BAB 2
LANDASAN TEORI

2.1 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.

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 .

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.

2.2 Elemen Citra

Citra mengandung sejumlah elemen dasar. Elemen dasar tersebut di manipulasi dalam pengolahan citra, elemen tersebut adalah:

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 25
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.

2.3 Pengolahan Citra (*Image Processing*)

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.

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

Secara umum, langkah-langkah dalam pengolahan citra dapat di jabarkan menjadi beberapa langkah yaitu :

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.

2.4 Pengenalan Pola (*Pattern Recognition*)

Menurut Fairhurst (1998), pengenalan pola merupakan bidang yang berhubungan dengan proses identifikasi objek pada citra atau interpretasi citra. Proses ini bertujuan untuk mengekstrak informasi yang disampaikan oleh citra. Pengenalan pola mempunyai arti bidang studi yang melakukan proses analisis citra yang bentuk masukannya adalah citra itu sendiri atau dapat juga berupa citra digital dan bentuk keluarannya adalah suatu deskripsi. Tujuan pengenalan pola adalah untuk meniru kemampuan manusia dalam mengenali objek tertentu.

Pengenalan pola merupakan proses pengenalan suatu objek dengan menggunakan berbagai metode. Teknik pencocokan pola adalah salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokkan tiap-tiap bagian dari suatu citra dengan citra yang menjadi acuan (*template*). Beberapa contoh pengenalan pola yang telah dilakukan seperti pengenalan wajah, fingerprint, pola permainan catur, retina mata, peramalan cuaca, pengenalan tekstur, dan lain lain.

2.4.1 Metode Pengenalan Pola

Ada 3 metode pengenalan pola yaitu:

1. Metode Statistik

   Metode ini dapat dilakukan dengan pendekatan *supervised* maupun pendekatan *unsupervised*. Pendekatan *supervised* (dengan pengarahan) menyediakan *training set* untuk mengarahkan atau memberi informasi atau pengetahuan tentang kelas-kelas yang ada. Pendekatan *unsupervised* (tanpa pengarahan) tidak menyediakan *training set*. Informasi yang disediakan adalah jumlah klaster yang ada. Sehingga pengelompokan dilakukan sepenuhnya berdasarkan karakteristik data.

2. Metode Sintaktik

   Dalam metoda sintaktik atau metoda struktural, pola dipilah berdasarkan keserupaan ukuran struktural, 'pengetahuan' direpresentasikan secara formal grammar atau deskripsi relasional (*graf*). Metode ini dipakai tidak hanya untuk pemilahan, tetapi juga untuk deskripsi. Biasanya, metode ini memformulasikan deskripsi hirarkis dari pola kompleks yang tersusun dari pola bagian yang lebih sederhana.

3. Metode Jaringan Syaraf

   Dalam metode jaringan syaraf, pemilahan dilakukan berdasarkan tanggapan suatu neuron jaringan pengolah sinyal (*neuron*) terhadap stimulus masukan (pola). 'Pengetahuan' disimpan dalam sambungan antar neuron dan kekuatan pembobot sinaptik.

2.5 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.

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 statistik 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 Matriks Ko-Okurensi Aras Keabuan (*Gray Level Co-Occurrence Matrix* - GLCM)

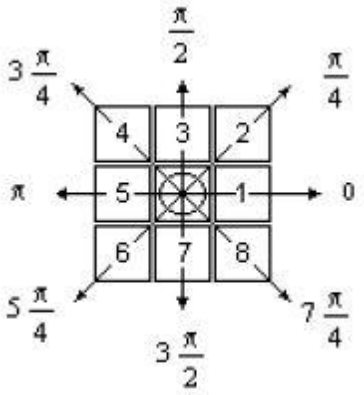
Salah satu teknik untuk memperoleh ciri statistik adalah dengan menghitung probabilitas hubungan ketetanggaan antara dua piksel pada jarak dan arah tertentu. Hal ini dapat dilakukan dengan membentuk sebuah matriks kookurensi dari data citra.

