

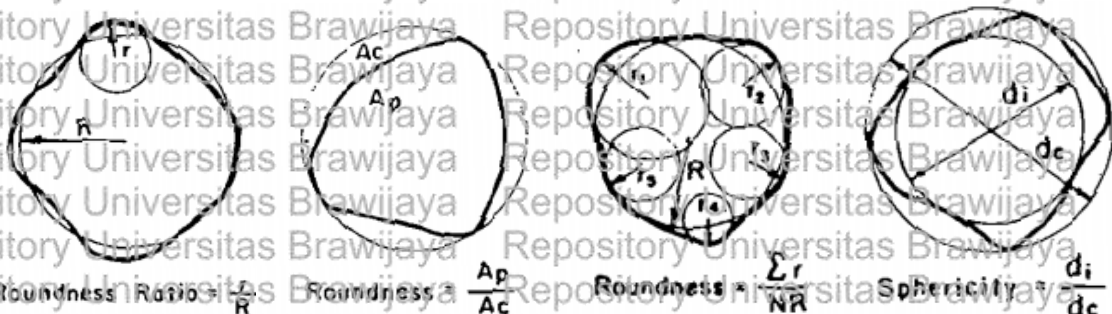
BAB II TINJAUAN PUSTAKA

2.1 Kacang

Kacang adalah istilah non-botani yang biasa dipakai untuk menyebut biji sejumlah tumbuhan polong-polongan (namun tidak semua). Dalam percakapan sehari-hari, kacang dipakai juga untuk menyebut buah (polong) atau bahkan tumbuhan yang menghasilkannya. Kata ini sebenarnya dipakai untuk menyebut biji kering yang berbentuk menyerupai ginjal dan dimakan setelah dikeringkan.

Pada umumnya orang mengenali jenis kacang berdasarkan ciri visual yang tampak pada kacang seperti warna dan bentuknya. Jenis kacang yang sangat beragam dan hampir memiliki kesamaan dalam bentuk maupun warna memungkinkan terjadinya kesalahan dalam mengenalinya. Secara obyektif, ciri warna lebih dapat diandalkan untuk dipercaya, namun akan lebih baik kedua sifat ini digunakan secara bersama-sama. Ciri warna dan bentuk tersebut digunakan sebagai ciri pembeda antara kacang satu dengan yang lain, tentu saja dengan syarat bahwa kacang yang akan diidentifikasi memenuhi kriteria yang ditetapkan, yaitu kacang dalam keadaan kering, bersih dan tanpa cacat.

Salah satu cara untuk mengenali ciri bentuk suatu kacang adalah berdasarkan sifat fisiknya dan sifat fisik ini dapat digunakan sebagai indeks pengukuran. Sifat fisik ini berupa bentuk dan ukuran yang meliputi kebulatan (*roundness*), rasio kebulatan (*roundness ratio*), kebundaran (*sphericity*), rasio aksial (*axial ratio*), luas (*area*), keliling (*perimeter*), dan lain-lain.



Gambar 2.1 Contoh sifat fisik sebagai indeks pengukuran

(Sumber: Nuri N. Mohsenin, 1986)



2.1.1 Kacang Merah (*Vigna angularis* (Willd.) Ohwi & H. Ohashi)

Kacang merah atau umumnya disebut dengan buncis, merupakan sejenis sayuran kacang yang berbuah dan sangat kaya dengan kandungan protein. Ia dipercayai berasal dari Amerika Tengah dan Amerika Selatan. Pokok kacang buncis bertabiat tumbuh melilit, mempunyai akar tunjang dan sisi yang panjang dan memerlukan pancang untuk memanjang.

Kacang merah ini memiliki 2 tipe yaitu, Kacang Merah (*Phaseolus vulgaris* L.) berasal dari Amerika, sedangkan kacang buncis tipe tegak (kidney bean) atau kacang jogo adalah tanaman asli lembah Tahuacan-Meksiko. Penyebarluasan tanaman buncis dari Amerika ke Eropa dilakukan sejak abad 16. Daerah pusat penyebaran dimulai di Inggris (1594), menyebar ke negara-negara Eropa, Afrika, sampai ke Indonesia.

Kacang merah akan dapat tumbuh baik di daerah basah atau dingin pada ketinggian 1400-2000 meter dari permukaan laut dan dipanen 6 bulan setelah penanaman. Kacang merah dapat digolongkan menjadi 2 macam, yaitu kacang merah yang tumbuhnya kerdil dan yang tumbuh memanjang dan memerlukan para-para. Warna bijinya merah bertotol-totol merah tua, sesuai dengan namanya. Buahnya (polong) berwarna kuning, kalau masih muda berwarna hijau dan kadang-kadang berwarna merah. Kalau sudah tua berubah menguning, mengering, dan siap panen. Buahnya yang berbentuk polong memanjang, hanya sedikit lebih panjang bila dibandingkan dengan buncis. Dalam satu polong ada 2-3 biji kacang merah. Bentuk kacang merah yang masih utuh sama dengan kacang buncis, baik daun, bunga maupun bentuk polongnya.

2.1.2 Kacang kedelai (*Glycine max* (L.) Merrill)

Kedelai merupakan tanaman pangan berupa semak yang tumbuh tegak. Kedelai jenis liar *Glycine ururiensis*, merupakan kedelai yang menurunkan berbagai kedelai yang dikenal sekarang kedelai (*Glycine max* (L.) Merrill). Berasal dari daerah Manshukuo (Cina Utara), di Indonesia, dibudidayakan mulai abad ke-17 sebagai tanaman makanan dan pupuk hijau. Penyebaran tanaman kedelai ke Indonesia berasal dari daerah Manshukuo menyebar ke daerah Mansyuria; Jepang (Asia Timur) dan negara-negara lain di Amerika dan Afrika. (AAK, 1989).

Akar tanaman kedelai terdiri atas akar tunggang, akar lateral, dan akar serabut. Pada tanah yang gembur, akar ini dapat menembus tanah sampai kedalaman 1,5 m. Pada akar lateral terdapat bintil-bintil akar yang merupakan kumpulan bakteri



rhizobium pengikat N dari udara. Bintil akar ini biasanya akan terbentuk 15-20 hari setelah tanam, selain sebagai penyerap unsur hara dan penyangga tanaman, pada perakaran merupakan tempat terbentuknya bintil/nodul akar yang berfungsi sebagai pabrik alami terfiksasinya nitrogen udara oleh aktivitas bakteri Rhizobium (Tambas dan Rakhman, 1986).

Polong kedelai pertama terbentuk sekitar 7-10 hari setelah munculnya bunga pertama. Panjang polong muda sekitar 1 cm, jumlah polong yang terbentuk pada setiap ketiak tangkai daun sangat beragam, antara 1-10 buah dalam setiap kelompok. Pada setiap tanaman, jumlah polong dapat mencapai lebih dari 50, bahkan ratusan. Kecepatan pembentukan polong dan pembesaran biji akan semakin cepat setelah proses pembentukan bunga berhenti. Ukuran dan bentuk polong menjadi maksimal pada saat awal periode pemasakan biji. Hal ini kemungkinan diikuti oleh perubahan warna polong, dari hijau menjadi kuning kecoklatan pada saat masak (Adisarwanto, 2005). Biji kedelai berkeping dua yang terbungkus oleh kulit biji. Embrio terletak di antara keping biji. Warna kulit biji bermacam-macam, ada yang kuning, hitam, hijau atau coklat.

2.1.3. Kacang Hijau (*Phaseolus radiatus* L.)

Kacang hijau adalah sejenis tanaman budidaya dan palawija yang dikenal luas di daerah tropika. Ciri fisik yang paling terlihat jelas adalah warnanya yang hijau, dan semakin tua kacang warna kacang ini akan semakin gelap dan kecoklatan. Tumbuhan yang termasuk suku polong-polongan (Fabaceae) ini memiliki banyak manfaat dalam kehidupan sehari-hari sebagai sumber bahan pangan berprotein nabati tinggi. Kacang hijau di Indonesia menempati urutan ketiga terpenting sebagai tanaman pangan legum, setelah kedelai dan kacang tanah.

Bagian paling bernilai ekonomi adalah bijinya. Biji kacang hijau direbus hingga lunak dan dimakan sebagai bubur atau dimakan langsung. Biji matang yang digerus dan dijadikan sebagai isi onde-onde, bakpau, atau gandas turi. Kecambah kacang hijau menjadi sayuran yang umum dimakan di kawasan Asia Timur dan Asia Tenggara dan dikenal sebagai tauge. Kacang hijau bila direbus cukup lama akan pecah dan pati yang terkandung dalam bijinya akan keluar dan mengental, menjadi semacam bubur. Tepung biji kacang hijau, disebut di pasaran sebagai tepung hunkue, digunakan dalam pembuatan kue-kue dan cenderung membentuk gel. Tepung ini juga dapat diolah menjadi mi yang dikenal sebagai soum.



2.2 Pengolahan Citra Digital

Pengolahan citra adalah suatu metode yang digunakan untuk mengolah gambar sehingga dihasilkan gambar lain sesuai dengan keinginan kita. Untuk mengambil gambar bisa dilakukan oleh kamera atau alat-alat lain yang bisa digunakan untuk menstransfer gambar missal scanner.

