksi yang didasarkan terhadap bentuk, ukuran, dan orientasi pada gambar (Zhou, et al., 2010). Operasi morphologi bertujuan untuk meningkatkan bentuk (struktur) sehingga lebih mudah untuk di kenali (Nainggolan, et al., 2018). Ada 2 operasi morphologi yaitu dilasi dan erosi. Kedua operasi dasar tersebut menjadi basis untuk membuat berbagai operasi seperti *opening, closing, hit and miss transform, thinning,* dan *thickening* (Putra, 2010). Operasi morphologi menggunakan dua input himpunan yaitu suatu citra (pada umumnya citra biner) dan suatu kernel, istilah kernel biasa disebut dengan *structuring elements* (elemen pembentuk struktur) merupakan suatu matrik dan pada umunya berukuran kecil (Putra, 2010). *Structuring elements* digunakan untuk membangun gambar dan output dari gambar tergantung dari kecocokan dari *structuring elements* pada gambar yang diberikan (Zhou, et al., 2010). Secara umum *structuring elements* digunakan sebagai berikut :



Gambar 2. 16 Berbagai bentuk structuring elements

(Hidayatullah, 2017)

Pada gambar posisi piksel yang dilingkari merupakan pusat koordinat dari *structuring elements*. Pemrosesan citra secara morphologi dilakukan dengan cara melakukan *passing* pada sebuah *structuring elements* terhadap sebuah citra, dimana *structuring elements* merupakan sebuah matriks yang terdiri dari bilangan “0” dan “1” (Nainggolan, et al., 2018). *Passing*  yang dilakukan dengan menyesuaikan hotspot *structuring elements* secara umum dilakukan sebagai berikut:



Gambar 2. 17 penerapan structuring elements pada citra

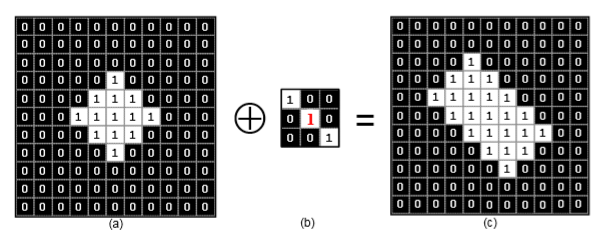
(Kadir & Susanto, 2013)

1. **Dilasi (*Dilation*)**

Dilasi (*dilation*) adalah proses morfologi yang ditujukan untuk memperluas area atau ukuran objek (Madenda, 2015)**.** Proses dilasi dilakukan dengan membandingkan setiap piksel citra input dan nilai citra pusat SE dengan cara melapiskan(*superimpose*) SE dan citra sehingga pusat SE tepat terhadap posisi piksel citra yang diproses. Jika paling sedikit ada 1 piksel pada SE sama dengan nilai piksel objek (*foreground*) citra maka piksel input diset nilainya dengan nilai piksel *foreground* dan bila semua piksel yang berhubungan adalah *background* maka input piksel diberi nilai piksel *background.* Proses serupa dilanjutkan dengan translasi SE piksel demi piksel pada citra input. Semakin besar ukuran SE maka semakin besar perubahan yang terjadi. SE berukuran kecil juga dapat memberikan hasil yang sama dengan SE berukuran besar dengan cara melakukan dilasi berulang kali. Efek dilasi adalah dapat memperbesar batas dari objek sehingga telihat semakin besar dan lubang yang terdapat pada objek akan tampak mengecil (Putra, 2010). Berdasarkan operasi matematika dilasi A oleh B, didefenisikan sebagai berikut :

(1)

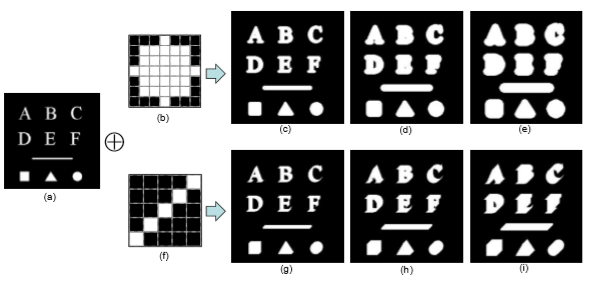
Berdasarkan rumus diatas A merupakan citra asli dan B merupakan *structure element,*



Gambar 2. 18 Operasi dilasi

(Zhou, et al., 2010)

