BAB 2
TINJAUAN PUSTAKA

Landasan teori diperoleh dari studi literatur. Studi literatur diperlukan untuk mengeksplorasi teori-teori yang diperlukan dalam menyelesaikan tugas akhir ini. Fungsi dari teori adalah sebagai alat dalam mencapai satuan pengetahuan yang sistematis dan menjadi pembimbing bagi penulis dalam melakukan penelitian.

2.1 *Optical Character Recognition* (OCR)

Optical Character Recognition (OCR) adalah teknik mengubah sebuah gambar berisi teks, tulisan tangan, menjadi teks yang dapat diubah untuk proses selanjutnya, Sejarah OCR dimulai sejak 1950 dan masih terus berkembang sampai saat ini. Teknologi ini terus berkembang seiring berkembangnya teknologi. Teknologi ini memungkinkan mesin untuk mengenali mengenali teks secara otomatis, teknik ini seperti kombinasi dari mata dan otak manusia, sebuah mata dapat melihat seuah teks dari gambar tapi sebenarnya otak yang memproses dan mengolah teks tersebut sehingga dapat dibaca oleh mata. Dalam pengembangan OCR terdapat beberapa masalah, diantaranya yang pertama adalah terdapat beberapa huruf dan angka yang sulit dibedakan karena memiliki ciri yang hampir sama, faktor kedua adalah faktor cahaya dalam citra yang mempersulit system mengenali huruf [7].

2.2 Dokumen Karya Tulis Ilmiah

Karya ilmiah adalah suatu karya yang memuat dan mengkaji suatu masalah tertentu dengan menggunakan kaidah-kaidah keilmuan. Adapun, yang dimaksud dengan kaidah-kaidah keilmuan adalah bahwa karya ilmiah menggunakan metode ilmiah di dalam membahas permasalahan, menyajikan kajiannya menggunakan bahasa baku dan tata tulis ilmiah, serta menggunakan prinsip-prinsip keilmuan yang lain seperti: bersifat objektif, logis, empiris (berdasarkan fakta), sistematis, lugas, jelas, dan konsisten [12]. Ragam bahasa ilmiah adalah sarana verba yang digunakan untuk mengkomunikasikan proses kegiatan dan hasil penalaran ilmiah, dalam penulisan [13]:

a. Penulisan laporan yang berbentuk surat, artikel, maupun berbentuk naskah, laporan hasil penelitian, makalah
b. Skripsi, tesis, dan disertasi.
c. Laporan pekerjaan yang berbentuk surat, artikel, maupun naskah
d. Laporan pertanggungjawaban, laporan kegiatan, laporan keuangan, laporan pemegang saham.

Sedangkan ciri ragam bahasa ilmiah sebagai berikut :

a. Jelas struktur kalimat dan maknanya
b. Singkat, berisi analisis dan pembuktian, menyajikan konsep secara lengkap
c. Cermat dalam memilih istilah/kata, ejaan, bentuk kata, kalimat, paragraf, dan penalaran
d. Mereproduksi konsep atau temua yang sudah ada dan mengembangkan dengan temuan baru atau konsep yang belum pernah ada
e. Objektif dapat diukur kebenarannya secara terbuka oleh umum, menghindarkan bentuk persona, dan ungkapan subjektif
f. Menggunakan unsur baku:kosakata/Istilah, bentuk kata, kalimat, dan penalaran ilmiah
g. Konsisten dalam menggunakan penalaran, istilah, sudut pandang, pengendalian variabel tiopik, permassalahan, tujuan, penggunaan landasan teori, pembahasan, sampai dengan kesimpulan dan saran.

2.3 Pengolahan Citra

Istilah citra digital sangat polpuler pada masa sekarang. Banyak perlatan elektronik,misalnya *scanner*, kamera digital, mikroskop digital, dan *fingerprint reader* (pembaca sidik jari), yang menghasilkan citra digital. Perangkat lunak untuk mengolah citra digital jugasangat popular digunakan oleh pengguna untuk mengolah foto atau untuk berbagai keperluan lain. Penolahan citra merupakan proses memanipulasi dan menganalisis citra dengan bantuan komputer, dalam hal ini mengolah informasi yang terdapat pada suatu gambar untuk keperluan pengenalan objek secara otomatis. Ada

berbagai teknik pengolahan citra tergantung kebutuhan dan keluaran yang diinginkan [8].