2.3 Citra Digital

Citra digital adalah citra yang bisa diolah langsung oleh komputer dan tersimpan dalam media simpan digital misalnya memori, harddisk, CD, dll. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai real maupun kompleks yang direpresentasikan dengan bit tertentu. Citra digital didefinisikan sebagai fungsi $f(x,y)$ berukuran M baris dan N kolom, dimana x dan y adalah koordinat spasial dan nilai $f(x,y)$ adalah intensitas citra pada koordinat tersebut. Citra digital dapat ditulis dalam bentuk matrik sebagai berikut:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix} \quad (2-1)$$

dengan:

$f(x,y)$ = intensitas pixel pada posisi x dan y

M = jumlah kolom

N = jumlah baris

Nilai pada suatu irisan antara baris dan kolom (pada posisi x,y) disebut dengan pixel.

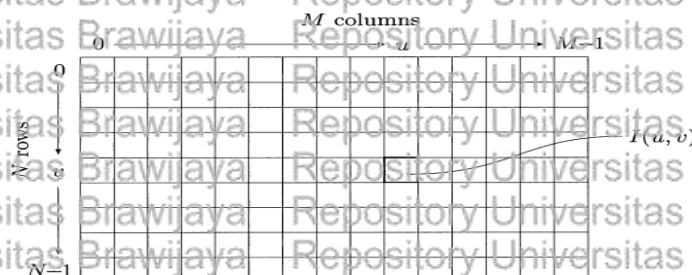
2.4 Pixel Dan Resolusi Citra

Satuan atau bagian terkecil dari suatu citra disebut pixel (picture element) yang berarti elemen citra. Umumnya citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal antar pixel adalah sama pada seluruh bagian citra.

Dalam komputer, setiap pixel diwakili oleh dua buah bilangan bulat (integer) untuk menunjukkan lokasinya dalam bidang citra dan sebuah nilai dalam bilangan bulat



(integer) untuk menunjukkan cahaya atau keadaan terang-gelap pixel tersebut. Untuk menunjukkan lokasi suatu pixel, koordinat $(0,0)$ digunakan untuk posisi kanan atas dalam bidang citra, dan koordinat $(m-1,n-1)$ digunakan untuk posisi kanan bawah dalam cita berukuran $m \times n$ pixel. Untuk menunjukkan tingkat pecahayaan suatu pixel, seringkali digunakan bilangan bulat yang besarnya 8-bit, dengan lebar selang nilai 00255, dimana 0 untuk warna hitam, 255 untuk warna putih dan tingkat abu-abu berada di antara nilai 0 dan 255 seperti yang diilustrasikan pada gambar dibawah ini:



Gambar 2.2 Image size and resolution

2.5 Jenis Citra

Pada citra nilai suatu pixel memiliki rentang atau range nilai tertentu. Range dari nilai citra dari minimum hingga maximum dan secara umum nilai range tersebut adalah 0-255, berikut adalah pembagian citra yang berdasarkan nilai pixelnya

2.5.1 Citra Biner

Citra Biner adalah citra yang hanya terdiri dari nilai pixel hitam dan putih. Sering kali citra biner disebut juga citra black and white atau citra monokrom. Pada citra biner ini hanya diperlukan 1 bit saja untuk merepresentasikan nilai dari setiap pixel. Biasanya citra biner ini digunakan pada pengolahan seperti segmentasi, morfologi dan lain-lain.



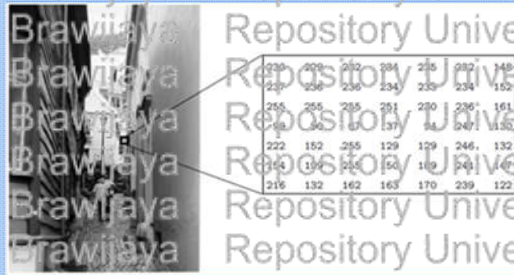
Gambar 2.3 Citra biner

(Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab, 2004)



2.5.2 Citra Grayscale

Citra *grayscale* merupakan citra digital dimana unsur RGB (Red Green Blue) hanya digunakan untuk menentukan tingkat intensitas pada suatu citra. Pada citra *grayscale* warna yang dimiliki adalah hitam, keabuan, putih. Tingkat keabuan merupakan warna abu dengan berbagai tingkatan dari hitam hingga mendekati warna putih dan citra *grayscale* memiliki kedalaman warna 8 bit.



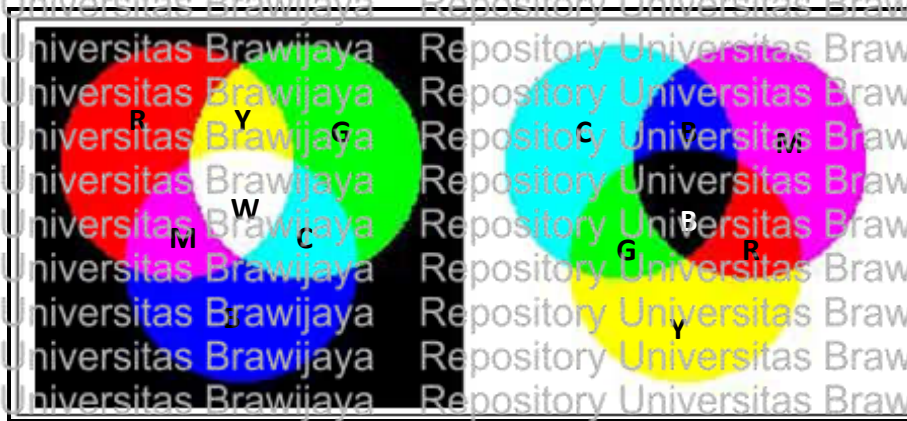
Gambar 2.4 Citra grayscale

(Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab, 2004)

2.5.3 Citra Berwarna

Citra berwarna merupakan gabungan dari beberapa lapis citra kanal warna. Teknologi dasar untuk menciptakan warna pada citra digital merupakan kombinasi dari kanal tiga warna dasar yaitu merah, hijau, dan biru (*RGB- Red, Green, Blue*) ditunjukkan pada Gambar 2.5. Kombinasi kanal warna ini didapat dari penangkapan intensitas cahaya pada frekuensi yang berbeda, yaitu frekuensi 560 nm (merah), 530 nm (hijau) dan 430 nm (biru). Pencampuran warna akan menghasilkan warna baru, seperti pada gambar 2.5.

Pada latar belakang warna hitam, warna merah dicampur dengan warna hijau menghasilkan warna kuning, warna merah dicampur dengan warna biru menghasilkan warna magenta, warna hijau dicampur dengan warna biru menghasilkan warna cyan, dan ketika warna merah, hijau dan biru dicampur akan menjadi warna putih. Sedangkan pada latar belakang warna putih, warna cyan dicampur dengan warna magenta menghasilkan warna biru, warna cyan dicampur dengan warna kuning menghasilkan warna hijau, warna magenta dicampur dengan warna kuning menghasilkan warna merah, dan ketika warna cyan, magenta, dan kuning dicampur akan menjadi warna hitam.



Gambar 2.5 Penambahan campuran warna merah, hijau dan biru

(Sumber: Adang Suhendra)

2.6 Model Warna RGB

Citra dalam format digital mempunyai informasi piksel dalam bentuk RGB, dengan kata lain informasi yang diperoleh bisa dikatakan masih dalam citra berwarna.

Setiap titik pada layar berisi angka yang bukan menunjukkan intensitas warna dari titik tersebut melainkan menunjukkan nomor warna dari warna yang dipilih. Pada setiap titik kita dapat memilih 256 warna, jika suatu citra memiliki 256 warna, maka fungsi-fungsi yang dimiliki oleh pengolahan citra tidak dapat mengolah atau memanipulasinya secara langsung karena citra tersebut tidak memiliki suatu tingkat kecerahan tertentu. Adapun masing-masing palette warna tabel memiliki tiga buah kombinasi angka yaitu R,G,B yang menentukan proporsi warna merah hijau dan biru. Dengan demikian di dalam suatu pixel akan diwakili oleh 3 byte memori yang masing-masing terdiri dari 1 byte untuk warna merah, 1 byte warna hijau dan 1 byte untuk warna biru.

Apabila suatu citra memiliki 254 warna, maka fungsi yang dimiliki oleh pengolahan citra tidak dapat mengolah atau memanipulasinya secara langsung karena citra tersebut tidak memiliki suatu tingkat kecerahan tertentu. Dimana tiap besaran pixel memiliki hubungan dengan indeks dalam palette warna.

Suatu palette warna tersusun dari warna-warna merah hijau dan biru (R,G,B) yang dalam bentuk tripelnya bernilai 0 sampai dengan 255 jadi suatu citra berwarna akan memiliki 768 byte informasi palette. Didalam suatu pixel akan diwakili oleh 3 byte memori, yang masing-masing terdiri dari :

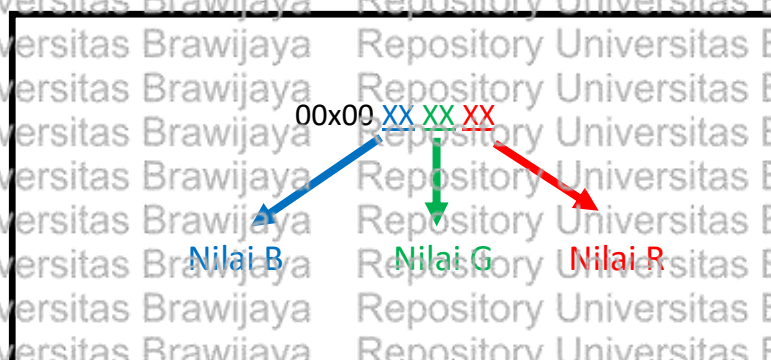


- 1 byte untuk Merah (Red/R)
- 1 byte untuk Hijau (Green/G)
- 1 byte untuk Biru (Blue/B)

Apabila terdapat 4 buah pixel di monitor dengan warna putih, hitam, biru dan kuning maka nilainya dapat ditentukan sebagai berikut :

- Putih, nilai ketiga byte nya adalah "F" atau tinggi
- Hitam, nilai ketiga byte nya adalah "0" atau rendah
- Biru, nilai Merah = 0, Hijau = 0, Biru = F
- Kuning, nilai Merah = F, Hijau = F, Biru = 0

Prinsip dasar pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna direpresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Definisi nilai warna tersebut berisikan variabel 0x00 yang menyatakan angka dibelakangnya adalah hexadesimal. Seperti yang dijelaskan dalam gambar 2.6



Gambar 2.6 Nilai warna RGB dalam hexadecimal
(Sumber: Perancangan)

Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimal bernilai 0) sampai ff (angka desimal bernilai 255) atau mempunyai derajat keabu-abuan $256 = 2^8$, dengan demikian range warna RGB adalah $(2^8) (2^8) (2^8) = 2^{24}$ atau dikenal dalam windows dengan istilah True Colour. Nilai warna yang digunakan merupakan gabungan warna cahaya merah, hijau, biru. Jadi untuk menyajikan warna warna tertentu dapat dengan mudah dilakukan dengan cara mencampur ketiga warna dasar RGB.



