BAB 2

LANDASAN TEORI

2.1 Pengolahan Citra (*Image Processing*)

Untuk melakukan pengenalan suatu objek berdasarkan tekstur yang dimiliki objek tersebut maka dibutuhkanlah suatu teknik untuk melakukannya, teknik tersebut adalah teknik pengolahan citra. Pengolahan citra merupakan proses pengolahan dan analisis citra yang banyak melibatkan persepsi visual. Proses ini mempunyai ciri data masukan dan informasi keluaran yang berbentuk citra. Istilah pengolahan citra digital secara umum didefinisikan sebagai pemrosesan citra dua dimensi dengan komputer. Dalam definisi yang lebih luas, pengolahan citra digital juga mencakup semua data dua dimensi. Citra digital adalah barisan bilangan nyata maupun kompleks yang diwakili oleh bit-bit tertentu.

Citra merupakan istilah lain dari gambar yang merupakan komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Citra mempunyai karakteristik yang tidak dimiliki oleh data teks, yaitu kaya akan informasi. Citra digital adalah citra hasil digitalisasi citra kontinu (analog). Tujuan dibuatnya citra digital adalah agar citra tersebut dapat diolah menggunakan komputer atau piranti digital dan memperbaiki kualitas citra agar mudah diinterpretasi oleh manusia atau mesin (komputer). Teknik-teknik pengolahan citra mentransformasikan citra menjadi citra lain yang mempunyai kualitas lebih baik.

2.2 Langkah-Langkah Penting dalam Pengolahan Citra (*image processing*)

Dalam pengolahan citra terdapat beberapa langkah penting yang harus dilakukan. Secara umum, langkah-langkah dalam pengolahan citra dapat di jabarkan menjadi beberapa langkah yaitu[5] :

1. Akuisisi Citra

Akuisisi citra adalah tahap awal untuk mendapatkan citra digital. Tujuan akuisisi citra adalah untuk menentukan data yang diperlukan dan memilih metode perekaman citra digital. Tahap ini dimulai dari objek yang akan diambil gambarnya, persiapan alat-alat, sampai pada pencitraan. Pencitraan adalah kegiatan transformasi dari citra tampak (foto, lukisan, gambar, patung, pemandangaan dan lain-lain) menjadi citra digital. Beberapa alat yang dapat digunakan untuk pencitraan adalah:

a. *Video* kamera

b. Kamera digital

c. Kamera konvesional dan *converter* analog to digital

d. Scanner

e. Photo sinar-x atau sinar infra merah

2. Preprocessing

Tahapan ini diperlukan untuk menjamin kelancaran pada proses berikutnya. Hal-hal penting yang dilakukan pada tingkatan ini diantaranya adalah:

a. Peningkatan kualitas citra (kontras, *brightness*, dan lain-lain)

b. Menghilangkan *noise*

c. Perbaikan citra (*image restoration*)

d. Transformasi (*image transformasi*)

e. Menentukan bagian citra yang akan diobservasi

3. Segmentasi

Tahapan ini bertujuan untuk mempartisi citra menjadi bagian-bagian pokok yang mengandung informasi penting. Misalnya, memisahkan objek dari latar belakang.

4. Representasi dan Deskripsi

Dalam hal ini representasi merupakan suatu proses untuk merepresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat dalam kurva yang tertutup, dengan deskripsi luasan atau perimeternya. Setelah suatu wilayah dapat direpresentasikan, proses selanjutnya adalah melakukan deskripsi citra dengan cara seleksi ciri dan ekstraksi ciri (*Feature Extraction and selection*). Seleksi ciri

bertujuan untuk memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas objek secara baik, sedangkan ekstraksi ciri bertujuan untuk mengukur besaran kuantitatif ciri setiap piksel, misalnya rata-rata, standar deviasi, koefisien variasi, signal to nois ratio (SNR), dan lain-lain.

5. Pengenalan dan interpretasi

Tahap pengenalan bertujuan untuk memberi label pada suatu objek yang informasinya disediakan oleh *descriptor,* sedangkan tahap interpretasi bertujuan untuk memberi arti atau makna kepada kelompok objek-objek yang dikenali.

6. Basis pengetahuan

Basis pengetahuan sebagai basis data pengetahuan berguna untuk memandu operasi dari masing-masing modul proses dan mengkontrol interaksi antara modul-modul tersebut. Selain itu, basis pengetahuan juga digunakan sebagai referensi pada proses *template matching* atau pada pengenalan pola.