Dilasi A oleh B adalah himpunan yang terdiri semua piksel B yang dipantulkan dan menimpah sebagian piksel A dan titik B yang menimpah citra A akan menyebabkan penebalan pada citra A. Titik referensi *structure element* Byang menimpah piksel pada citra A memiliki nilai 1. Berdasarkan rumus, contoh dilasi sebagai berikut:



Gambar 2. 19 Contoh Operasi Dilasi

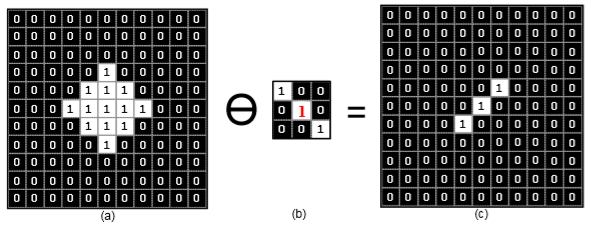
(Zhou, et al., 2010)

1. **Erosi (*Erosion*)**

Erosi (*Erosion*) adalah proses morfologi yang ditujukan untuk mengikis atau memperkecil luas permukaan ukuran suatu area atau objek (Madenda, 2015)**.** Erosi dilakukan dengan membandingkan setiap piksel citra input dan nilai citra pusat SE dengan cara melapiskan(*superimpose*) SE dan citra sehingga pusat SE tepat terhadap posisi piksel citra yang diproses. Jika semua piksel pada SE tepat sama dengan semua nilai piksel objek(*foreground*) citra maka piksel input diset nilainya dengan nilai piksel *foreground,* bila tidak maka input piksel diberi nilai piksel *background.* Proses serupa dilanjutkan dengan translasi SE piksel demi piksel pada citra input. Proses erosi merupakan kebalikan dari prose dilasi. Jika dalam dilasi menghasilkan objek yang lebih luas maka dalam proses erosi akan menghasilkan objek yang mengecil. Lubang pada objek juga akan tampak membesar seiring mengecilnya batas pada objek tersebut (Putra, 2010).Berdasarkan operasi matematika, erosi A oleh B, didefenisikan sebagai berikut :

(2)

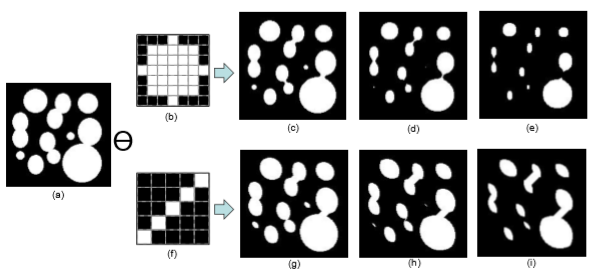
Berdasarkan rumus diatas A merupakan citra asli dan B merupakan *structure element,*



Gambar 2. 20 Operasi erosi

(Zhou, et al., 2010)

Erosi A oleh B adalah himpunan yang terdiri semua piksel B yang dipantulkan terhadap piksel A dan titik B yang menimpah citra A akan menyebabkan pengikisan terhadap citra A . Titik referensi *structure element* Byang menimpah piksel pada citra A tidak memiliki nilai sama dengan 0 maka nilai pada *structure element* B akan dieliminasi sehingga gambar pada citra asli akan semakin mengecil. Berdasarkan rumus, contoh erosi sebagai berikut:



Gambar 2. 21 Operasi erosi

(Zhou, et al., 2010)

1. ***Opening***

*Opening* digunakan untuk menghilangkan objek-objek kecil yang dianggap sebagai noise dimana besar kecilnya objek yang dapat dihilangkan tergantung pada besar kecilnya kernel**.**

1. ***Closing***

*Closing* memperbesar batas luar dari objek *foreground* dan juga menutup lubang kecilyang terletak ditengah objek, namun hasil *closing* tidak sebesar hasil dilasi. Hasil dilasi akan menyebabkan pembengkakan bentuk keseluruhan objek dan dikurangi dengan menerapkan proses erosi setelah proses dilasi dilakukan (Putra, 2010).