2.7 Segmentasi Citra

Pada pemrosesan citra digital, terdapat proses penting yang sering digunakan sebagai pemroses awal yang selanjutnya akan digunakan untuk proses yang lain. Proses tersebut adalah segmentasi. Segmentasi merupakan teknik untuk membagi suatu citra menjadi beberapa daerah (*region*) dimana setiap daerah memiliki kemiripan atribut.

Dengan proses segmentasi tersebut, masing-masing obyek pada gambar dapat diambil secara individu sehingga dapat digunakan sebagai input bagi proses yang lain.

Sebagai contoh, pada proses rekonstruksi obyek tiga dimensi, diperlukan proses segmentasi untuk memisahkan obyek yang akan direkonstruksi terhadap *background* yang ada. Pada proses pengenalan wajah manusia, proses segmentasi dibutuhkan untuk memisahkan wajah manusia terhadap *background* atau terhadap bagian tubuh yang lainnya sehingga didapatkan gambar wajah yang akan dikenali. Untuk proses pengenalan jenis obyek, proses segmentasi diperlukan untuk melakukan pemisahan masing-masing obyek terhadap *background* sehingga pada saat proses pengenalan, bagian *background* yang tidak diinginkan tidak ikut terproses. Itulah beberapa contoh yang membutuhkan proses segmentasi, masih banyak lagi proses-proses yang membutuhkan segmentasi. Mengingat pentingnya proses segmentasi tersebut sebagai *pre-processing*, maka dibutuhkan metode segmentasi yang dapat melakukan pemisahan obyek dengan akurat. Ketidak akuratan proses segmentasi dapat menyebabkan ketidak akuratan pada hasil proses selanjutnya.

Secara umum, proses segmentasi terbagi menjadi tiga kelompok yaitu segmentasi berdasarkan klasifikasi (*classification based segmentation*), segmentasi berdasar daerah tepi (*edge based segmentation*), dan segmentasi berdasar daerah (*region based segmentation*). Segmentasi berdasar klasifikasi adalah proses segmentasi yang dilakukan dengan mencari kesamaan ukuran dari nilai pada pixel. Segmentasi berdasar tepi adalah proses segmentasi untuk mendapatkan garis yang ada pada gambar dengan anggapan bahwa garis tersebut merupakan tepi dari obyek yang memisahkan obyek yang satu dengan obyek yang lainnya atau antara obyek dengan *background*. Segmentasi yang terakhir yaitu segmentasi berdasar daerah adalah proses segmentasi yang dilakukan untuk mendapatkan daerah yang diyakini merupakan sebuah obyek. Untuk mendapatkan daerah tersebut, dilakukan analisa terhadap kesamaan tekstur, warna pada pixel yang terdapat pada gambar.



2.7.1 Pengembangan (*Thresholding*)

Proses pengembangan akan menghasilkan citra biner, yaitu citra yang memiliki dua nilai tingkat keabuan yaitu hitam dan putih. Secara umum proses pengembangan citra *grayscale* untuk menghasilkan citra biner adalah sebagai berikut.

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \geq T \\ 0 & \text{if } f(x,y) < T \end{cases} \quad (2-2)$$

Dengan $g(x,y)$ adalah citra biner dari citra *grayscale* $f(x,y)$, dan T menyatakan nilai ambang. Nilai T memegang peranan yang sangat penting dalam proses pengembangan. Kualitas hasil citra biner sangat tergantung pada nilai T yang digunakan.

Terdapat dua jenis pengembangan, yaitu pengembangan global (*global thresholding*) dan pengembangan secara lokal adaptif (*locally adaptive thresholding*).

Pada pengembangan global, seluruh pixel pada citra dikonversikan menjadi hitam atau putih dengan satu nilai ambang T . Kemungkinan besar pada pengembangan global akan banyak informasi yang hilang karena hanya menggunakan satu nilai T untuk keseluruhan pixel. Untuk mengatasi masalah ini dapat digunakan pengembangan secara lokal adaptif. Pada pengembangan lokal, suatu citra dibagi menjadi blok-blok kecil dan kemudian dilakukan pengembangan lokal pada setiap blok dengan nilai T yang berbeda. Berikut ini beberapa metode yang dapat digunakan untuk menentukan nilai ambang (T).

2.7.2 Penandaan Komponen Terhubung (*Connected Component Labeling*)

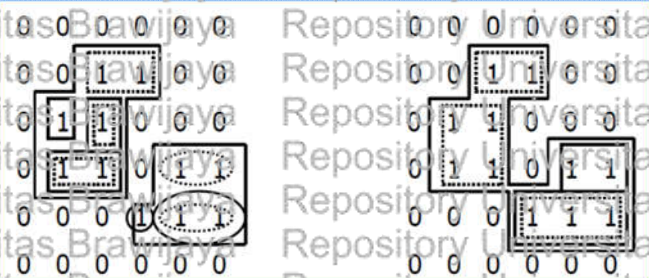
Bila setelah melewati proses segmentasi dan menemukan objek pada citra lebih dari satu maka yang penting untuk dilakukan dalam pengolahan citra seperti ini adalah menemukan komponen yang terkoneksi pada citra tersebut. Hal seperti ini didasarkan pada suatu kenyataan bahwa suatu komponen yang terkoneksi dapat merupakan bagian yang mewakili sebuah objek dalam citra yang memiliki objek lebih dari satu. Dengan cara memeriksa konektivitas dari suatu kumpulan pixel pada suatu citra biner maka kumpulan pixel ini dapat dianggap sebagai suatu objek tunggal apabila suatu kumpulan pixel ini tidak lagi tersambung dengan kumpulan pixel lainnya yang masih berada dalam suatu citra biner.

Penandaan komponen dilakukan dengan memeriksa suatu citra, dengan cara pixel per pixel diperiksa dari kiri ke kanan dan dari atas ke bawah hal ini dilakukan



untuk mengidentifikasi area pixel yang terhubung yaitu suatu area dari pixel berbatasan yang memiliki nilai intensitas sama atau berada dalam himpunan V bila dalam citra biner $V=1$. Penandaan komponen terhubung dapat dilakukan pada citra biner maupun citra *grayscale*.

Setelah melakukan operasi pelabelan pada komponen terhubung pada suatu citra maka dapat menghitung ciri-ciri mendasar pada masing-masing objek. Tanpa pelabelan maka perhitungan ciri-ciri dari objek-objek yang ada akan menjadi sulit sebab hasil yang didapat merupakan hasil gabungan dari semua objek yang ada pada suatu citra. Dengan demikian pelabelan komponen seperti pada gambar dibawah ini



Gambar 2.7 Ilustrasi komponen terhubung

(Sumber : Bertalya. Representasi Citra. 2005)

Hanya digunakan bila pada suatu citra biner terdapat lebih dari satu objek sebab pelabelan komponen terhubung dilakukan untuk membedakan objek yang satu dengan objek yang lainnya tetapi masih berada dalam satu citra.

2.7.3 Segmentasi berbasis *Clustering*

Segmentasi berbasis *cluster* menggunakan data multidimensi untuk mengelompokkan pixel citra ke dalam beberapa *cluster*. Pada umumnya pixel di-*cluster* berdasarkan kedekatan jarak antar pixel.

Segmentasi berbasis *cluster* ini mulai populer sejak diimplementasikan pada aplikasi OCR (*Optical Character Recognition*), pengenalan sidik jari hingga *remote sensing*. Keberhasilan dari proses segmentasi berbasis *cluster* ditentukan dari keberhasilan dalam mengelompokkan fitur-fitur yang berdekatan ke dalam satu *cluster*.

Metode-metode dalam segmentasi berbasis *cluster* di antaranya adalah iterasi, *K-means*, *fuzzy C-means*, jaringan syaraf Kohonen, dan berbagai trik *cluster* lainnya.



Metode Iterasi

Metode iterasi adalah bentuk khusus dari *K-means* di mana $K=2$. Metode iterasi dimulai dengan memilih nilai batas (*threshold*) secara sembarang sebagai nilai awal, lalu secara iterasi nilai tersebut diperbaiki berdasarkan sebaran nilai intensitas citra yang bersangkutan. Nilai *threshold* yang baru diharapkan dapat menghasilkan pemisahan yang lebih baik dari citra sebelumnya. Langkah-langkah dalam menentukan nilai batas T dalam metode iterasi adalah sebagai berikut.