2.4 *Preprocessing*

Pre-processing adalah bagisan penting dari setiap system pemrosesan Bahasa alami, karena karakter, kata, dan kalimat yang diintifikasi pada tahap ini adalah unit dasar atau awal sebelum pemrosesan lebih lanjut. *Pre-processing* dilakukan karena data teks sering mengandung berbagai macam format atau berbeda-beda seperti format angka dan kata-kata yag tidak membantu dan dapat dihilagkan sehingga memudahkan proses selanjutya.

Image *preprocessing* adalah suatu bentuk pengolahan atau pemrosesan sinal dengan input berupa gambar (image) dan ditransformasikan menjadi gambar lain sebagai keluatan dengan teknik tertentu. *Image preprocessing* dilakukan untuk memprbaiki kesalahan data sinyal akibat transmisi dan selama akusisi sinyal, serta untuk meningkatkan kualitas penampakan gambar agar lebih mudah diinterpretasi oleh system penglihatan manusia baik dengan melakukan manipulasi dan juga penganalisisan terhadap gambar. Operasi *image preprocessing* dapat dikelompokan berdasarkan tujuan transformasinya.

2.4.1 *Grayscale*

Grayscale adalah istilah untuk menyebutkan satu citra yang memiliki warna abu-abu, hitam, dan putih. Grayscale adalah koleksi atau kisaran corak monokromik (abu-abu), mulai dari putih murni di ujung yang paling terang hingga hitam murni di ujung yang berlawanan.

Citra *grayscale* adalah citra yang hanya memiliki 1 buah kanal sehinga yang ditapilan hanyalah nilai intensitas atau dikenal juga dengan istilah derajat keabuan. Karena jenis citra ini hanya memiliki 1 kanal saja, maka citra *grayscale* memiliki tempat penyimpanan yang lebih hemat. Jenis citra ini disebut juga sebagai 8-bit. Foto hitam putih maupun gambar yang ditampilkan oleh televisi hitam putih sebenarnya meggunakan citra *grayscale*, bukan dalam warna hitam dan warna putih. Namun

dikalagan masyarakat istilah foto hitam putih maupun televisi hitam putih sudah terbiasa digunakan falam kehidupan sehari-hari [14].

Secara teori ada beberapa cara dalam mengonversi citra berwarna RGB ke dalam citra *grayscale*. Untuk mendapatkan hasil konversi yang baik, persamaan berikut dapat digunakan

$$GS = 0.299R' + 0.587G' + 0.114B' \quad (2.1)$$

Keterangan :

GS : Citra *grayscale*

R’ : Komponen merah dari citra RGB

G’ : Komponen hijau dari citra RGB

B’ : Komponen biru dari citra RGB

2.4.2 *Thresholding*

Citra biner atau citra hitam putih adalah citra yang hanya memiliku 2 kemungkinan nilai untuk setiap pikselnya, yaitu 0 atau 1. Nilai 0 akan tampil sebagai sebagai warna hitam sedangkan nilai 1 akan tampil sebagai warna putih. Maka dari itu, jenis citra ini hanya membutuhkan 1-bit untuk menyimpan setiap nilai pada setiap pikselnya. Jenis citra ini sering digunakan untuk proses *masking* ataupun proses segmentasi citra [14]. Proses thresholding digunakan untuk mengekstrak *foreground* (tinta) dan *background* (kertas) dan mengubah menjadi citra biner. Proses *thresholding* mengubah warna gambar menjadi citra biner (*binary image*) dimana ditentukan sebuah nilai level *threshold* kemudian piksel yang memiliki nilai level dibawah level *threshold* di set menjadi warna putih (1 pada nilai biner) dan nilai diatas nilai *threshold* di set menjadi warna hitam (0 pada nilai biner).