1. ***Tresholding***

*Tresholding* merupakan proses pengembangan yang akan menghasilkan citra biner dengan dua nilai tingkat keabuan yaitu hitam dan putih, sehingga daerah yang termasuk objek dan *background* dari sebuah citra dapat diketahui dengan jelas. Terdapat dua jenis pengembangan pada metode *tresholding* yaitu *global thresholding* dan *adaptive thresholding* (Putra, 2010). Berikut metode *tresholding* :

* 1. *Global Thresholding*

Pada bagian ini *tresholding* dilakukan dengan cara mempartisi histogram dengan menggunakan sebuah *treshold global* T yang digunakan untuk semua bagian citra. Namum, pada pengambangan *global* akan banyak informasi yang hilang karena menggunakan hanya satu nilai T untuk semua nilai piksel. Persamaan yang dipakai untuk mengkonversi nilai piksel citra *grayscale* menjadi biner dengan cara sebagai berikut :

(3)

Keterangan:

*f(x,y)* : Citra *grayscale*

*g(x,y)* : Citra biner

T : Nilai *threshold*

Kualitas hasil citra biner bergantung pada nilai T yang digunakan. Sehingga nilai T sangat penting dalam proses pengambangan.

* 1. *Adaptive Thresholding*

Pengambangan adaptif (*adaptive thresholding*) merupakan pengambangan yang menggunakan nilai ambang lokal, dihitung berdasarkan stastik piksel-piksel tetangga. Hal ini didasarkan bagian-bagian kecil dalam citra mempunyai iluminasi sama. Sehingga, lebih tepat kalau nilai ambang dihitung berdasarkan bagian-bagian kecil dalam citra dan bukan berdasarkan seluruh piksel dalam citra (Kadir & Susanto, 2013).

Terdapat 3 pendekatan yang digunakan dalam pengambangan adaptif. Pendekatan pertama menggunakan statistika rerata terhadap intensitas lokal. Pada saat tertentu nilai konstan ikut dilibatkan. Dengan rumus sebagai berikut:

(4)

Pendekatan pengambangan adaptif yang kedua menggunakan statistik median. Secara matematispendekatan ini dapat ditulis sebagai berikut :

(5)

Pendekatan pengambangan adaptif yang ketiga menggunakan statistik maksimum dan minimum, dan dinyatakan dengan rumus.

(6)

Dengan keterangan masing-masing rumus :

W : Jendela pada citra

Nw : Jumlah *pixel* dalam jendela

C : Konstanta

* + 1. Deteksi Tepi

Deteksi tepi (*edge detection*) adalah suatu proses yang menghasilkan tepi dari objek citra. Dengan tujuan memperbaiki detail dari citra yang kabur,bisa terjadi karena *error* atau adanya efek dari akusisi citra (Achmad Sahir Ramdani, 2013). Pada umumnya tepi objek merupakan pertemuan objek bagian latar belakang ditandai oleh titik yang nilai intensitas keabuan yang sangat berbeda dengan titik yang berada pada sebuah objek (Sutoyo, et al., 2009). Deteksi tepi bertujuan untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area, tepi mengandung informasi sangat penting dan dapat berupa bentuk ataupun ukuran objek (Kadir & Susanto, 2013).

Deteksi tepi dibagi menjadi dua golongan. Golongan pertamadisebut deteksi tepi orde pertama, yang bekerja dengan menggunakan turunan atau difrensial orde pertama. Golongan kedua dinamakan deteksi tepi orde kedua. Termasuk dalam kedua kelompok ini ialah :

1. Operator *Roberts*

Operator *roberts* adalah operator pendeteksi tepi yang menggunakan kernel berukuran 2x2 piksel (Sutoyo, et al., 2009). Operator *roberts* memberikan tanggapan lemah terhadap tepi kecuali kalau tepi sangat tajam (Kadir & Susanto, 2013).

Dimisalkan berikut adalah posisi pada citra f

X x+1

|  |  |
| --- | --- |
|  |  |
|  |  |

Y

Y+1

Dimana nilai operator *roberts* dihitung menggunakan kernel konvolusi sebagai berikut :

(5)

Misalkan f adalah citra yang akan dikenai operator *roberts.* Maka, nilai operator *roberts* pada (y,x) didefinisikan sebagai

r(y,x) (5)

Dalam hal ini = f(y,x), = f(y,x+1), =f(y+1,x), dan = f(y+1,x+1).