1. Pilih nilai T awal, biasanya dipakai rata-rata dari intensitas citra.
2. Segmentasi citra menjadi dua daerah, misalnya R_1 dan R_2 dengan menggunakan nilai T awal sebelumnya.
3. Hitung nilai rata-rata intensitas pada daerah R_1 dan R_2 . Kedua nilai rata-rata tersebut berturut-turut disebut r_1 dan r_2 .
4. Hitung nilai T baru dengan rumus $T=(r_1 + r_2)/2$.
5. Ulang langkah 2 sampai 4 hingga nilai T tercapai. Nilai T dikatakan telah tercapai bila nilai T tidak mengalami perubahan nilai T lagi.

2.8 Morfologi Citra

Arti kata morfologi secara sederhana adalah bentuk dan struktur suatu objek atau dalam deskripsi yang lainnya dijelaskan bahwa morfologi adalah susunan dan hubungan antar bagian dari suatu objek. Di dalam dunia digital dapat diartikan sebuah cara untuk mendeskripsikan ataupun menganalisa bentuk dari suatu objek. Pada morfologi suatu citra dinyatakan sebagai himpunan diskrit atau kontinu. Himpunan tersebut berhubungan dengan point atau pixel objek pada suatu citra karena itu objek pada suatu citra dianggap sebagai himpunan.

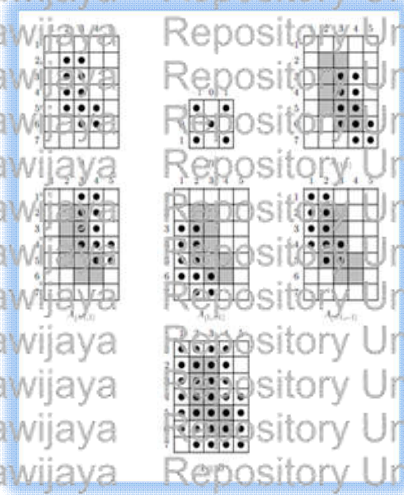
Pada operasi morfologi menggunakan dua input himpunan yaitu suatu citra dalam hal ini pada umumnya digunakan citra biner dan suatu kernel. Khusus dalam morfologi kernel disebut juga sebagai *structuring elements* (elemen pembentuk struktur). SE merupakan suatu matrik dan pada umumnya matrik berukuran kecil. Terdapat beberapa operasi morfologi erosi, dilasi, opening, closing.

2.8.1 Dilasi

Proses dilasi dilakukan dengan membandingkan setiap pixel citra input dengan nilai pusat SE dengan cara melapiskan SE dengan citra sehingga SE tepat pada posisi pixel citra yang diproses. Jika paling sedikit ada 1 pixel pada SE sama dengan nilai



pixel pada objek (*foreground*) citra maka pixel input diset nilainya dengan nilai pixel *foreground* dan bila semua pixel yang berhubungan adalah *background* maka input pixel diberi nilai pixel *background*.



Gambar 2.8 Dilasi

(Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab, 2004)

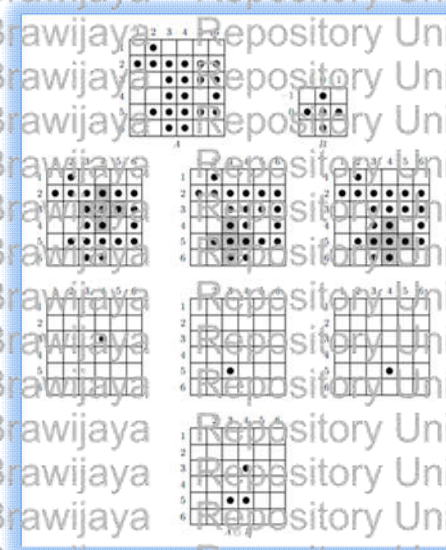
Semakin besar ukuran SE maka semakin besar perubahan yang terjadi. Bila SE berukuran kecil juga dapat memberikan hasil yang sama dengan SE yang berukuran besar jika dilakukan proses dilasi berulang-ulang.

Pada citra biner dilasi dapat memberikan efek memperbesar batas dari objek yang ada sehingga objek terlihat semakin besar dan lubang-lubang yang berada dibagian tengah objek akan nampak mengecil atau menghilang.

2.8.2 Erosi

Erosi merupakan kebalikan dari proses dilasi bila pada proses dilasi menghasilkan objek yang lebih besar dan lebih luas maka dalam proses erosi akan menghasilkan objek yang mengecil dengan demikian lubang-lubang yang berada ditengah objek juga akan terlihat lebih membesar.

Semakin besar SE yang digunakan maka hasil yang didapatkan akan semakin kecil begitu juga bila proses erosi dilakukan berulang kali akan terus membuat objek semakin terlihat mengecil walaupun dalam hal ini yang digunakan adalah SE berukuran kecil.

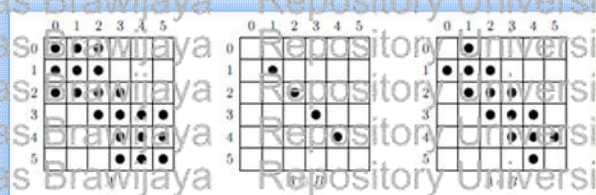


Gambar 2.9 Erosi

(Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab, 2004)

2.8.3 Opening

Proses opening merupakan gabungan proses erosi dan dilasi dimana dilakukan proses erosi selanjutnya dilakukan proses dilasi. Pada operasi opening hal ini dilakukan karena pada saat dilakukan proses erosi maka objek yang ada akan mengalami penurunan ukuran citra biner untuk mengatasi hal tersebut maka dilakukan proses dilasi dengan menggunakan ukuran SE yang sama. Secara matematis operasi opening dapat dituliskan :

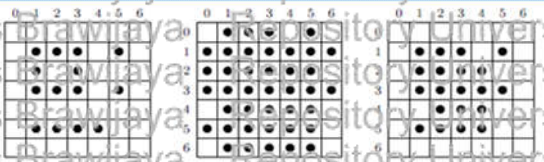


Gambar 2.10 Opening

(Sumber : Alasdair McAndrew. An Introduction to Digital Image Processing with Matlab, 2004)

2.8.4 Closing

Proses closing merupakan kebalikan dari proses opening dimana pada proses ini dilasi dilakukan terlebih dahulu proses dilasi kemudian baru erosi. Secara matematis operasi closing dapat dinyatakan dengan



Gambar 2.11 Closing

(Sumber : Alasdair McAndrew. An Introduction to Digital Image

Processing with Matlab, 2004)

Hasil operasi closing hampir mirip dengan operasi dilasi yaitu memperbesar batas luar dari objek dan juga menutup lubang ditengah objek akan tetapi hasil operasi closing tidak sebesar hasil pada dilasi hal ini disebabkan karena proses erosi yang dilakukan setelah proses dilasi.

2.9 Ekstraksi Fitur

Ekstraksi fitur merupakan pengambilan fitur atau ciri pada objek melalui proses tertentu. Ciri atau fitur merupakan karakteristik unik dari suatu objek. Karakteristik fitur yang baik sebisa mungkin memenuhi persyaratan berikut.

- Dapat membedakan suatu objek dengan yang lainnya.
- Memperhatikan kompleksitas komputasi dalam memperoleh fitur. Kompleksitas yang tinggi akan menjadi beban tersendiri dalam menemukan fitur.
- Tidak terikat dalam arti bersifat invarian terhadap berbagai transformasi.
- Jumlahnya sedikit.

Ekstraksi fitur dikategorikan menjadi 3 level, yaitu level rendah (*low-level*), level medium (*middle-level*), dan level tinggi (*high-level*). *Low-level* fitur merupakan ekstraksi ciri berdasarkan fitur visual seperti warna dan tekstur. *Middle-level* fitur merupakan ekstraksi berdasarkan wilayah citra yang ditentukan dengan segmentasi. *High-level* fitur merupakan ekstraksi ciri berdasarkan informasi semantik yang terkandung dalam citra. Untuk memperoleh suatu fitur atau ciri dari citra diperlukan metode analisis citra. Berbagai metode ekstraksi fitur suatu citra dijelaskan sebagai berikut.

2.9.1 Histogram

Ciri histogram didasarkan pada histogram dari suatu citra. Bila x menyatakan tingkat keabuan pada suatu citra maka probabilitas dari x dinyatakan dengan



$$P(x) = \frac{\text{Banyaknya titik-titik yang memiliki tingkat } x}{\text{Total banyaknya titik pada daerah suatu}} \quad (2-3)$$

2.9.2. Deteksi tepi

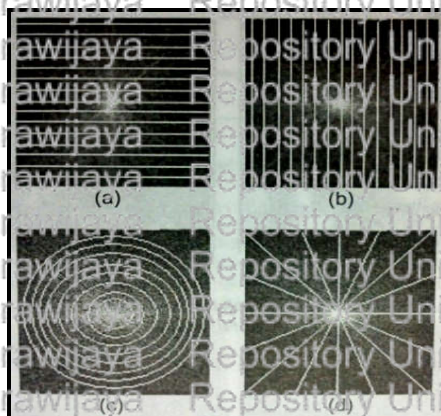
Banyaknya tepi pada pixel tetangga dapat diukur sebagai suatu fitur tekstur. Untuk mengukur ciri tekstur ini, suatu citra harus mengalami proses deteksi tepi. Citra deteksi tepi dapat diperoleh dengan menggunakan metode deteksi tepi seperti, operator robert, sobel, prewit, krishch dan canny. Ciri tekstur pada suatu daerah citra yang berukuran $W \times W$ didefinisikan sebagai berikut.

$$T_{i,k} = \frac{1}{W} \sum_{m=-w}^w \sum_{n=-w}^w \quad (2-4)$$

$E_{i,k}$ merupakan citra hasil proses deteksi tepi.