2.4.3 *Resize*

Resize adalah proses yang digunakan untuk mengubah ukuran citra digital dalam piksel, baik menjadi lebih kecil atau lebih besar dari ukuran sebenarnya. Proses resize pada penelitian ini tidak memerlukan metode khusus, caranya hanya dengan dilakukan

perbandingan ukuran antara citra hasil thresholding (pada tahap latih) dan hasil segmentasi (pada tahap uji) dengan menggunakan persamaan (2.4 **)** untuk mendapatkan posisi koordinat dari titik x yang baru dan persamaan (2.5) untuk mendapatkan posisi koordinat dari titik y yang baru. Dimana kedua persamaan tersebut adalah sebagai berikut.

$$Xbaru = \frac{pb \times pp}{pa} \qquad (2.4)$$

Keterangan:

$Xbaru$ = Posisi koordinat x baru

pb = Ukuran panjang dari matriks baru

pp = Posisi koordinat x lama

pa = Ukuran panjang dari matriks lama

Untuk mencari koordinat y yang baru, digunakan persamaan (2.5 **)** sebagai berikut.

$$Ybaru = \frac{lb \times pp}{la} \qquad (2.5)$$

Keterangan:

$Ybaru$ = Posisi koordinat y baru

lb = Ukuran lebar dari matriks baru

pp = Posisi koordinat y lama

la = Ukuran lebar dari matriks lama

Pada penelitian ini *resize* digunakan untuk merubah ukuran *pixel* sesuai dengan kebutuhan dalam melakukan proses ektraksi ciri. Hal ini dilakukan agar semua citra karakter dari hasil segmentasi mempunyai ukuran yang sama (normalisasi ukuran citra) sebelum masuk ke tahap ekstraksi fitur.

2.4.4 Segmentasi

Segmentasi bertujuan untuk memotong huruf per-huruf. Pemotongan tersebut dilakukan dengan cara mencari pixel-pixel terluar dari setiap sisi (atas, bawah, kiri, kanan). Pixel-pixel terluar itulah yang akan menjadi batas pemotongan, sehingga didapat citra segiempat yang siap diproses lebih lanjut.

2.4.5 Profile Projection

Metode *Profile Projection* digunakan untuk memisahkan tulisan perbaris dan perkarakter. Metode *Profile Projection* akan menghitung jumlah piksel non-background secara vertikal dan horizontal dan nilai tersebut akan dibandingkan dengan nilai ambang tertentu untuk memisahkan tulisan perbaris dan perkarakter [15].

2.4.5.1 *Horizontal Profile projection*

Horizontal Profile Projection adalah jumlah banyaknya piksel hitam yang tegak lurus dengan sumbu x. *Horizontal Profile Projection* dipresentasikan dengan suatu vektor Ph berukuran M. *Horizontal Profile Projection* pada kolom ke-j, yaitu Ph[j], didefinisikan sebagai berikut :

$$P_h[h] = \sum_{j=1}^{N} S[i,j] \qquad (2.6)$$

2.4.5.2 *Vertical Profile Projection*

Vertical Profile Projection adalah jumlah banyaknya piksel hitam yang tegak lurus dengan sumbu y. *Vertical Profile Projection* dipresentasikan dengan suatu vektor Pv berukuran N. *Vertical Profile Projection* pada baris ke-I, yaitu Pv[i], didefinisikan sebagai berikut :

$$P_v[v] = \sum_{i=1}^{M} S[i,j] \qquad (2.7)$$

Keterangan :

S : Citra Biner

M : Banyak kolom pada citra

N : Banyak baris pada citra

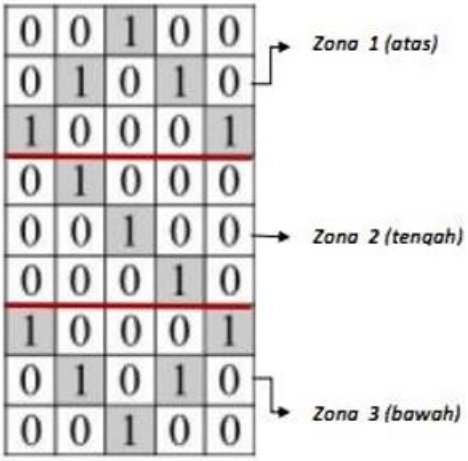
2.4.6 Ekstraksi Fitur

Ekstraksi fitur bertujuan untuk mendapatkan karakteristik suatu karakter yang membedakan dari karakter lain. dalam penelitian ini, untuk mendapatkan karakteristik dengan cara memasukan nilai pixel gambar ke dalam sebuah vektor atau array 1 dimensi. Sebelum dimasukkan ke array, pixel diubah terlebih dahulu menjadi nilai 1 dan 0 dimana 1 mewakilkan warna putih dan 0 mewakilkan warna hitam pada gambar.