1. Operator *Prewitt*

Operator *prewitt* dikemukan oleh prewitt pada tahun 1966. Untuk mempercepat komputasi, bagian berniali nol tidak diproses. Oleh karena itu perhitungan operasi *prewitt* ditulis menjadi (Kadir & Susanto, 2013)

r(y,x)= 2 + (f(y+1,x-1)+f(y+1,x)+f(y+1,x+1)-f(y-1,x-1)-f(y-1,x)-f(y-1,x+1))2 (6)

1. Operator *Sobel*

Operator *sobel* operator pendeteksi tepiyang berfungsi untuk menghitung gradien sehingga perkiraan gradien berada ditengah kernel dan menghindari adanya perhitungan gradien dititik interpolasi (Sutoyo, et al., 2009). Operator *sobel*  lebih sensitif terhadap tepi diagonal daripada tepi vertikal dan horizontal (Kadir & Susanto, 2013). Operator ini menggunakan kernel yang berukuran 3x3 piksel (x,y) sebagai berikut :

(7)

Secara matematis, gradien pada operator *sobel* dapat dihitung menggunakan persamaan sebagai berikut:

M= (7)

Dimana M adalah besar gradien di titik tengah kernel dan dan dapat dihitung menggunakan persamaan berikut:

= ( + ) - ( + ) (8)

= ( + ) - ( + ) (9)

Dimana c adalah konstanta yang bernilai 2. dan diimplementasikan menjadi kernel berikut:

= = (10)

1. Operator *Frei-Chen*

Operator *Frei-Chen* sering disebut juga sebagai operator isotropik. Operator ini mirip dengan operator *sobel*  namun setiap angka 2 diganti menjadi

x-1 x x+1

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Y-1

Y

Y+1

Dengan nilai kernel pada operasi *frei-chen*

= = (10)

1. Operator *Laplacian of Gaussian*

Deteksi tepi orde kedua yang makin kurang sensitif terhadap derau adalah *Laplacian of Gaussian.* Disebabkan penggunaan fungsi *gaussian* yang memuluskan citra dan berdampak terhadap pengurangan derau pada citra (Kadir & Susanto, 2013). Operator *Laplacian of Gaussian* diperoleh melalui konvolusi dengan :

LoG(y,x) = ] (11)

* + 1. Ekstraksi Fitur

Ciri yang dapat diekstrak dari suatu objek dalam citra antara lain adalah warna, bentuk, ukuran, dan tekstur. Dimana ciri tersebut digunakan sebagai parameter untuk membedakan antara objek yang satu dengan objek lainnya. Parameter yang digunakan untuk mewakili ciri bentuk adalah *mectic* dan *eccentricity* sedangkan parameter yang mewakili ciri ukuran adalah luas dan keliling (Pamungkas, 2017). Ada beberapa fitur yang digunakan dalam mengekstrak objek pada citra, diantaranya (Kadir & Susanto, 2013) :

1. Tanda Tangan Kontur

Tanda tangan kontur dalam matematis didefinisikan sebagai :

d(n)= (12)

Dalam hal ini (,) menyatakan pusat massa kontur yang diperoleh melalui rumus:

= = (13)

1. Deskriptor *fourier*

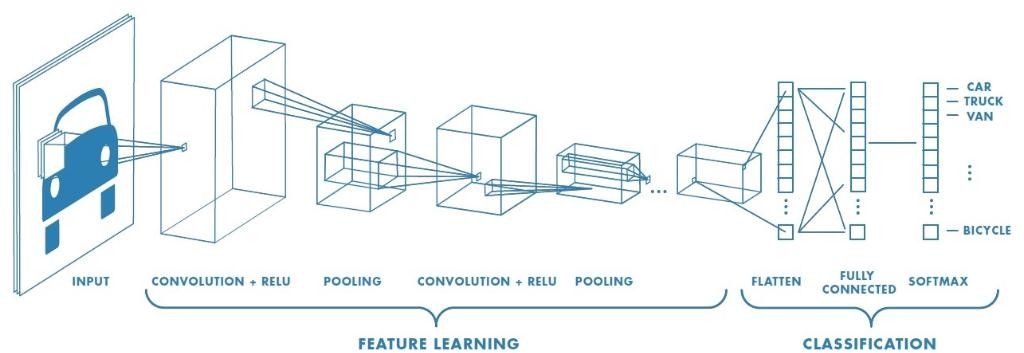
Deskriptor *fourier* biasa dipakai untuk menjabarkan bentuk dalam dua dimensi dengan menggunakan trasnformasi *fourier*. Dengan menggunakan deskriptor *fourier,* suatu bentuk dapat dinyatakan dengan sejumlah bilangan yaitu koefisien *fourier.* Dasar yang digunakan untuk memperoleh koefisien *fourier* adalah transformasi *fourier* berdimensi satu. Transformasi *fourier* diskret berupa :

= ), n=0,1,2,....,N-1 (14)

Untuk n=0,1,2,....,N-1 merupakan deskriptor *fourier* untuk bentuk (Kadir & Susanto, 2013).