2.9.3. Spektrum fourier

Transformasi fourier membawa suatu citra dari domain spasial $f(x,y)$ ke domain frekuensi $F(p,q)$. Beberapa fitur penting yang dapat diperoleh dari spektrum fourier adalah celah horisontal, celah vertikal, cincin, dan sektor seperti yang ditunjukkan gambar 2.12



Gambar 2.12 Ciri spektrum Fourier

(a) celah horisontal (b) celah vertikal (c) cincin (d) sektor

(Sumber: Darma Putra, 2010)

2.9.4. Fitur berdasarkan warna



Metode yang dapat digunakan untuk mendapatkan ciri berdasarkan warna adalah dengan menggunakan *mean* warna dan histogram warna. *Mean* warna adalah nilai rata-rata kanal dari suatu objek. *Mean* dihitung pada setiap piksel untuk masing-masing nilai kanal piksel. Berikut ini merupakan mean warna kanal RGB.

$$r_{avg} = \frac{\sum_{i=1}^N r(i)}{N}, g_{avg} = \frac{\sum_{i=1}^N g(i)}{N}, b_{avg} = \frac{\sum_{i=1}^N b(i)}{N} \quad (2-5)$$

dengan:

$r_{avg}, g_{avg}, b_{avg}$ = rata-rata intensitas warna R, G, B

$r(i), g(i), b(i)$ = nilai intensitas warna R, G, B pada suatu pixel

N = total pixel pada objek.

Histogram warna merupakan fitur yang juga dapat digunakan untuk merepresentasikan ciri warna suatu citra. Citra pada umumnya dikonversi ke dalam suatu ruang warna tertentu, kemudian setiap komponen ruang warna dibuat histogramnya.

2.9.5 Tapis gabor

Tapis gabor merupakan metode yang handal untuk menghasilkan fitur dalam hal biometrika seperti fingercode dan iriscode. Pada tapis gabor 2D terdapat beberapa parameter yang berpengaruh terhadap hasil ekstraksi fitur antara lain ukuran kernel, parameter θ yang mengontrol orientasi dari tapis gabor, parameter μ , dan parameter standar deviasi yang menyesuaikan dengan ukuran tapis.

2.9.6 Analisis Bentuk Atribut Geometri

Salah satu obyek yang memiliki bentuk yang sangat sederhana adalah lingkaran. Di samping sederhana, lingkaran memiliki bentuk yang sangat simetris. Seperti diketahui, luas dan keliling sebuah lingkaran dapat diketahui dengan:

$$P = 2\pi R \quad (2-6)$$

$$A = \pi R^2 \quad (2-7)$$

dimana R adalah jari-jari dan D merupakan garis tengah lingkaran. Komputasi untuk kedua properti lingkaran ini dapat dilakukan pada region-region citra, dan ini merupakan bentuk dasar dari ukuran kebulatan (circularity). Rasio P^2/A untuk



sebuah lingkaran adalah 4π , dan ini merupakan nilai minimum untuk setiap region.

Dari sini, dapat dirumuskan ukuran kebundaran obyek adalah:

$$C = \frac{p^2}{4\pi A} \quad (2-8)$$

C memiliki nilai minimum 1 untuk lingkaran dan terus menaik untuk region yang lebih kompleks.



Gambar 2.13 Bounding Circle

(Sumber: Lukman Talibo, 2004)

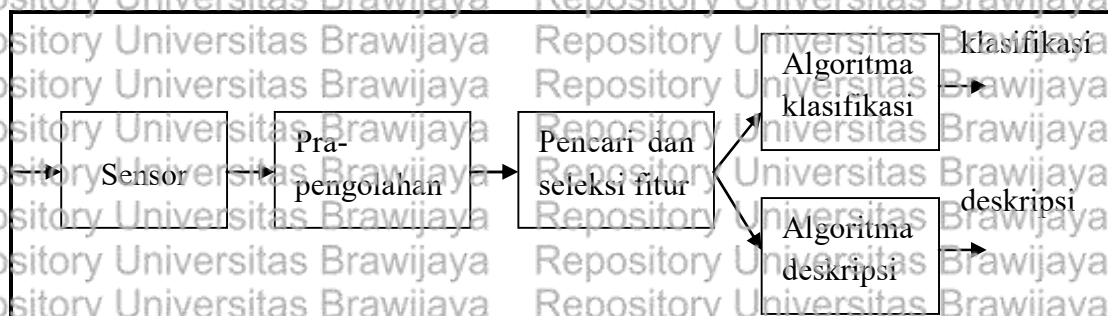
Untuk mendapatkan nilai *bounding circle* dilakukan pendekatan dengan memberikan nilai maksimum 1 untuk lingkaran dan nilai yang terus menurun untuk obyek yang lebih kompleks. Nilai ini dapat mewakili *bounding circle*. Perbandingan antara luas suatu lingkaran dengan luas lingkaran terkecil yang dapat menutupi lingkaran tersebut adalah 1.

$$C = \frac{1}{\left(\frac{P^2}{4\pi A}\right)} = \frac{4\pi A}{P^2} \quad (2-9)$$

2.10 Pengenalan Pola

Pengenalan Pola adalah mengelompokkan data numerik dan simbolik (termasuk citra) secara otomatis oleh mesin (dalam hal ini komputer). Tujuan pengelompokan adalah untuk mengenali suatu objek di dalam citra. Manusia bisa mengenali objek yang dilihatnya karena otak manusia telah belajar mengklasifikasi objek-objek di alam sehingga mampu membedakan suatu objek dengan objek lainnya. Kemampuan sistem visual manusia inilah yang dicoba ditiru oleh mesin. Komputer menerima masukan berupa citra objek yang akan diidentifikasi, memproses citra tersebut, dan memberikan keluaran berupa deskripsi objek di dalam citra.

Struktur dari sistem pengenalan pola terdiri dari sensor, algoritma atau mekanisme pencari fitur, dan algoritma untuk klasifikasi atau pengenalan seperti yang ditunjukkan oleh gambar 2.14.



Gambar 2.14 Struktur sistem pengenalan pola

(Sumber: Darma Putra, 2010)

Sensor berfungsi untuk menangkap objek dan mengubah menjadi sinyal digital melalui digitalisasi. Pra-pengolahan berfungsi untuk mempersiapkan citra agar dapat menghasilkan ciri yang baik. Pencari dan seleksi fitur berfungsi untuk menemukan karakteristik pembeda yang mewakili sifat utama sinyal dan sekaligus mengurangi dimensi sinyal menjadi sekumpulan bilangan yang sedikit tetapi representatif.

Algoritma klasifikasi berfungsi untuk mengelompokkan fitur ke dalam kelas yang sesuai. Algoritma deskripsi berfungsi memberikan deskripsi pada sinyal.

Fitur atau disebut juga atribut adalah semua hasil pengukuran yang bisa diperoleh dan merupakan karakteristik pembeda dari objek fitur. Fitur dapat berupa simbol seperti warna, numerik seperti berat, atau gabungan keduanya. Fitur dapat dinyatakan dengan variabel kontinu, diskret, atau diskret-biner.

Vektor fitur adalah gabungan atau kombinasi dari beberapa fitur dan dinyatakan sebagai vektor kolom. Ruang fitur adalah ruang yang dibentuk oleh vektor fitur dan merupakan cara untuk memvisualisasikan distribusi vektor fitur. Scatter plot merupakan alat visualisasi untuk menentukan distribusi dari distribusi vektor bila dimensi vektor kurang dari sama dengan 3. Pola dapat dikatakan sama dengan fitur atau vektor fitur yang merupakan sifat utama dari suatu objek. Namun dalam pengenalan (klasifikasi), pola merupakan sepasang variabel (x, ω) dengan x menyatakan sekumpulan pengamatan atau fitur atau vektor fitur dan ω merupakan konsep dibalik pengamatan.

Pemilah merupakan teknik atau metode untuk mengelompokkan vektor fitur ke dalam kelas-kelas tertentu. Vektor fitur untuk pelatihan merupakan vektor fitur yang telah diketahui kelasnya dan digunakan untuk merancang pemilah.



2.11 Jaringan Saraf Tiruan

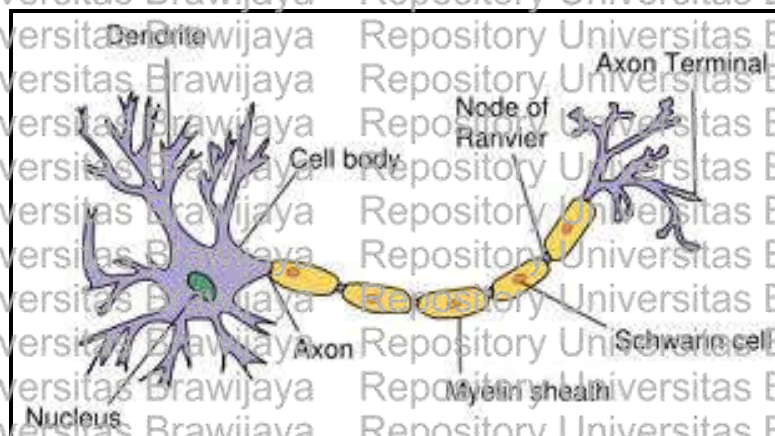
Jaringan saraf tiruan atau *Artificial Neural Network* yang sering disingkat dengan ANN adalah sistem pemroses informasi yang memiliki karakteristik kinerja tertentu yang sama dengan jaringan saraf biologis. ANN telah dikembangkan sebagai generalisasi model matematika suatu saraf biologis manusia, berdasarkan asumsi bahwa:

- Pengolahan informasi terjadi pada elemen sederhana yang disebut neuron.
- Sinyal antar neuron dilewatkan melalui suatu hubungan.
- Setiap hubungan memiliki berat yang terkait, yang dalam jaringan saraf, mengalikan sinyal yang ditransmisikan.
- Setiap neuron menerapkan fungsi aktivasi terhadap input jaringan untuk menentukan sinyal outputnya.