2.3 Elemen Citra

Dalam pengolahan citra objek yang digunakan adalah sebuah citra dari sebuah objek tertentu yang mana citra tersebut mengandung sejumlah elemen dasar. Elemen dasar tersebut di manipulasi dalam pengolahan citra, elemen tersebut adalah[5]:

1. Warna

Warna adalah persepsi yang dirasakan oleh sistem visual manusia terhadap panjang gelombang cahaya yang dipantulkan oleh objek. Setiap warna mempunyai panjang gelombang. Warna yang diterima oleh mata merupakan hasil kombinasi cahaya dengan panjang gelombang berbeda. Kombinasi warna yang memberikan rentang warna yang paling lebar adalah *red* (R), *green* (G), *blue* (B)

2. Kecerahan (*brightness*)

Kecerahan disebut juga intensitas cahaya. Kecerahan pada sebuah piksel (titik) didalam citra bukanlah intensitas yang *rell*, tetapi sebenarnya adalah intensitas rata-rata dari suatu area yang melingkupinya.

3. Kontras (*contrast*)

Kontras menyatakan sebaran terang dan gelap di dalam sebuah gambar. Citra dengan kontras rendah dicirikan oleh sebagian besar komposisi citranya adalah terang atau sebagian besar gelap. Pada citra dengan kontras yang baik, komposisi gelap dan terang tersebar secara merata.

4. Kontur (*contour*)

Kontur adalah keadaan yang ditimbulkan oleh perubahan intensitas pada piksel yang bertetangga. Karena adanya perubahan intensitas, mata manusia dapat mendeteksi tepi objek didalam citra.

5. Bentuk (*shape*)

Bentuk adalah properti intrinsik dari objek tiga dimensi, dengan pengertian bahwa *shape* merupakan properti intrinsik utama untuk sistem visual manusia. Pada umumnya citra yang dibentuk oleh mata merupakan citra *dwimatra* (dua dimensi), sedangkan objek yang dilihat umumnya berbentuk *trimatra* (tiga dimensi). Informasi bentuk objek dapat diekstraksi dari citra pada permulaan pra-pengolahan dan segmentasi citra.

6. Tekstur (*texture*)

Tekstur diartikan sebagai distribusi spasial dari derajat keabuan di dalam sekumpulan piksel-piksel yang bertetangga. Jadi tekstur tidak dapat didefinisikan untuk sebuah piksel. Sistem visual manusia menerima informasi citra sebagai suatu kesatuan. Resolusi citra yang diamati ditentukan oleh skala dimana tekstur tersebut dipersepsi.

7. Waktu dan Pergerakan

Respon suatu sistem visual tidak hanya berlaku pada faktor ruang, tetapi juga pada faktor waktu. Sebagai contoh, bila citra diam ditampilkan secara cepat, akan berkesan melihat citra yang bergerak.

8. Deteksi dan Pengenalan

Dalam mendeteksi dan mengenali suatu citra, ternyata tidak hanya sistem visual manusia saja yang bekerja, tetapi juga ikut melibatkan ingatan dan daya pikir manusia.

2.4 Citra Analog dan Citra Digital

Secara umum terdapat 2 jenis citra yaiu citra analog dan citra digital. Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak dikertas foto, lukisan, pemandangan, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain-lain sebagainya[4].

Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak dapat diproses dikomputer secara langsung. Oleh sebab itu, agar citra ini dapat diproses dikomputer, 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 system USG, dan lain-lain[4] .

Citra Digital adalah citra yang dapat diolah oleh komputer dan citra digital yaitu gambar pada bidang dua dimensi. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi. Ketika sumber cahaya menerangi objek, objek memantulkan kembali sebagian cahaya tersebut. Pantulan ini ditangkap oleh alat-alat pengindera optik, misalnya mata manusia, kamera, scanner dan sebagainya. Bayangan objek tersebut akan terekam sesuai intensitas pantulan cahaya. Ketika alat optik yang merekam pantulan cahaya itu merupakan mesin digital, misalnya kamera digital, maka citra yang dihasilkan merupakan citra digital. Pada citra digital, kontinuitas intensitas cahaya dikuantisasi sesuai resolusi alat perekam[4].

2.5 Tekstur

Dalam pengolahan citra terdapat berbagai teknik pengenalan suatu citra yang salah satunya adalah pengenalan citra berdasarkan analisis tekstur. Analisis tekstur memegang peranan penting dalam pengolahan citra digital karena analisis tekstur dikembangkan dengan tujuan agar komputer dapat memahami, membuat model, serta memproses tekstur untuk dapat menirukan proses pembelajaran mata atau penglihatan manusia. Tekstur dapat dianggap sebagai pengelompokan kesamaan didalam suatu citra. Sifat sifat subpola tersebut menimbulkan cahaya yang diterima, keseragaman, kerapatan, kekasaran, keteraturan, kelinieran,

frekuensi, fase, keterarahan, ketidakteraturan, kehalusan, dan lain-lain. Karena komputer tidak memiliki indra penglihatan, maka komputer hanya mengetahui pola suatu citra digital dari ciri atau karakteristik teksturnya[5].