* 1. **Convolution Neural Network**

*Convolutional Neural Network* (CNN) merupakan pengembangan dari *multilayer perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap. Masukan (*input*) dan keluaran (*output*) dari setiap tahap adalah terdiri dari beberapa *array* yang biasa disebut *feature map.* Setiap tahap terdiri dari tiga *layer* yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*. Berikut adalah jaringan arsitektur *Convolutional Neural Network* :

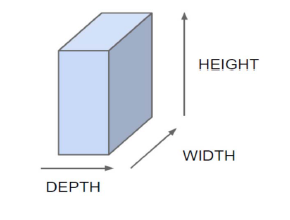


Arsitektur *Convolutional Neural Network*(Medium Samuel Sena, 2017)

Berdasarkan gambar diatas, tahap pertama pada arsitektur CNN adalah tahap konvolusi. Tahap ini dilakukan dengan menggunakan sebuah kernel dengan ukuran tertentu. Perhitungan jumlah kernel yang dipakai tergantung dari jumlah fitur yang dihasilkan. Kemudian dilanjutkan menuju fungsi aktivasi, biasanya menggunakan fungsi aktivasi ReLU ( *Rectifier Linear Unit* ), Selanjutnya setelah keluar dari proses fungsi aktivasi kemudian melalui proses *pooling*. Proses ini diulang beberapa kali sampai didapatkan peta fitur yang cukup untuk dilanjutkan ke *fully connected neural network,* dan dari *fully connected network* adalah *output class.*

* + 1. ***Convolution Layer***

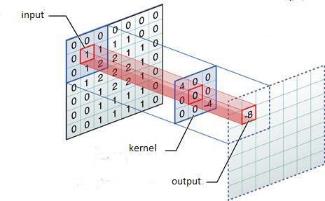
Konvolusi adalah istilah matematis dimana pengaplikasian sebuah fungsi pada *output* fungsi lain secara berulang. Operasi konvolusi merupakan operasi pada dua fungsi argumen bernilai nyata. Operasi ini menerapkan fungsi *output* sebagai *Feature Map* dari input citra. Representasi CNN dapat dituliskan sebagai berikut :

****

**Gambar Inputan 3 dimensi Representasi dari CNN**

(ALBAWI, et al., 2017)

Operasi konvolusi menggunakan *filter* yang juga disebut sebagai *kernel* untuk mendeteksi fitur- fitur, seperti sudut yang ada di seluruh gambar. *Filter* hanyalah sebuah matriks nilai yang disebut bobot, dilatih untuk mendeteksi fitur tertentu. *Filter* bergerak di setiap bagian gambar untuk memeriksa apakah fitur yang dimaksudkan untuk dideteksi tersedia. Untuk memberikan nilai yang menunjukkan seberapa tepat fitur yang tersedia tersebut melakukan operasi konvolusi (Escontrela, 2018).



Gambar 2.35 Operasi Konvolusi pada Tahap *Convolution Layer*

(Escontrela, 2018)

Dalam matematika yang tepat, prosedur konvolusi dapat dinyatakan sebagai persamaan berikut ;

(1)

Persamaan 1 diulang untuk semua 0 ≤ d <D = Dl + 1, dan untuk setiap lokasi spasial (il + 1, jl + 1) memuaskan

0 ≤ il+1 < Hl −H + 1 = Hl+1 , (2)

1. ≤ jl+1 < Wl −W + 1 = Wl+1  (3)

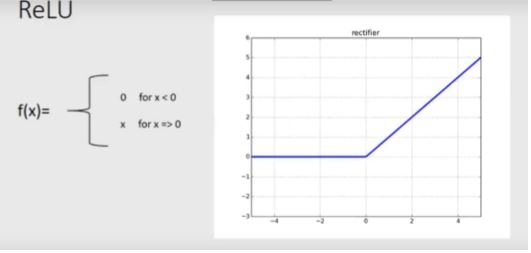
Dalam persamaan ini, mengacu pada elemen xl yang diindeks oleh triplet (il + 1 + i, jl + 1 + j, dl).