Sebuah jaringan saraf ditandai oleh (1) polanya koneksi antara neuron (disebut arsitektur), (2) metode menentukan bobot (disebut pelatihan, atau belajar, algoritma), dan (3) fungsi aktivasi.

2.11.1 Model Neuron Biologis

Pada dasarnya jaringan saraf atau *neural network* terdiri atas banyak elemen pemroses sederhana yang disebut neuron, sel unit, atau simpul. Sebagai bahan perbandingan, otak seekor cacing diperkirakan memiliki 1.000 neuron dan otak manusia memiliki sekitar 100 miliar. Setiap sel saraf berhubungan dengan sel saraf lainnya memakai saluran komunikasi yang teratur dengan suatu bobot penghubung. Gambar 2.7 berikut menunjukkan gambar struktur neuron pada otak manusia.



Gambar 2.15 Struktur neuron



(Sumber: Darma Putra, 2010)

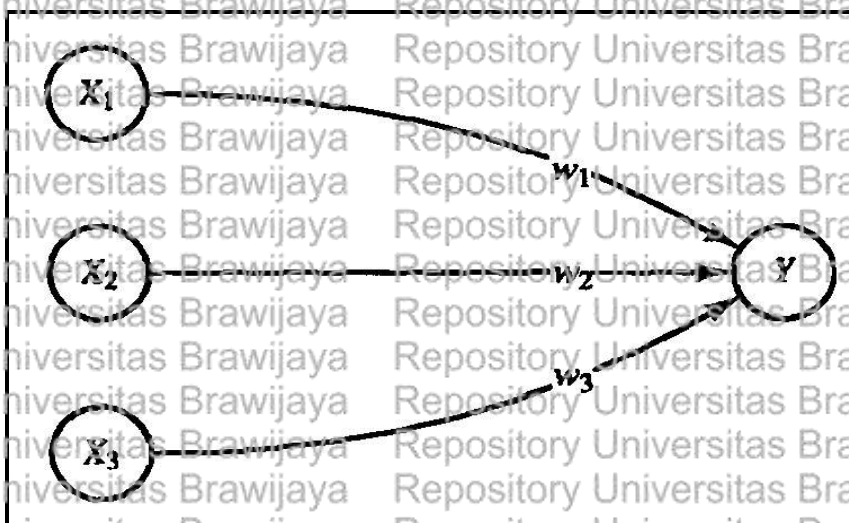
2.11.2 Model dan konsep dasar ANN

Sebuah jaringan saraf terdiri dari sejumlah elemen pemroses sederhana yang disebut neuron, unit, sel, atau node. Setiap neuron terhubung ke neuron lain melalui jaringan komunikasi terarah, masing-masing dengan bobot yang terkait. Bobot tersebut merupakan informasi yang digunakan oleh jaringan untuk memecahkan masalah. Jaringan saraf dapat diterapkan pada berbagai macam masalah, seperti menyimpan dan mengingat data atau pola, mengklasifikasi pola, melakukan pemetaan umum dari pola input ke pola output, pengelompokan pola yang sama, atau mencari solusi bagi permasalahan optimasi yang dibatasi.

Setiap neuron memiliki keadaan internal, disebut aktivasi atau tingkat aktivitas, yang mana fungsi dari input telah diterima. Biasanya, neuron mengirimkan aktivasi sebagai sinyal ke neuron lainnya. Penting untuk dicatat bahwa neuron dapat mengirim hanya satu sinyal pada satu waktu, meskipun sinyal tersebut disebarkan ke neuron lainnya.

Sebagai contoh, suatu neuron Y, diilustrasikan pada Gambar 2.16, menerima masukan dari neuron X_1 , X_2 , dan X_3 . Aktivasi (sinyal output) dari neuron tersebut masing-masing x_1 , x_2 , dan x_3 . Bobot pada koneksi dari X_1 , X_2 , dan X_3 ke neuron Y masing-masing w_1 untuk w_2 , dan w_3 . Input jaringan y_{in} ke neuron Y adalah jumlah dari sinyal berbobot dari neuron X_1 , X_2 , dan X_3 yaitu,

$$y_{in} = w_1 x_1 + w_2 x_2 + w_3 x_3$$



Gambar 2.16 Neuron sederhana



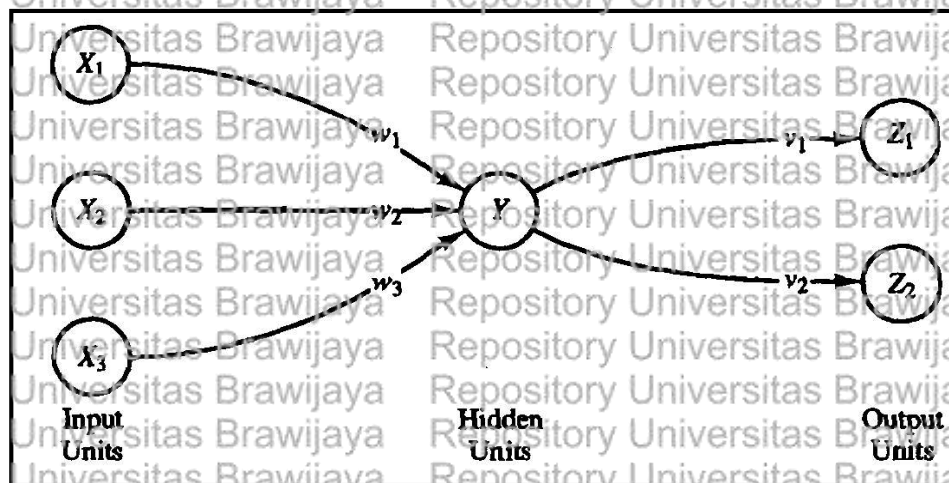
(Sumber: Laurene Fausett, 1993)

Aktivasi y dari neuron Y diberikan oleh beberapa fungsi jaringan input tersebut $y=f(y_{in})$, misalnya fungsi sigmoid atau salah satu dari sejumlah fungsi aktivasi lainnya.

Sekarang anggaplah lebih lanjut bahwa neuron Y terhubung ke neuron Z_1 dan Z_2 , dengan masing-masing bobot v_1 dan v_2 , seperti yang ditunjukkan pada Gambar 2.17.

Neuron Y mengirimkan sinyal y untuk masing-masing unit. Namun, secara umum, nilai-nilai yang diterima oleh neuron Z_1 dan Z_2 akan berbeda, karena setiap sinyal diskalakan oleh bobot v_1 dan v_2 . Dalam jaringan yang khas, aktivasi z_1 dan z_2 neuron Z_1 dan Z_2 akan tergantung pada masukan dari beberapa neuron.

Meskipun jaringan saraf pada gambar 2.17 adalah sangat sederhana, namun adanya unit tersembunyi, bersama dengan fungsi aktivasi non-linear, memberikan kemampuan untuk memecahkan banyak masalah yang lebih baik daripada yang dapat diselesaikan dengan jaringan hanya dengan input dan output unit. Di sisi lain, lebih sulit untuk melatih (yaitu, menemukan nilai-nilai optimal untuk bobot) jaringan dengan unit tersembunyi.



Gambar 2.17 Jaringan saraf tiruan sederhana

(Sumber: Laurene Fausett, 1993)

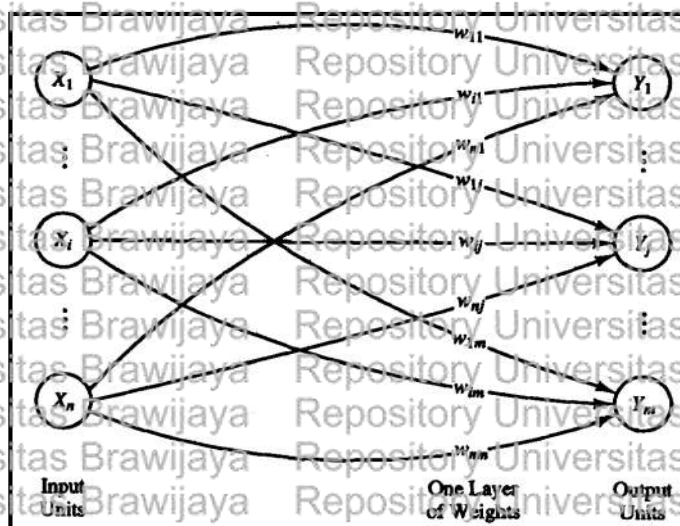
2.11.3 Arsitektur ANN

ANN memiliki beberapa arsitektur yang sering dipakai dalam berbagai aplikasi. Arsitektur tersebut antara lain:

2.11.3.1 Jaringan lapisan tunggal (single layer network)



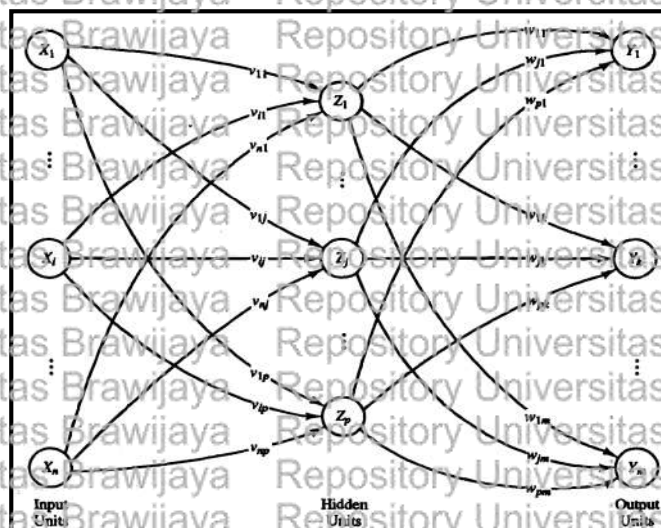
Jaringan dengan lapisan tunggal terdiri dari 1 *layer* input dan 1 *layer* output. Setiap neuron/unit yang terdapat di dalam lapisan input selalu terhubung dengan setiap neuron yang terdapat pada *layer* output. Jaringan ini hanya menerima input kemudian secara langsung akan mengolannya menjadi output tanpa harus melalui lapisan tersembunyi.