Secara umum tekstur mengacu pada repetisi elemen-elemen tekstur dasar yang sering disebut primitif atau *texel* (*texture element*). Suatu *texel* terdiri dari beberapa *pixel* dengan aturan posisi bersifat periodik, kuasiperiodik, atau acak. Syarat-syarat terbentuknya tekstur setidaknya ada dua, yaitu:

1. Adanya pola-pola primitif yang terdiri dari satu atau lebih pixel. Bentuk-bentuk pola primitif ini dapat berupa titik, garis lurus, garis lengkung, luasan dan lain-lain yang merupakan elemen dasar dari sebuah bentuk.
2. Pola-pola primitif tadi muncul berulang-ulang dengan interval jarak dan arah tertentu sehingga dapat diprediksi atau ditemukan karakteristik pengulangannya.

Pada analisis citra, pengukuran tekstur dikategorikan menjadi tiga kategori utama yaitu : statistik, struktural, dan pengolahan sinyal.

1. Metode statistik mempertimbangkan bahwa intensitas dibangkitkan oleh medan acak dua dimensi, metode ini berdasar pada frekuensi-frekuensi ruang. Contoh metode statistis adalah fungsi autokorelasi, matriks ko-okurensi, transformasi Fourier, frekuensi tepi, run-length.
2. Metode struktural berkaitan dengan penyusunan bagian-bagian terkecil suatu citra. Contoh metode struktural adalah model fraktal.
3. Metode pengolahan sinyal adalah metode yang berdasarkan analisis frekuensi seperti transformasi Gabor dan transformasi wavelet

2.6 Metode *Run-Length*

Seperti yang telah dijelaskan diatas bahwa metode run length merupakan salah satu metode yang dapat digunakan dalam analisis tekstur. Metode run-length menggunakan distribusi suatu pixel dengan intensitas yang sama secara berurutan dalam satu arah tertentu sebagai primitifnya. Masing – masing primitif didefinisikan atas panjang, arah, dan level keabuan. Panjang dari primitif tekstur pada arah yang berbeda dapat digunakan intuk menggambarkan suatu tekstur[2].

Analisis metode run-length ini digunakan untuk membedakan citra halus dan citra kasar. Tekstur kasar menunjukan banyaknya pixel tetangga yang memiliki intensitas yang sama sedangkan tekstur halus menunjukan sedikit pixel tetangga yang menunjukan intensitas yang sama[2].

Untuk melakukan ekstraksi ciri dengan menggunakan metode run-length, citra aras keabuan dengan matriks *f(x,y)*harus ditransformasikan terlebih dahulu kedalam matriks Grey Level Run-Length (GLRL), *B(a,r).*

$$F(x,y) \xrightarrow{GLRL} B(a,r) \tag{2.1}$$

Elemen matriks dari GLRL B(a,r) menghitung banyaknya primitif (*run*) dengan panjang r dan level keabuan a. Jumlah dari primitif (*run*) dapat diperoleh dengan persamaan berikut[2]:

$$K = \sum_{a=1}^{L} \sum_{r=1}^{Nr} B(a,r) \tag{2.2}$$

Dengan:
- L : Banyaknya level keabuan citra
- Nr : Panjang maksimal dari primitif
- K : Jumlah run
- M,N : Dimensi citra
- a :level keabuan
- r :panjang run

Adapun ciri dari tekstur dapat diperoleh dari persamaan – persamaan berikut ini[2] :

1. *Short Run Emphasis* (SRE)

   SRE mengukur distribusi short run. SRE sangat tergantung pada banyaknya short run dan diharapkan bernilai besar pada tekstur halus.

$$SRE = \frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{Nr} \frac{B(a,r)}{r^2} \tag{2.3}$$

2. *Long Run Emphasis* (LRE)

LRE mengukur distribusi long run. LRE sangat bergantung pada banyaknya long run da diharapkan bernilai besar pada tekstur kasar.

$$LRE = \frac{1}{K}\sum_{a=1}^{L}\sum_{r=1}^{Nr} r^2 B(a,r) \tag{2.4}$$

3. *Grey Level Uniformity* (GLU)

   GLU mengukur persamaan nilai derajat keabuan seluruh citra dan diharapkan bernilai kecil jika nilai derajat keabuan serupa diseluruh citra.