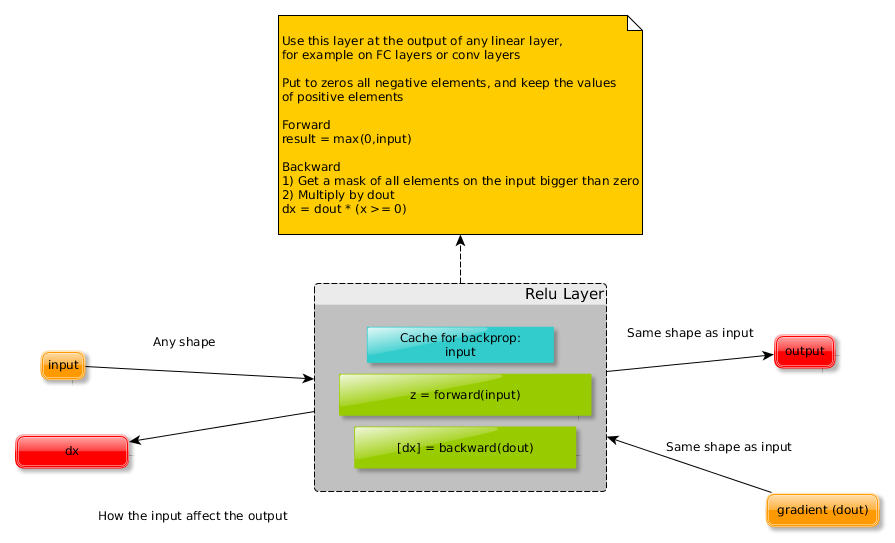
* + 1. **Fungsi Aktivasi**

Fungsi aktivasi merupakan salah satu yang penentu performa dari neural network. Ada beberapa jenis fungsi aktivasi yang ada, dalam penelitian ini fungsi aktivasi yang digunakan adalah ReLU . ReLu pada dasarnya merupakan sebuah fungsi non-linier yang paling sederhana (Vydana et al., 2017). Apabila mendapat input positif, maka turunannya hanya 1, dengan kata lain aktivasi hanya men-threshold pada nilai 0. Adapun rumus dari ReLU adalah

𝑓𝑥= max (𝑥,0) .

ReLU digunakan dalam setiap lapisan konvolusi. Output yang diberikan oleh ReLU ditunjukkan seperti gambar dibawah ini :





* + 1. **Lapisan ReLu**

Lapisan ReLU tidak mengubah ukuran input, yaitu, xl dan y berbagi ukuran yang sama. Bahkan, Unit Linear yang diperbaiki dapat dianggap sebagai pemotongan yang dilakukan secara individual untuk setiap elemen dalam tensor input

**Yi,j,d = max{0, xl i,j,d},** (1) **with**

**0 ≤ i < Hl = Hl+1, 0 ≤ j < Wl = Wl+1, and 0 ≤ d < Dl = Dl+1.**

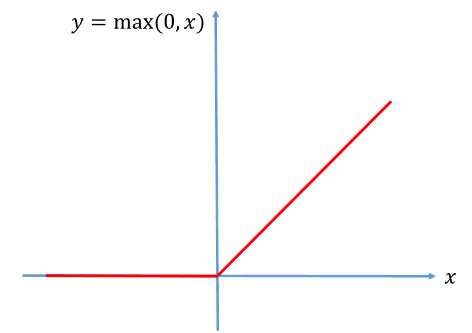
Tidak ada parameter di dalam lapisan ReLU, karenanya tidak perlu untuk pembelajaran parameter di lapisan ini. Berdasarkan Persamaan 1, jelas bahwa

(2)

di mana adalah fungsi indikator, yang menjadi 1 jika argumennya benar, dan 0 jika argumennya salah. Karena itu, kita punya

(3)

Perhatikan bahwa y adalah alias untuk xl + 1. Secara tegas, fungsi max (0, x) tidak dapat dibedakan pada x = 0, oleh karena itu Persamaan 2 sedikit bermasalah dalam teori. Dalam praktiknya, ini bukan masalah dan ReLU aman digunakan. Tujuan ReLU adalah untuk meningkatkan nonlinier dari CNN. Meskipun sederhana, fungsi ReLu adalah fungsi nonlinear, seperti yang diilustrasikan dalam Gambar dibawah ini.