Matriks kookurensi adalah suatu matriks yang menggambarkan frekuensi munculnya pasangan dua piksel dengan intensitas tertentu dalam jarak (d) dan orientasi sudut (θ) tertentu dalam citra. Penggunaan metode ini berdasar pada hipotesis bahwa dalam suatu tekstur akan terjadi perulangan konfigurasi atau pasangan aras keabuan. Jarak dinyatakan dalam piksel dan orientasi sudut dinyatakan dalam derajat. Orientasi dibentuk dalam empat arah sudut dengan interval sudut 45°, yaitu 0°, 45°, 90°, dan 135°. Sedangkan jarak antar piksel ditetapkan sebesar 1 piksel.

Langkah - langkah pembentukan matriks *Co-Occurrence*:

1. Menentukan jarak antara dua titik dalam arah vertikal dan horizontal.
2. Menghitung jumlah pasangan piksel yang mempunyai nilai intensitas *i* dan *j*.
3. Kemudian hasil perhitungan setiap pasangan nilai intensitas diletakan pada matriks sesuai dengan koordinatnya.
4. Matriks kemudian dijumlahkan dengan transposenya untuk menghitung nilai simetrisnya.
5. Kemudian dilakukan normalisasi dengan cara membaginya dengan bilangan yang merupakan jumlah dari total pasangan piksel.

Suatu piksel yang bertetangga yang memiliki jarak *d* diantara keduanya, dapat terletak di delapan arah yang berlainan. Hal ini ditunjukan pada gambar 2.1



Gambar 2.1 piksel bertetangga dalam delapan arah

Dalam matriks ko-okurensi, terdapat empat ciri tekstur yang dapat diperoleh dari suatu citra yang digunakan sebagai pembeda antara citra dengan kelas tertentu, dengan kelas lainnya. Ciri – ciri tersebut adalah:

Dengan : d = Jarak

P = Probabilitas

θ = Sudut (Digunakan untuk perhitungan jarak antar piksel)

i,j = Jarak antar piksel

1. Kontras (*Contrast*)

   Kontras adalah fitur yang digunakan untuk mengukur kekuatan perbedaan intensitas dalam citra. nilai kontras membesar jika variasi intensitas citra tinggi dan menurun bila variasi rendah. Berikut rumus kontras:

   $$kontras = \sum_i \sum_j (i-j)^2 Pd\ (i,j) \quad (2.1)$$

2. Homogenitas (*Homogenity*)

   Homogenitas digunakan untuk mengukur kehomogenan variasi intensitas citra. Nilai homogenitas akan semakin membesar bila variasi intensitas dalam citra mengecil.

   Berikut rumus homogenitas:

$$Homogenitas = \sum_i \sum_j \frac{Pd(i,j)}{1+|i-j|} \quad (2.2)$$

3. Energi (*Energy*)

   Energi adalah fitur untuk mengukur konsentrasi pasangan intensitas pada matriks kookurensi. Nilai energi akan semakin membesar bila pasangan piksel yang memenuhi syarat matriks intensitas kookurensi terkosentrasi pada beberapa koordinat dan mengecil bila letaknya menyebar. Berikut rumus energi:

   $$Energi = \sum_i \sum_j P_d^2\,(i,j) \quad (2.3)$$

4. Entropi (*entropy*)

   Entropi menunjukan ketidakteraturan distribusi intensitas suatu citra. Berikut rumus entropi:

   $$Entropi = -\sum_i \sum_j Pd\,(i,j)\log(Pd(i,j)) \quad (2.4)$$

2.7 Metode *Run-Length*

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.

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.

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

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

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:

$$K = \sum_{a=1}^{L} \sum_{r=1}^{Nr} B(a,r) \tag{2.5}$$

Dengan: L : Banyaknya level keabuan citra

Nr : Panjang maksimal dari primitif

K : Jumlah run

M,N : Dimensi citra

Adapun ciri dari tekstur dapat diperoleh dari persamaan – persamaan berikut ini :

1. *Short Run Emphasis* (SRE)

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

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

2. *Long Run Emphasis* (LRE)

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

   $$LRE = \frac{1}{K} \sum_{a=1}^{L} \sum_{r=1}^{Nr} r^2 B(a,r) \tag{2.7}$$

3. *Gray Level Uniformity* (GLU)