$$GLU = \frac{1}{K}\sum_{a=1}^{L}\left(\sum_{r=1}^{Nr} B(a,r)\right)^2 \tag{2.5}$$

4. *Run Length Uniformity* (RLU)

   RLU mengukur persamaan panjangnya run diseluruh citra dan diharapkan bernilai kecil jika panjangnya run serupa diseluruh citra.

$$RLU = \frac{1}{K}\sum_{r=1}^{Nr}\left(\sum_{a=1}^{L} B(a,r)\right)^2 \tag{2.6}$$

5. *Run Percentage* (RPC)

   RPC mengukur keserbasaman dan distribusi run dari sebuah citra pada arah tertentu. RPC bernilai paling besar jika panjangnya run adalah 1 untuk semua derajat keabuan pada arah tertentu.

$$RPC = \frac{K}{\sum_{a=1}^{L}\sum_{r=1}^{Nr} rB(a,r)} = \frac{K}{MN} \tag{2.7}$$

2.7 Jaringan Saraf Tiruan (JST)

Jaringan saraf tiruan adalah salah satu representasi buatan dari otak manusia yang selalu mencoba untuk mensimulasikan proses pembelajaran pada otak manusia. Istilah buatan ini diimplementasikan dengan menggunakan program komputer yang mampu menyelesaikan sejumlah proses perhitungan selama proses pembelajaran [8].

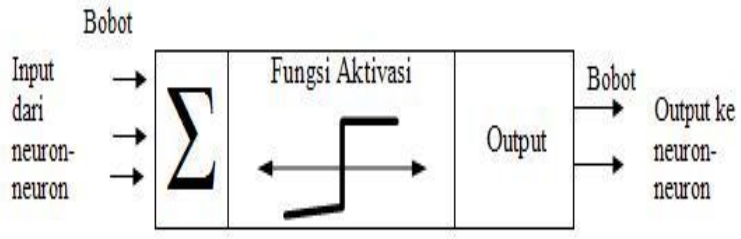
Otak manusia berisi berjuta-juta sel saraf yang bertugas untuk memproses informasi. Setiap sel saraf (neuron) akan memiliki satu ini sel, inti sel ini yang akan bertugas untuk melakukan pemrosesan informasi.

2.7.1 Komponen dan Arsitektur

Pembuatan struktur jaringan saraf tiruan diilhami oleh struktur jaringan biologi, khususnya jaringan otak manusia.

Jaringan saraf terdiri dari beberapa komponen berikut [10] :

1. Neuron, sel saraf yang akan mentrsnformasikan informasi yang diterima melalui sambungan keluarnya menuju neuron-neuron lain.
2. Hubungan antar neuron dikenal dengan nama bobot.
3. Neuron layers, neuron-neuron aka dikumpulkan dalam lapisan-lapisan (layer).
4. Informasi yang diberikan pada jaringan saraf akan dirambatkan dari lapisan ke lapisan, mulai dari lapisan input sampai lapisan output melalui lapisan lainnya, tergantung pada algoritma pembelajarannya, bisa jadi informasi tersebut akan dirambatkan secara mundur pada jaringan.



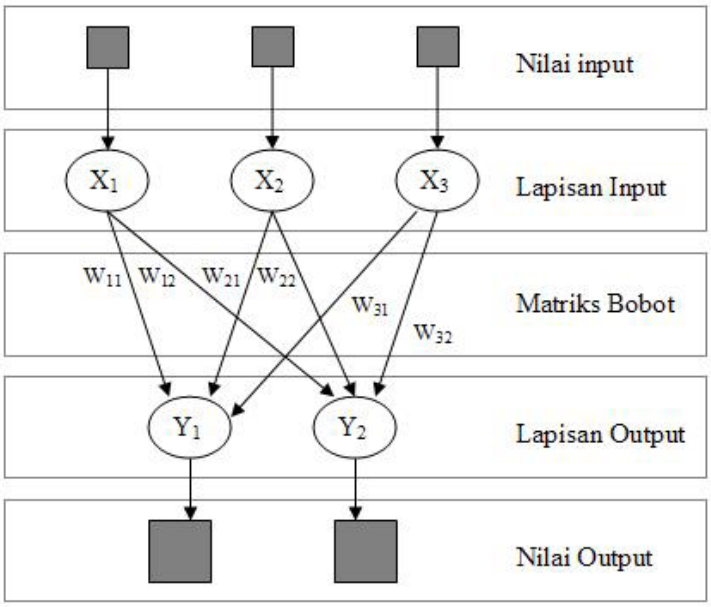
Gambar 2. 1 Model Tiruan Sebuah Neuron

Berdasarkan Gambar 2.1 faktor terpenting untuk menentukan kelakuan suatu neuron adalah fungsi aktivasi dan pola bobotnya. Ada beberapa arsitektur jaringan saraf, antara lain :

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

   Seperti yang terdapat dalam Gambar 2.2 berikut adalah ciri dari jaringan *single layer.*

a. Memiliki satu lapisan dengan bobot terhubung
b. Menerima input kamudian langsung mengolahnya menjadi output tanpa harus melewati lapisan tersembunyi.