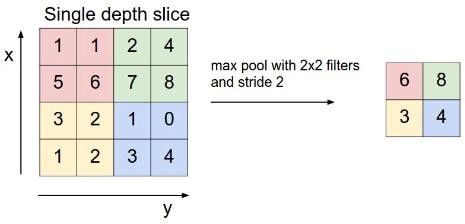


Namun, sigmoid logistik bekerja secara signifikan lebih buruk daripada ReLU dalam pembelajaran CNN. Pengenalan ReLU untuk menggantikan sigmoid adalah perubahan penting dalam CNN, yang secara signifikan mengurangi kesulitan dalam mempelajari parameter CNN dan meningkatkan akurasinya. ReLU menetapkan gradien dari beberapa fitur di lapisan ke-l ke 0, tetapi fitur-fitur ini tidak diaktifkan.

* + 1. **Operasi Pooling**

*Pooling* merupakan pengurangan ukuran matriks dengan menggunakan operasi *pooling. Pooling Layer* biasanya berada setelah *convolution layer*. Pada dasarnya *pooling layer* terdiri dari sebuah filter dengan ukuran dan stride tertentu yang akan secara bergantian bergeser pada seluruh area *feature map.* Dalam *pooling layer* terdapat dua macam *pooling* yang biasa digunakan yaitu *average pooling* dan *max- pooling*. Nilai yang diambil pada *average pooling* adalah nilai rata-rata, sedangkan pada *max-pooling* adalah nilai maksimal.

Lapisan *Pooling* yang dimasukkan diantara lapisan konvolusi secara berturut-turut dalam arsitektur model CNN dapat secara progresif mengurangi ukuran volume output pada *Feature Map*, sehingga mengurangi jumlah parameter dan perhitungan di jaringan, untuk mengendalikan *Overfitting*. Lapisan pooling bekerja di setiap tumpukan *feature map* dan melakukan pengurangan pada ukurannya. Bentuk lapisan *pooling* umumnya dengan menggunakan filter dengan ukuran 2x2 yang diaplikasikan dengan langkah sebanyak dua dan beroperasi pada setiap irisan dari inputnya. Berikut ini adalah contoh gambar operasi *max-pooling* :



Operasi *Max-Pooling*

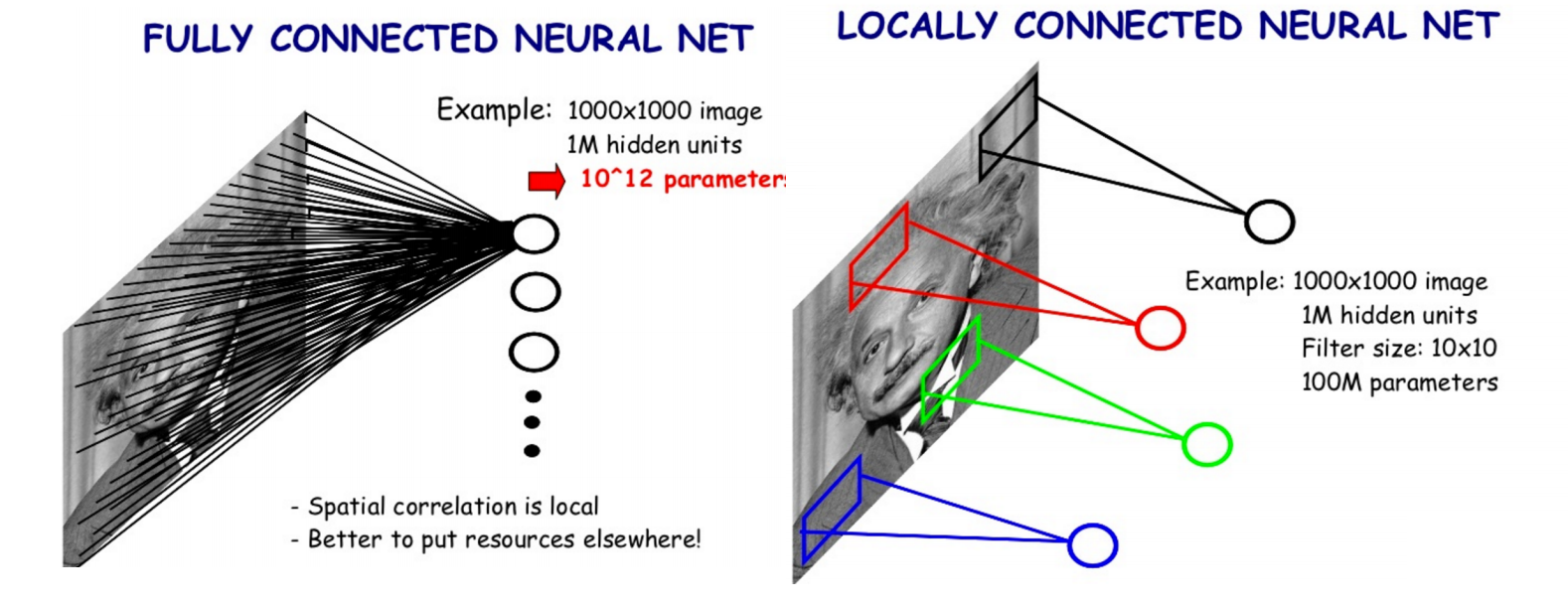
(Medium Samuel Sena, 2017)