2.4.6.1 Zone Based Feature Extraction

Zoning adalah salah satu ekstraksi fitur yang paling popular dan sederhana untuk diimplementasikan [16]. Setiap citra dibagi menjadi N x M zona dan dari setiap zona tersebut dihitung nilai fitur sehingga didapatkan fitur dengan panjang N x M. Salah satu cara menghitung nilai fitur setiap zona adalah dengan menghitung jumlah piksel hitam setiap zona dan membaginya dengan jumlah piksel hitam terbanyak pada

yang terdapat pada salah satu zona. Contoh pembagian 3 zona pada citra biner dapat dilihat dari Gambar 2.1.



Gambar 2.1 Pembagian Zona pada Citra Biner

Metode ekstraksi fitur berbasis zona memberikan hasil yang baik bahkan ketika langkah sebelum proses tertentu dimulai seperti filtering. Smoothing dan menghapus zona yang dianggap tidak ada. Konsep metode ekstraksi ciri yang digunakan untuk mengekstraksi fitur untuk klasifikasi yang efisien dan pengenalan yaitu [16] :

1. Hitung *centroid* dari citra. Menghitung centroid dari citra biner masukan dengan persamaan (2.8) dan (2.9 **)**.

   Rumus mencari *centroid* X

$$C_x = \frac{(x_1.p_1 + x_2.p_2 + \cdots + x_n.p_n)}{(p_1 + p_2 + \cdots + p_n)} \qquad (2.8)$$

   Rumus mencari *centroid* Y

$$C_y = \frac{(y_1.p_1 + y_2.p_2 + \cdots + y_n.p_n)}{(p_1 + p_2 + \cdots + p_n)} \qquad (2.9)$$

   Keterangan :

   a. Cx = *centroid* koordinat x

b. Cy = *centroid* koordinat y
c. Xn = koordinat x dari piksel ke-n
d. Yn = koordinat y dari piksel ke-n
e. Pn = nilai piksel ke-n

2. Bagi matriks kedalam *n* buah zona yang sama besar proporsinya.
3. Hitung jarak antara titik *centroid* dengan koordinat *pixel* yang memiliki nilai. Persamaan untuk menghitung jarak piksel.

$$d(P,C) = \sqrt{(x_p - C_x)^2 + (y_p - C_y)^2} \qquad (2.10)$$

Keterangan :

a. d = jarak antara dua titik
b. P = koordinat piksel
c. C = koordinat centroid
d. X_p = koordinat piksel X
e. Y_p = koordinat piksel Y
f. C_x = koordinat centroid X
g. C_y = koordinat centroid Y

4. Ulangi langkah 3 untuk *pixel* yang ada di semua zona.
5. Hitung rata-rata dari jarak yang telah didapat pada langkah 3.

$$rata - rata\ jarak = \sum \frac{d(P,C)}{\sum P} \qquad (2.11)$$

6. Ulangi langkah 5 hingga didapat masing-masing rata-rata jarak dari setiap zona.
7. Akhirnya *n* buah fitur akan didapat untuk melakukan klasifikasi dan pengenalan

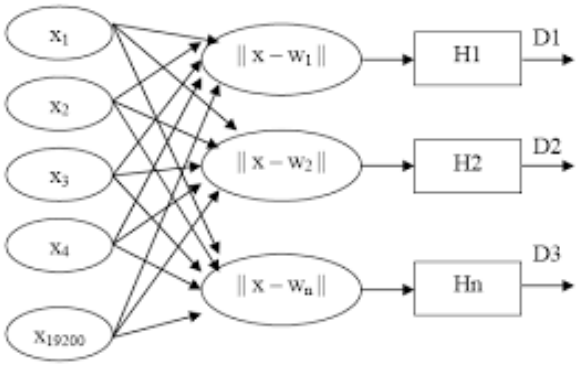
2.5 *Learning Vector Quantization* (LVQ)

Leaning Vector Quantization (LVQ) merupakan salah satu bagian algoritma dari Jaringan Syaraf Tiruan (JST), danbisa disebut dengan jaringan LVQ, yang dimana

setiap unit outputya mempresentasikan sebuah kelas, metode LVQ digunakan untuk klasifikasi dimana arsitekturnya sudah sitentukan (kelas sudah ditentukan) [17]. Keunggulan dari metode LVQ adalah mampu untuk memberikan pelatihan terhadap lapisan-lapisan kompetitf sehingga secara otomatis dapat mengklasifikasikan vektor masukan yang diberikan. Jaringan saraf tiruan merupakan representasi buatan yang mencoba mensimulasikan proses pembelajaran pada otak manusia. Jaringan saraf tiruan diimplementasikan dengan menggunakan program komputer sehingga mampu menyelesaikan proses perhitungan ketika proses pembelajaran berlangsung [18].