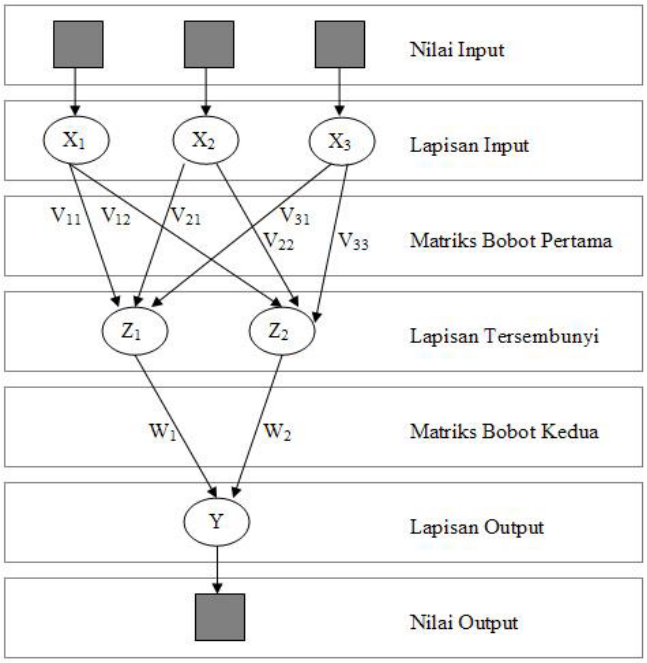


Gambar 2. 2 Model Single Layer

2. Jaringan dengan banyak lapisan (*multi layer net*)

Dibawah ini merupakan keterangan dari Gambar 2.3 yang menjelaskan tentang ciri dari jaringan *multilayer*.

a. Memiliki 1 atau lebih lapisan yang terletak diantara lapisan input dan lapisan output.
b. Ada lapisan yang berbobot yang terletak antara 2 lapisan yang bersebelahan.



Gambar 2. 3 Model Multi Layer

2.7.2 Fungsi Aktivasi

Mengaktifkan saraf tiruan berarti mengaktifkan setiap neuron yang dipakai pada jaringan tersebut. Banyak fungsi yang dipakai sebagai pengaktif, diantaranya [8] :

1. Fungsi Undak Biner (*Hard Limit*)

   Jaringan dengan lapisan tunggal sering menggunakan fungsi undak untuk mengkonversi input dari suatu variabel yang bernilai kontinu ke suatu output biner. Fungsi hard limit dirumuskan

   $$Y = \begin{cases} 0, jika\ x \leq 0 \\ 1, jika\ x > 0 \end{cases} \quad (2.8)$$

2. Fungsi Undak Biner (*Threshold*)

   Fungsi ini sering disebut fungsi nilai ambang atau fungsi *headviside*, dirumuskan :

   $$Y = \begin{cases} 0, jika\ x < \theta \\ 1, jika\ x \geq \theta \end{cases} \quad (2.9)$$

3. Fungsi Bipolar

   Hampir sama dengan undak biner, hanya output yang dihasilkan berupa 1,0 atau -1. Fungsi ini dirumuskan :

$$Y = \begin{cases} 1, jika\ x > 0 \\ 0, jika\ x = 0 \\ -1, jika\ x < 0 \end{cases} \quad (2.10)$$

4. Fungsi Bipolar (Dengan *Threshold*)

   Fungsi yang menghasilkan output berupa 1,0 atau -1

$$Y = \begin{cases} 1, jika\ x\ \geq \theta \\ -1, jika\ x < \theta \end{cases} \quad (2.11)$$

5. Fungsi Linear (Identitas)

   Fungsi linear memilki nilai output yang sama dengan nilai input, dirumuskan :

$$Y = X \quad (2.12)$$

6. Fungsi Sturating Linear

   Fungsi ini dirumuskan :

$$Y = \begin{cases} 1, jika\ x \geq 0 \\ x + 0{,}5\ , jika - o, 5 \leq x \leq 0{,}5 \\ 0, jika\ x \leq 0 \end{cases} \quad (2.13)$$