Berdasarkan gambar diatas menunjukan proses dari *max-pooling*. *Output* dari proses *pooling* adalah sebuah matriks dengan dimensi yang lebih kecil dibandingkan dengan citra awal. Lapisan pooling diatas akan beroperasi pada setiap irisan kedalaman volume input secara bergantian. Jika dilihat dari gambar diatas operasi *max-pooling* dengan menggunakan ukuran filter 2x2. Masukan pada proses tersebut berukuran 4x4, dari masing-masing 4 angka pada input operasi tersebut diambil nilai maksimalnya kemudian dilanjutkan membuat ukuran *output* baru menjadi ukuran 2x2.

* + 1. **Fully-Connected Layer**

*Fully-Conected Layer* adalah sebuah lapisan dimana semua *neuron* aktivasi dari lapisan sebelumnya terhubung semua dengan *neuron* di lapisan selanjutnya sama seperti halnya dengan *neural network biasa.* Pada dasarnya lapisan ini biasanya digunakan pada MLP ( *Multi Layer Perceptron* ) yang mempunyai tujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear.

Perbedaan antara lapisan *Fully-Connected* dan lapisan konvolusi biasa adalah neuron di lapisan konvolusi terhubung hanya ke daerah tertentu pada input, sementara lapisan *Fully-Connected* memiliki neuron yang secara keseluruhan terhubung. Namun, kedua lapisan tersebut masih mengoperasikan produk dot, sehingga fungsinya tidak begitu berbeda. Berikut ini adalah proses *fully-connected* :



*Processing of a Fully-Connected Layer*

Seperti disebutkan di atas, satu keuntungan dari lapisan konvolusi adalah bahwa konvolusi adalah operasi lokal. Luas spasial dari sebuah kernel seringkali kecil (mis., 3 × 3). Satu elemen dalam xl + 1 biasanya dihitung hanya menggunakan sejumlah kecil elemen dalam input xl. *Fully Connected Layer* merujuk ke lapisan yang jika perhitungan elemen apa pun dalam output xl + 1 (atau y) membutuhkan semua elemen dalam input xl. *Fully Connected Layer*  kadang-kadang berguna di akhir model *deep* CNN. Sebagai contoh, setelah konvolusi dilakukan secara berulang kali, lapisan ReLU dan *pooling* adalah *output* dari layer saat ini berisi representasi terdistribusi untuk gambar input. Misalkan input dari layer xl memiliki ukuran Hl × Wl × Dl. Jika kita menggunakan kernel konvolusi yang ukurannya Hl × Wl × Dl, maka D kernel tersebut membentuk order 4 tensor dengan ukuran Hl × Wl × Dl × D. Outputnya adalah y ∈ RD. Jelaslah bahwa untuk menghitung elemen apa pun di y, kita perlu menggunakan semua elemen dalam input xl. Oleh karena itu, lapisan ini adalah lapisan *Fully Connected,* tetapi dapat diimplementasikan sebagai lapisan konvolusi. Oleh karena itu, kita tidak perlu menurunkan aturan pembelajaran untuk *Fully Connected Layer* secara terpisah.