LVQ menggunakan konsep kuantisasi perbedaan antara vektor masukan dengan bobot yang dimiliki oleh setiap neuron. Pada LVQ, satu kelas dapat diwakili oleh lebih dari satu neuron, oleh karena itu LVQ tidak membutuhkan hidden layer seperti pada MLP. Operasi yang dilakukan antara vektor dengan bobot tidak menggunakan *inner-product,* melainkan menggunakan kuantisasi perbedaan *Euclidean* kuadrat. Jika vektor input mendekati sama maka lapisan kompetitif akan mengklasifikasikan kedua vektor tersebut kedalam kelas yang sama.

Adapun arsitektur LVQ dapat dilihat seperti gambar berikut :



Gambar 2.2 Arsitektur LVQ

1. Inisialisasi nilai vektor dan menentukan kelas untuk setiap vektor dan ambil satu ciri bobot (W) untuk setiap kelas masukan. Sedangkan sisanya dijadikan data masukan (X).

2. Menentukan nilai *max epoch* (MacEpoch), *error minimum* (Eps), *learning rate* (α) dan pengurangan rasio.
3. Selama kondisi bernilai salah kerjakan langkah 4-6. Dan lakukan langkah 5-6 untuk setiap data masukan (X).
4. Untuk masing-masing vektor dilakukan perhitungan dengan persamaan (2.12 **)**.

$$Cj = \sqrt{\sum_{i=1}^{n}(X_i - W_j)^2} \quad (2.12)$$

Keterangan :

X_1 sampai X_{19200} = nilai input

W_1 sapai W_j = nilai bobot

$\| X - W_1 \|$ sampai $\| X - W_n \|$ = jarak bobot

n = jumlah data karakter (jumlah kelas)

5. Setelah didapatkan nilai C_j, kemudian periksa nilai terkecil
6. Nilai bobot terkecil dilakukan pembaharuan dengan persamaan (2.13 **)** jika nilai nilai kelas sama dengan C_j dan persamaan (2.14 **)** digunakan jila nilai kelas tidak sama dengan C_j

$$\overrightarrow{W_{(new)}} = \overrightarrow{W_{(old)}} + \alpha \,.\, (\overrightarrow{X} - \overrightarrow{W_{(old)}}) \quad (2.13)$$

$$\overrightarrow{W_{(new)}} = \overrightarrow{W_{(old)}} - \alpha \,.\, (\overrightarrow{X} - \overrightarrow{W_{(old)}}) \quad (2.14)$$

Keterangan:

W = bobot

α = *learning rate*

$\overrightarrow{X}$ = input

7. Setelah itu dilakukan perhitungan perubahan nilai *learning rate* dan nilai rasio dengan persamaan

$$\alpha = \alpha * pengurangan\, rasio \quad (2.15)$$

$$rasio = \alpha - pengurangan\ rasio \quad (2.16)$$

8. Perhitungan dilakukan sampai mendapatkan nilai bobot terakhir untuk digunakan pada proses testing.

2.6 Pengujian Sistem

Pengujian sistem adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diterapkan dengan hasil yang terjadi [19].
Pengujian seharusnya meliputi tiga konsep berikut :

1. Demonstrasi validasi perangkat lunak pada masing-masing tahap diskusi pengembangan sistem.
2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai.
3. Pemeriksaan prilaku sistem dengan mengeksekusi sistem pada data sample pengujian.