Gambar 2.18 Arsitektur lapisan tunggal
(Sumber: Laurene Fausett, 1993)

2.11.3.2 Jaringan lapisan jamak (*multi layer network*)

Jaringan dengan lapisan jamak memiliki ciri khas tertentu yaitu memiliki 3 jenis *layer* yakni *layer* input, *layer* output dan *layer* tersembunyi. Jaringan dengan lapisan banyak ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan dengan jaringan lapisan tunggal. Namun proses pelatihan cenderung membutuhkan waktu yang lebih lama.



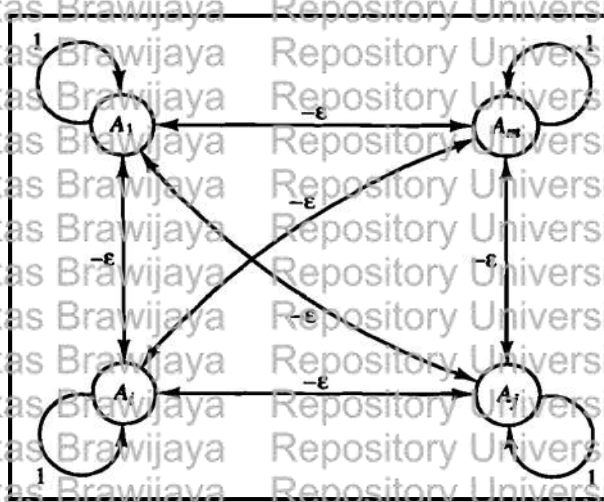


Gambar 2.19 Arsitektur lapisan jamak

(Sumber: Laurene Fausett, 1993)

2.11.3.3 Jaringan dengan lapisan kompetitif (*competitive layer network*)

Pada jaringan ini, sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Lapisan kompetitif merupakan bagian dari sejumlah jaringan saraf yang besar. Biasanya, interkoneksi antara neuron pada lapisan kompetitif tidak ditampilkan dalam arsitektur diagram untuk jaringan tersebut.



Gambar 2.20 Arsitektur lapisan kompetitif

(Sumber: Laurene Fausett, 1993)

2.11.4 Pengaturan bobot

Selain arsitektur, metode pengaturan nilai-nilai bobot (pelatihan) adalah ciri khas penting dari suatu jaringan saraf tiruan. Terdapat dua jenis pelatihan yaitu pelatihan yang diawasi dan tidak diawasi, di samping itu ada jaringan yang bobotnya adalah tetap tanpa proses berulang-ulang pelatihan. Klasifikasi dan asosiasi pola dapat dianggap sebagai bentuk khusus dari permasalahan umum pemetaan pola atau vektor input ke pola atau vector output yang lebih spesifik.

a. *Supervised learning* (pembelajaran terawasi)

Pada metode ini, setiap pola yang diberikan kedalam ANN telah diketahui outputnya. Selisih antara pola output aktual (output yang dihasilkan) dengan pola output target (output yang dikehendaki) yang disebut eror digunakan untuk mengoreksi bobot ANN sehingga ANN mampu untuk menghasilkan output sedekat mungkin dengan pola target yang telah diketahui oleh ANN.

b. *Unsupervised learning* (pembelajaran tak terawasi)



Pada metode ini tidak memerlukan target output. Pada metode ini tidak dapat ditentukan hasil seperti apakah yang diharapkan selama proses pembelajaran. Selama proses pembelajaran, nilai bobot disusun dalam *range* tertentu tergantung pada nilai input yang diberikan. Tujuan dari pembelajaran ini adalah mengelompokkan unit-unit yang hampir sama dalam suatu area tertentu. Pembelajaran ini biasanya sangat cocok dalam klasifikasi pola.

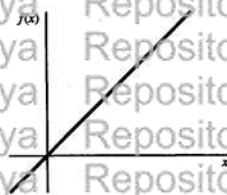
2.11.5 Fungsi aktivasi umum

Seperti disebutkan sebelumnya, operasi dasar dari sebuah neuron buatan melibatkan penjumlahan sinyal input berbobot dan menerapkan output, atau aktivasi, fungsi

a. Fungsi identitas (linier)

$$f(x) = x \text{ for all } x$$

Fungsi linier memiliki nilai output yang sama dengan nilai inputnya.



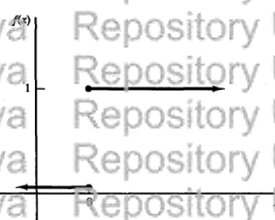
Gambar 2.21 Fungsi identitas

(Sumber: Laurene Fausett, 1993)

b. Fungsi biner dengan threshold θ

$$f(x) = \begin{cases} 1 & \text{if } x \geq \theta \\ 0 & \text{if } x < \theta \end{cases}$$

Jaringan layer tunggal sering menggunakan fungsi step untuk mengkonversi input, yang mana nilai variabelnya *continue*, ke unit output yang merupakan sinyal biner (1 atau 0) atau bipolar (1 atau -1). Fungsi biner juga dikenal sebagai fungsi *threshold*.



Gambar 2.22 Fungsi threshold



(Sumber: Laurene Fausett, 1993)

2.11.6 Learning rate

Dalam pemilihan learning rate terdapat beberapa pertimbangan. Pelatihan yang lambat membutuhkan waktu yang lebih lama untuk menghasilkan sistem yang terlatih. Pelatihan yang cepat memungkinkan waktu yang singkat akan tetapi jaringan belum tentu menghasilkan sistem yang terlatih seperti pada pelatihan yang lambat. Fungsi pelatihan memiliki beberapa ketentuan untuk nilai learning rate. Secara umum bernilai antara 0 sampai positif 1. Bila learning rate-nya kecil, maka pencapaian nilai konvergensinya lambat dan akan menyebabkan lambat dalam mempelajari kesalahan, sedangkan bila learning rate-nya besar, pencapaian nilai konvergensinya cepat namun ada bahaya osilasi yang dapat mengakibatkan nilai minimum tidak tercapai. Maka perlu digunakan variasi lagi sebagai berikut :

- Variable Learning Rate

Metode ini bekerja dengan berusaha menaikkan learning rate bila menjumpai permukaan yang datar dan kemudian menurunkan learning rate bila terjadi peningkatan slope.

- Modifikasi parameter learning rate.

Besarnya learning rate akan selalu update serta menerapkan teknik pembelajaran pada jaringan yang harus direncanakan. Semisal nilai learning rate 0,1 dan maksimum epoch 100. Kemudian jaringan perlu dilatih kembali dengan meningkatkan jumlah neuron hidden layer. Pada pemrosesan tiap neuron diperlukan pengupdatean Learning rate.

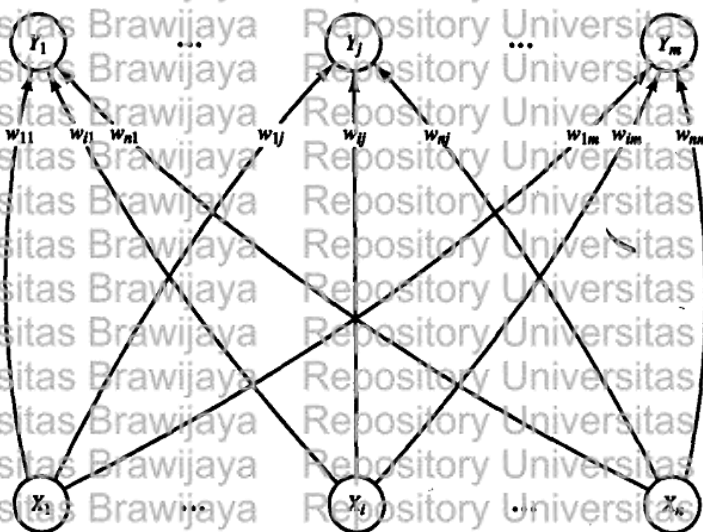
2.11.7 Learning Vector Quantization (LVQ)

Learning Vector Quantization (LVQ) [Kohonen, 1989a, 1990a] adalah sebuah metode klasifikasi dimana setiap unit output mewakili suatu kelas atau kategori. Vektor bobot suatu unit output sering disebut sebagai vektor referensi suatu kelas pada suatu unit. Selama pelatihan, unit keluaran diposisikan dengan menyesuaikan bobot mereka melalui pelatihan diawasi.