7. Fungsi Symetric Saturating Linear

   Fungsi ini dirumuskan :

$$Y = \begin{cases} 1, jika\ x \geq 0 \\ x\ , jika - 1 \leq x \leq 1 \\ -1, jika\ x \leq -1 \end{cases} \quad (2.14)$$

8. Fungsi Sigmoid Biner

   Digunakan untuk jaringan saraf yang dilatih dengan menggunakan metode backpropagation. Fungsi ini dirumuskan :

$$y = f(x) = \frac{1}{(1+e^{\acute{\alpha}x})} \quad (2.15)$$

9. Fungsi Sigmoid Bipolar

   Output dari fungsi ini memiliki range antara 1 sampai -1. Fungsi ini dirumuskan :

   $$y = f(x) = \frac{1-e^{-x}}{(1+e^{-x})} \quad (2.16)$$

2.7.3 Proses Pembelajaran

Metode pembelajaran terdiri dari beberapa metode [10] :

1. Metode Pelatihan Terbimbing

   Metode pembelajaran pada jaringan saraf disebut terawasi jika output yang diharapkan telah diketahui sebelumnya. Metode ini memasukan target keluaran alam data untuk proses pelatihan. Beberapa metode terbimbing diantaranya : Jaringan Single Perceptron, Hebb Rule, Delat Rule, Backpropagation, Hetroassociative Memory, Bidirectional Associative Memory dan Learning Vector Quantization.

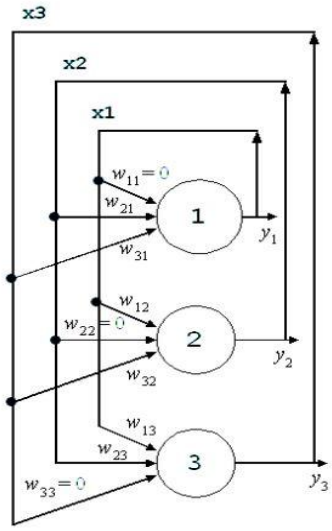
2. Metode Pelatihan Tak Terbimbing

   Metode pelatihan tak terbimbing adalah pelatihan tanpa memerlukan target pada keluarannya. Proses pelatihan berdasarkan proses transformasi dari bentuk variabel kontinyu menjadi variabel diskrit yang dikenal dengan kuantisasi vektor. Jaringan yang digunakan untuk proses pelatihan tak terbimbing ini adalah jaringan umpan balik (*feedback network*). Beberapa metode tak terbimbing adalah : Metode Kohonen/Self-Organizing Map (SOM) dan Metode Hopfield.

2.8 Metode Hopfield

Pada tahun 1982, John Hopfield dari *California Institute of Technology* merancang sebuah jaringan saraf tiruan yang kemudian dikenal dengan nama jaringan hopfield. Dalam jaringan hopfield, semua neuron saling berhubungan penuh. Neuron yang satu mengeluarkan output dan kemudian menjadi input bagi semua neuron yang lain. Proses pengiriman dan penerimaan sinyal antar neuron ini secara *feedback* tertutup dan terus menerus sampai dicapai kondisi stabil [4].

JST Hopfield merupakan salah satu metode optimasi untuk pencarian nilai minimum dari kombinasi fungsi objektif. Pada gambar 2.4 semua neuron saling berhubungan penuh. Neuron yang satu mengeluarkan output dan kemudian menjadi input bagi neuron yang lain ouptut dari setiap simpul diumpanbalikan ke input dari simpul lainnya melalui bobot koneksi W_{ij} yang tetap. Nilai W_{ij} mula-mula diinisialisasikan menggunakan rumus yang diberikan hopfield untuk seluruh pola-pola contoh [4].



Gambar 2. 4 Topologi

JST Hopfield Diskrit

Gambar 2.4 menunjukan sebuah jaringan jaringan hopfield dengan neuron yang terhubung satu sama lain. Apabila dalam pengolahan citra angka 1,2 dan 3 mewakili kelas, untuk x adalah inputan sedangkan y adalah output dan w adalah bobot. Berikut bobot-bobot tersebut digambarkan sebagai vektor W:

$$W = \begin{bmatrix} 0 & W_{12} & W_{13} & \cdots & W_{1n} \\ W_{21} & 0 & W_{23} & \cdots & W_{2n} \\ W_{31} & W_{32} & 0 & \cdots & W_{3n} \\ \vdots & \vdots & \vdots & 0 & \vdots \\ W_{n1} & W_{n2} & W_{n3} & \cdots & 0 \end{bmatrix}.$$