Pada dasarnya pengujian diartikan sebagai aktivitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dilakukan bagitu skesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik pengujian. Kategori pengujian dapat dikategorikan menjadi dua [19], yaitu :

1. Berdasarkan ketersediaan *logic* sistem, terdiri dari *Black Box testing* dan *White Box testing.*
2. Berdasarkan arah pengujian, terdiri dari *top down* dan pengujian *Bottom up.*

2.6.1 Pengujian *Black Box*

Konsep *black box* digunakan untuk mempresentasikan sistem yang cara kerja di dalamnya tidak tersedia untuk diinspeksi. Dalam *black box,* item-item yang diuji dianggap “gelap” karena logiknya tidak diketahui, yang diketahui hanya apa yang masuk dana pa yang keluar dari *black box* [20].

2.6.2 Pengujian Akurasi

Akurasi merupakan seberapa dekat suatu angka hasil pengukuran terhadap angka sebenarnya (*true value* dan *reference value*). Tingkat akurasi diperoleh dengan persamaan sebagai berikut [20] :

$$Akurasi = \frac{jumlah\ karakter\ sama}{jumlah\ seluruh\ karakter} \times 100\% \qquad (2.17)$$

2.7 Python

Python adalah Bahasa pemrograman bersifat umum. *Python* diciptakan pada tahun 1990 oleh Guido van Rossum. Bahasa level tinggi, stabil, dinamis, orientasi objek dan cross *platform* adalah karakteristik yang membuat Bahasa python disukai oleh banyak pengembang. Tidak seperti bahasa lain yang susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih mudah untuk memahami sintaks. Hal ini membuat python sangat mudah dipelajari baik untuk pemula maupun untuk yang sudah menguasai Bahasa pemrograman lain. Bahasa pemrograman python berjalan dikebanyakan hardware dan system operasi, sehingga kebanyakan computer bias menjalankannya. Bahasa pemrograman *Python* saat ini dikembangkan dan dikelola oleh suatu tim relawan dengan nama *Python Software Foundation* [21].

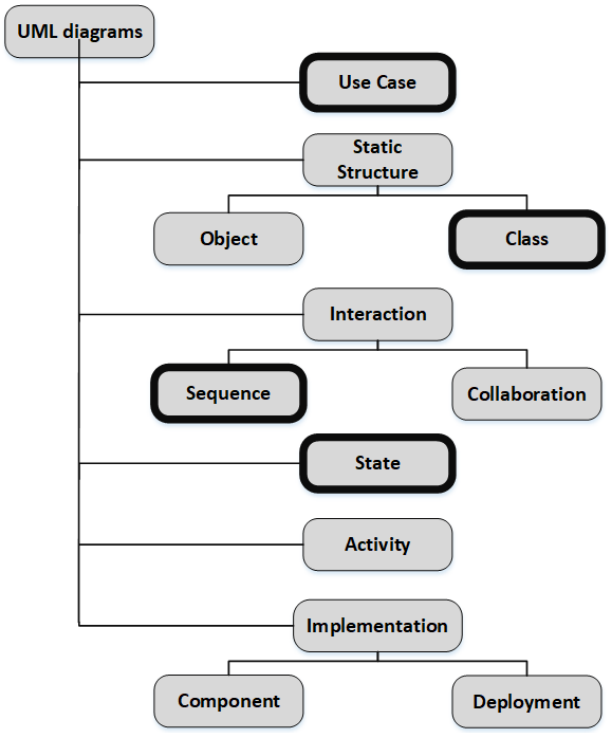
2.8 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah bahasa pemodelan visual yang digunakan untuk menspesifikasikan, menvisualisasikan, membangun, dan mendokumentasikan rancangan dari suatu sistem perangkat lunak [22].

Pemodelan memberikan gambaran yang jelas mengenai sistem yang akan dibangun baik dari sisi struktural ataupun fungsional. UML dapat diterapkan pada semua model pengembangan, tingkatan siklus sistem, dan berbagai macam domain aplikasi. Dalam UML terdapat konsep semantik, notasi, dan panduan masing-masing

diagram. UML bertujuan menyatukan teknik-teknik pemodelan berorientasi objek-objek menjadi terstandarisasi [22].