LVQ adalah salah satu jaringan saraf tiruan yang merupakan algoritma pembelajaran kompetitif terawasi versi dari algoritma *Kohonen Self-Organizing Map* (SOM). Tujuan dari algoritma ini adalah untuk mendekati distribusi kelas vektor untuk



meminimalkan kesalahan dalam pengklasifikasian. Arsitektur jaringan LVQ ini terdiri dari input, lapisan kohonen, dan lapisan output, seperti pada gambar 2.25



Gambar 2.25 Arsitektur jaringan LVQ

(Sumber: Laurene Fausett, 1993)

Algoritma jaringan LVQ mencari unit output yang terdekat dengan unit input. Jika x dan w berada pada kelas yang sama, maka bobot vektor dipindahkan mendekati input vektor yang baru. Jika x dan w merupakan kelas yang berbeda, maka bobot vektor dipindahkan menjauhi input vektor. Algoritma LVQ adalah sebagai berikut:

Langkah 0. Inisialisasi bobot awal, *learning rate*, $\alpha(0)$.

Langkah 1. Selama kondisi berhenti adalah salah, kerjakan langkah 2-6.

Langkah 2. Untuk setiap input pelatihan vektor x , lakukan langkah 3-4

Langkah 3. Temukan J agar $\|x - w_j\|$ adalah minimum

Langkah 4. Perbaiki w_j dengan ketentuan:

Jika $T = C_j$ maka:

$$w_j(\text{baru}) = w_j(\text{lama}) + \alpha[x - w_j(\text{lama})]$$

Jika $T \neq C_j$ maka:

$$w_j(\text{baru}) = w_j(\text{lama}) - \alpha[x - w_j(\text{lama})]$$

Langkah 5. Kurangi *learning rate*.

Langkah 6. Uji kondisi berhenti:

Kondisi ini dapat ditentukan dengan menetapkan jumlah eksekusi langkah 1 atau saat *learning rate* mencapai nilai yang cukup kecil.



Keterangan

x	vektor pelatihan (x_1, \dots, x_i).
T	kategori atau kelas yang tepat untuk vektor pelatihan.
w_j	bobot vektor untuk unit output J ($w_{j1}, \dots, w_{jI}, w_{jJ}$).
C_j	kategori atau kelas yang diwakili oleh unit output J .
$\ x - w_j\ $	Euclidean distance antara vektor input dan bobot vektor untuk unit output ke- J .

Terdapat beberapa metode dalam penentuan bobot awal dalam LVQ. Metode pertama adalah mengambil data vektor pelatihan pertama tiap kelas kemudian menggunakannya sebagai bobot vektor. Metode kedua adalah menggunakan K-means clustering dan ketiga adalah menggunakan *self-organizing map*.

2.11.8 Penentuan bobot

Jaringan saraf terdiri dari elemen pemroses sederhana yang dinamakan neuron, unit, sel atau node. Setiap neuron terkoneksi dengan neuron yang lain masing-masing dengan bobot terhubung. Bobot merepresentasikan informasi yang digunakan oleh jaringan untuk menyelesaikan masalah. Jaringan saraf dapat diaplikasikan untuk masalah yang sangat luas, seperti penyimpanan dan pemanggilan kembali data atau pola, mengklasifikasikan pola, menampilkan pemetaan secara umum dari pola input menjadi pola output, mengelompokkan pola yang sama, atau menemukan solusi untuk mengoptimisasikan masalah.

Nilai bobot memiliki rentang antara negatif tak terhingga ($-\infty$) sampai positif tak terhingga (∞), maka dari itu untuk hasil yang terbaik sangat disarankan jika kita menentukan bobot awal menggunakan variabel yang netral atau nilai yang mendekati dengan kondisi manapun yaitu nol (0).

2.12 Euclidean Distance

Jarak digunakan untuk menentukan tingkat kesamaan atau ketidaksamaan dua vektor fitur. Tingkat kesamaan berupa suatu nilai dan berdasarkan nilai tersebut dua



vektor fitur akan dikatakan mirip atau tidak. Salah satu metode menghitung jarak dua vektor fitur yaitu *euclidean distance*.

Euclidean distance adalah pengukuran yang sering digunakan untuk menghitung kesamaan 2 vektor. *Euclidean distance* menghitung akar dari kuadrat dua vektor. Rumusnya adalah sebagai berikut:

$$d_{ij} = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2}$$

$$= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + (x_{i3} - x_{j3})^2 + \dots + (x_{in} - x_{jn})^2} \quad (2-10)$$

dengan:

d_{ij} = euclidean distance vektor i dan j

k = jumlah variabel vektor (1,2,3,...,n)

x_{ik} = nilai variabel vektor i

x_{jk} = nilai variabel vektor j

2.13 Basis Data

2.13.1 Pengertian Basis Data

Basis data adalah sebuah cara mendokumentasikan berbagai macam data yang kemudian dimanajemen dengan sebuah sistem untuk kemudian disimpan dalam sebuah media penyimpanan. Dengan demikian data-data tersebut dapat diakses dengan mudah dan cepat. Media penyimpanan tersebut dapat kita ibaratkan sebuah *storage* penyimpanan, misalnya hardisk. Dalam basisdata, data yang ada tidak hanya diletakkan dan disimpan begitu saja dalam sebuah media penyimpanan, akan tetapi dikelola dengan sebuah sistem pengaturan basisdata yang seiring disebut dengan *Database Management System (DBMS)*. Dengan begitu, suatu data dengan jumlah besar dan kompleks dapat tersusun sangat baik sehingga memungkinkan pengaksesan data dengan mudah dan cepat oleh pengguna. Basisdata merupakan sekumpulan informasi yang sangat kompleks yang berguna untuk mengatur semua data yang ada di dalamnya sehingga dapat diakses oleh pengguna dengan mudah dan cepat.



2.13.2 Elemen Basis Data

1) Entitas

Adalah sekumpulan objek yang terdefinisikan yang mempunyai karakteristik sama dan bisa dibedakan satu dengan lainnya. Objek dapat berupa barang, orang, tempat atau suatu kejadian.

2) Atribut

Adalah deskripsi data yang bisa mengidentifikasi entitas yang membedakan entitas tersebut dengan entitas yang lain. Seluruh atribut harus cukup untuk menyatakan identitas obyek, atau dengan kata lain, kumpulan atribut dari setiap entitas dapat mengidentifikasi keunikan suatu individu.

3) Data Value (*Nilai Data*)

Data Value adalah data aktual atau informasi yang disimpan pada tiap data, elemen, atau atribut. Atribut nama pegawai menunjukkan tempat dimana informasi nama karyawan disimpan, nilai datanya misalnya adalah Anjang, Arif, Suryo, dan lain-lain yang merupakan isi data nama pegawai tersebut.

4) File/Tabel

Kumpulan record sejenis yang mempunyai panjang elemen yang sama, atribut yang sama, namun berbeda nilai datanya.

5) Record/Tuple

Kumpulan elemen-elemen yang saling berkaitan menginformasikan tentang suatu entitas secara lengkap. Satu record mewakili satu data atau informasi.

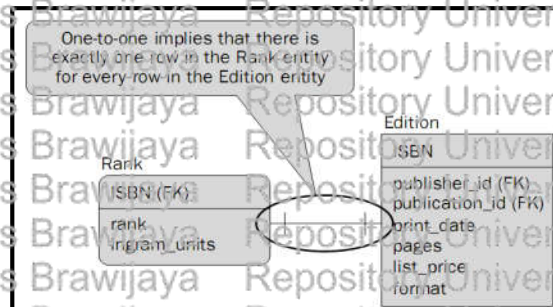
2.13.3 Relational Database Model

Relational Database Model merupakan salah satu jenis basis data yang paling banyak digunakan. Pada model ini, tabel-tabel dapat dihubungkan satu sama lain sehingga timbul suatu relasi antar tabel. Ada berbagai macam relasi antar tabel, diantaranya:

1). *One-to-One*



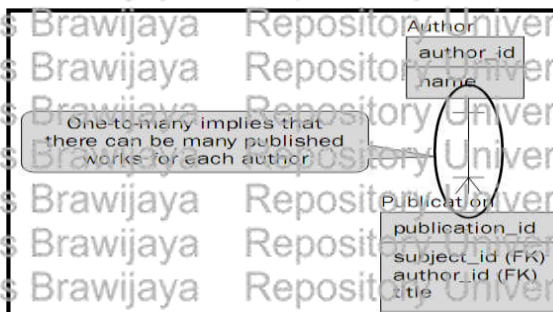
Relasi *One-to-One* biasanya digunakan untuk menghilangkan nilai NULL pada baris kolom tabel. Satu baris tabel mengacu pada satu baris pada tabel lainnya. Relasi *One-to-one* ditunjukkan pada gambar 2.26 sebagai berikut.



Gambar 2.26 Contoh Relasi *One to One*
(Sumber: Gavin Powell, 2006)

2). *One-to-Many*

Relasi *One-to-Many* merupakan relasi yang paling sering dijumpai pada *relational database model*. Satu baris tabel mengacu pada satu baris atau lebih pada tabel lainnya. Relasi *One-to-Many* ditunjukkan pada gambar 2.27 sebagai berikut.



Gambar 2.27 Contoh Relasi *One to Many*
(Sumber: Gavin Powell, 2006)

3). *Many-to-Many*

Relasi *many-to-many* berarti bahwa setiap baris tabel memiliki kemungkinan memiliki relasi dengan banyak baris pada tabel lainnya, dan sebaliknya. Biasanya relasi *many-to-many* membutuhkan tabel tambahan sebagai penghubung. Relasi *many-to-many* ditunjukkan pada gambar 2.28 sebagai berikut.