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

$$GLU = \frac{1}{K}\sum_{a=1}^{L}\left(\sum_{r=1}^{Nr} B(a,r)\right)^2 \quad (2.8)$$

4. *Run Length Uniformity* (RLU)

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

$$RLU = \frac{1}{K}\sum_{r=1}^{Nr}\left(\sum_{a=1}^{L} B(a,r)\right)^2 \quad (2.9)$$

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.

$$RPC = \frac{K}{\sum_{a=1}^{L}\sum_{r=1}^{Nr} rB(a,r)} = \frac{K}{MN} \quad (2.10)$$

2.8 K-Mean

Data Clustering merupakan salah satu metode *Data Mining* yang bersifat tanpa arahan (*unsupervised*). Ada dua jenis data clustering yang sering dipergunakan dalam proses pengelompokan data yaitu *hierarchical* (hirarki) data clustering dan *non-hierarchical* (non-hirarki) data clustering. *K-Means* merupakan salah satu metode data clustering non hirarki yang berusaha mempartisi data yang ada ke dalam bentuk satu atau lebih *cluster*/kelompok. Metode ini mempartisi data ke dalam cluster/kelompok sehingga data yang memiliki karakteristik yang sama dikelompokkan ke dalam satu cluster yang sama dan data yang mempunyai karakteristik yang berbeda dikelompokkan ke dalam kelompok yang lain. Adapun tujuan dari data clustering ini adalah untuk meminimalisasikan *objective function* yang diset dalam proses clustering, yang pada umumnya berusaha meminimalisasikan variasi di dalam suatu cluster dan memaksimalisasikan variasi antar cluster.

Data clustering menggunakan metode K-Means ini secara umum dilakukan dengan algoritma dasar sebagai berikut:

1. Menentukan pusat cluster

   Pusat cluster digunakan untuk menentukan jarak data mana yang lebih dekat dan untuk mengelompokan data dalam cluster.

2. Menghitung jarak data pada pusat cluster

   Menghitung jarak pada pusat cluster dilakukan untuk mendapatkan jarak antara data pada pusat cluster yang terdekat. Berikut rumus perhitungan jarak data pada pusat cluster:

$$D_e = \sqrt{(x_i - s_i)^2 + (y_i - t_i)^2} \tag{2.11}$$

3. Pengelompokan data

   Pengelompokan data dilakukan dengan merubah jarak menjadi 0 dan 1. Hal ini dilakukan dengan cara merubah jarak terdekat menjadi nilai 1 dan jarak terjauh menjadi nilai 0.

4. Penentuan pusat cluster baru

   Penentuan pusat cluster baru digunakan untuk menguji data yang dihasilkan tetap dan tidak berpindah pada pusat cluster lain. Penentuan pusat cluster baru dilakukan dengan cara membagi pusat cluster awal dengan jumlah anggotanya.

5. Pengulangan langkah kedua

   Pengulangan langkah kedua dilakukan untuk memastikan posisi data tidak mengalami perubahan atau perpindahan pada cluster lain.

2.8 K-Fold Cross validation

Dalam machine learning weka, pengujian keakurasian dapat dilakukan dengan 3 tipe pengujian, yaitu

1. Training set test
2. Supplied set test
3. K-fold cross validation

1. Training set test

Metode pengujian menggunakan data yang telah di training, dengan kata lain, data training dan data uji adalah data yang sama

2. Supplied set test

Metode pengujian menggunakan data yang berbeda, dengan kata lain, data training berbeda dengan data yang akan diujikan

3. K-fold cross validation

Cross Validation merupakan salah satu teknik untuk menilai/memvalidasi keakuratan sebuah model yang dibangun berdasarkan dataset tertentu. Pembuatan model biasanya bertujuan untuk melakukan prediksi maupun klasifikasi terhadap suatu data baru yang boleh jadi belum pernah muncul di dalam dataset. Data yang digunakan dalam proses pembangunan model disebut data latih/training, sedangkan data yang akan digunakan untuk memvalidasi model disebut sebagai data test.

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.

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:

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 diubah-ubah.

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 :

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 terdisi 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.