Bobot yang terletak pada diagonalnya adalah nol yang menunjukan bahwa neuron-neuron pada jaringan hopfield tidak memiliki hubungan dengan dirinya

sendiri[8]. Sedangkan untuk menghitung jarak terdekat adalah dengan rumus seperti berikut :

$$d([a,b],[c,d])=\sqrt{(a-c)^2+(b-d)^2} \quad (2.17)$$

d= selisih jarak

a,b= titik 1

c,d= titik 2

Dimana dalam kasus pengolahahn citra a dan b mewakili nilai aktivasi dari data latih sedangkan c dan d mewakili nilai aktivasi dari citra uji. Maksud dari rumus diatas adalah rumus untuk mencari jarak terdekat yang berarti semakin kecil selisih antara nilai aktivasi dari dataset dengan citra uji semakin mirip juga citra uji tersebut dengan dataset. Sebaliknya semakin besar selisih antara nilai aktivasi dataset dan citra uji semakin berbeda juga citra uji dengan dataset yang ada.

2.8.1 Jaringan Hopfield Kontinu

Jaringan hopfield kontinu merupakan pengembangan dari metode jaringan hopfield. Cara kerja dari metode jaringan hopfield kontinu ini meyerupai jaringan hopfield diskrit tetapi metode ini memiliki kemampuan lebih karena tidak dibatasi pada nilai biner (0 dan 1) sehingga nilai input dan output yang diharapkan pada jaringan tidak hanya biner tetapi juga bilangan riil[7].

Dalam jaringan hopfield kontinu, arsitektur dari jaringan ditentukan sehingga perubahan setiap neuron-neuronnya digambarkan secara kontinu. Arsitektur dan topologi yang digunakan dalam penelitian ini adalah sama dengan topologi dan arsitektur jaringan saraf tiruan hopfield diskrit seperti pada gambar 2.4, namun yang membedakan adalah nilai bobot yang dimilikinya yaitu pada jaringan saraf tiruan hopfield diskrit hanya ada nilai bineer sedangkan yang digunakan adalah bilangan riil antara 0 sampai 1.

2.9 OOP (Object Oriented Programming)

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun

melaluipendekatan objek secara sistematis. Metode berorientasi objek didasarkan pada penerapan prinsip-prinsip pengelolaan kompleksitas. Metode berorientasi onjek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek[12].

Pada saat ini, metode berorientasi objek banyak dipilih karena metodologi lama banyak menimbulkan masalah seperti adanya kesulitan pada saat mentransformasi hasil dari satu tahap pengembangan ke tahap berikutnya, misalnya pada metode pendekatan terstruktur, jenis aplikasi yang dikembangkan saat ini berbeda dengan masa lalu. Aplikasi yang dikembangkan saat ini sangat beragam (aplikasi bisnis, real-time, utility, dan sebagainya) dengan *platform* yang berbeda-beda, sehingga menimbulkan tuntutan kebutuhan metodologi pengembangan yang dapat mengakomodasi ke semua jenis aplikasi tersebut.

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut[12]:

a. Meningkatkan produktivitas

   Karena kelas dan objek yang ditemukan dalam suatu masalahmasih dapat dipakai ulang untuk masalah lainnya yang melibatkan objek tersebut (*reusable).*

b. Kecepatan pengembangan

   Karena sistem yang dibangun dengan baik dan benar pada saat analisis dan perancangan akan menyebabkan berkurangnya kesalahan pada saat pengkodean.

c. Kemudahan pemeliharaan

   Karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering diubah0ubah.

d. Adanya konsistensi

   Karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.

e. Meningkatkan kualitas perangkat lunak

Karena adanya pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengambangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

Berikut beberapa contoh bahasa pemrograman yang mendukung pemrograman berorientasi objek [12]:

a. Smalltalk

   Smalltalk merupakan salah satu bahasa pemrograman yang dikembangkan untuk mendukung pemrograman berorientasi objek.

b. Bahasa Pemrograman Eiffel

   Eiffel merupakan bahasa pemrograman yang dikembangkan untuk mendukung pemrograman berorientasi objek mulai tahun 1985 oleh Bertrand Meyer dan *compiler* Eiffel selesai pada tahun 1987.

c. Bahasa Pemrograman C++

   C++ merupakan pengembangan lebih lanjut bahasa pemrograman C untuk mendukung pemrograman berorientasi objek.

d. Bahasa Pemrograman (web) PHP

   PHP dibuat pertama kali oleh seorang perekayasa perangkat (*software engineering)* yang bernama Rasmus Lerdoff.

e. Bahasa Pemrograman Java

   Java dikembangkan oleh perusahaan Sun Microsystem. Java menurut definisi dari Sun Microsystem adalah nama untuk sekumpulan teknologi untuk membuat dan menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan.

2.10 UML (Unified Modeling Language)

Unified Modelling Language (UML) adalah sekumpulan spesifikasi yang dikeluarkan oleh OMG. UML terbaru adalah UML 2.3 yang terdiri dari 4 macam spesifikasi, yaitu *Diagram Interchange Specification, UML, Infrastructure, UML Superstructure, dan Object Constraint Language (OCL).* Pada UML 2.3 terdiri 13 macam diagram yang dikelompokkan pada 3 kategori, yaitu :

A. Structure Diagram, yaitu kumpulan diagram yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

1. Diagram Kelas

   Diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut attribut dan metode atau operasi.

2. Diagram Objek

   Diagram objek menggambarkan struktur sistem dari segi penamaan objek dan jalannya objek dalam sistem.

3. Diagram Komponen

   Diagram komponen dibuat untuk menunjukan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem.

4. *Composite Structure Diagram*

   Composite structure diagram baru mulai ada pada UML versi 2.0. Diagram ini dapat digunakan untuk menggambarkan struktur dari bagian-bagian yang saling terhubung maupun mendeskripsikan struktur pada saat berjalan (*runtime*).

5. *Package Diagram*

   Package diagram menyediakan cara mengumpulkan elemen-elemen yang saling terkait dalam diagram UML. Hampir semua diagram dalam UML dapat dikelompokkan menggunakan *package diagram.*

6. *Deployment Diagram*

   Deployment menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

B. *Behavior Diagram,* yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

1. *Use Case Diagram*

   Use case diagram merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah

interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat.

2. *Activity Diagram*

   Activity diagram menggambarkan *workflow* atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

3. *State Machine Diagram*

   State machine diagram digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem atau objek.

C. *Interactions Diagram*, yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi antar subsistem pada suatu sistem.

1. *Sequence Diagram*

   Sequence diagram menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek.

2. *Communication Diagram*

   Communication Diagram menggambarkan interaksi antar objek/bagian dalam bentuk urutan pengiriman pesan. Diagram komunikasi merepresentasikan informasi yang diperoleh dari diagram kelas, diagram sekuen, dan diagram *use case* untuk mendeskripsikan gabungan antara struktur statis dan tingkah laku dinamis dari suatu sistem.

3. *Timing Diagram*

   Timing diagram merupakan diagram yang fokus pada penggambaran terkait batasan waktu.

4. *Interaction Overview Diagram*

   Interaction overview diagram mirip dengan diagram aktivitas yang berfungsi untuk menggambarkan sekumpulan urutan aktivitas, diagram ini adalah bentuk aktivitas diagram yang setiap titik merepresentasikan diagram interaksi.

2.11 Bahasa C#

C# (tanda '#' dibaca "Sharp") merupakan bahasa pemograman baru yang diciptakan Microsoft seacra khusus sebagai salah satu bahasa pemrograman dalam teknologi .Net sebagai bahasa baru, C# tidak berevolusi dari bahasa C# versi bukan teknologi .Net. dengan demikian C# dapat memaksimalkan kemampuannya tanpa khawatir dengan masalah kompatibilitas dengan versi-versi sebelumnya. Keharusan sebuah perangkat lunak untuk tetap dapat kompatibel dengan versi-versi sebelumnya sebagaimana yang terjadi pada Visual Basic (VB) maupun C++biasanya menghambat optimalitas kemampuan dari perangkat lunak tersebut[12].

Sejak diluncurkan pada tahun 2000, C# dengan cepat merebut hati progammer C++ bahkan VB. Dengan tata cara penulisan yang mirip C++ dan *interface* mirip VB 6.0 menurut wikipedia, sebuah ensiklopedia gratis di internet pengguna C# .Net pada saat ini sudah melebihi pengguna VB.Net. sementara itu jumlah pengguna bahasa pemrograman lain masih berada dibawah jumlah pengguna VB .Net. masih menurut wikipedia, jumlah buku C# yang terjual pun berada dikisaran 2 hingga 3 kali lebih banyak dari jumlah buku VB yang terjual.
Dari informasi ini dapat disimpulkan bahwa C# merupakan bahasa pemrogrman baru yang sedang berkembang dan dapat diterima dengan baik oleh kebanyakan progammer dan kalangan industri. Di Microsoft sendiri, C# merupakan bahasa pemrograman yang digunakan untuk membuat perangkat lunak yang bertteknologi .Net dengan demikian dapat diperkirakan bahwa C# akan menjadi bahasa pemrograman yang akan banyak digunakan di masa-masa mendatang.