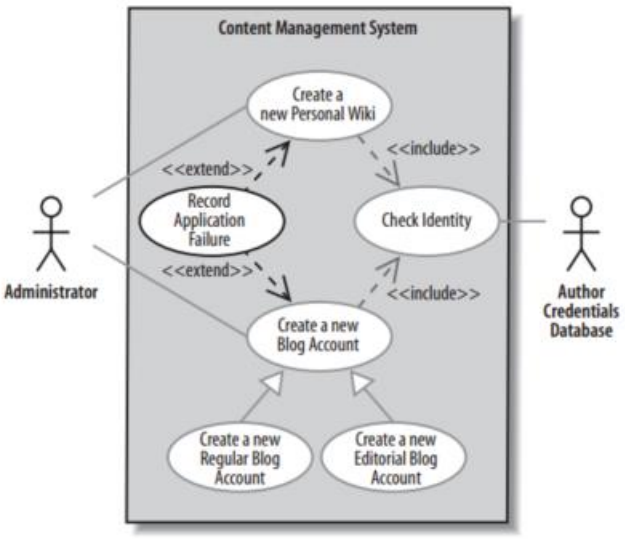
UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera Rational Software Corps. UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai prespetktif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan [23].



Gambar 2.3 *Unified Modeling Language* Diagram [23]

2.8.1 Use Case Diagram

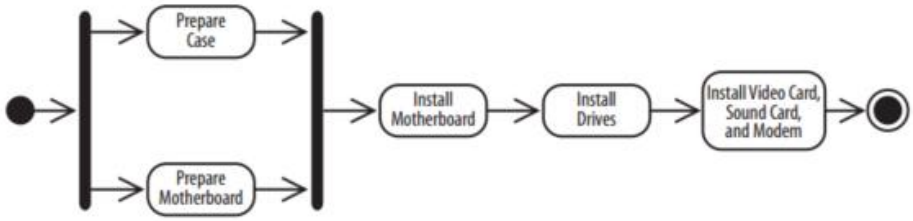
Use case diagram menyajikan interakasi antara use case dan aktor. Dimana, aktor dapat berupa orang, peralatan, atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. Use case menggambarkan fungsionalitas sistem atau persyaratan-persyaratan yang harus dipenuhi sistem dari pandangan pemakai. Sebuah use case digambarkan sebagai elips horizontal dalam suatu diagram UML use case.



Gambar 2.4 Contoh *Use Case* Diagram [23]

2.8.2 Activity Diagram

Activity diagram merupakan diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. *Activity* diagram juga digunakan untuk mendefinisikan urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan serta rancang menu yang ditampilkan pada perangakat lunak.

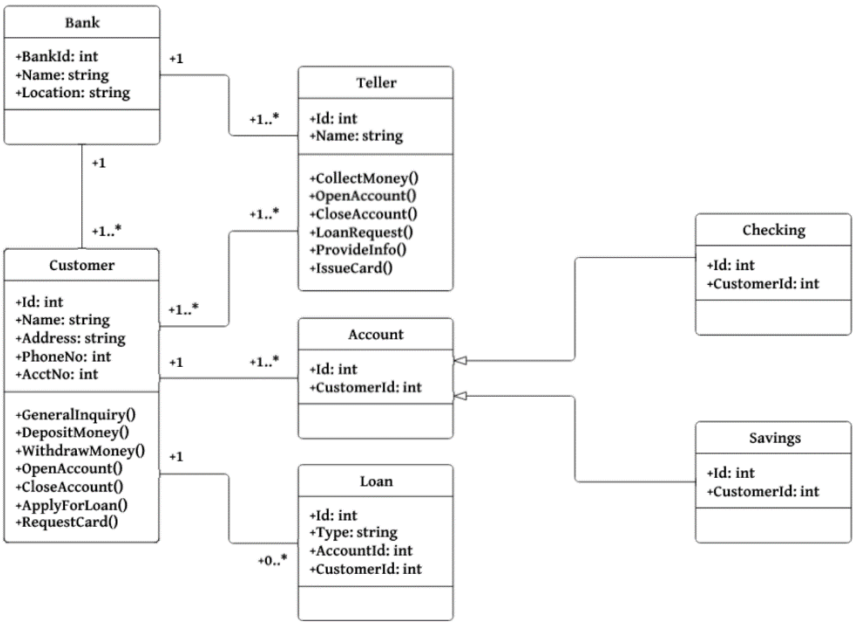


Gambar 2.5 Contoh Activity Diagram [23]

2.8.3 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class* diagram memiliki apa yang disebut atribut dan metode atau operasi. *Class* diagram mendeskripsikan jenis-jenis objek dalam sistem dan berbagai hubungan statis yang terdapat di antara mereka. *Class*

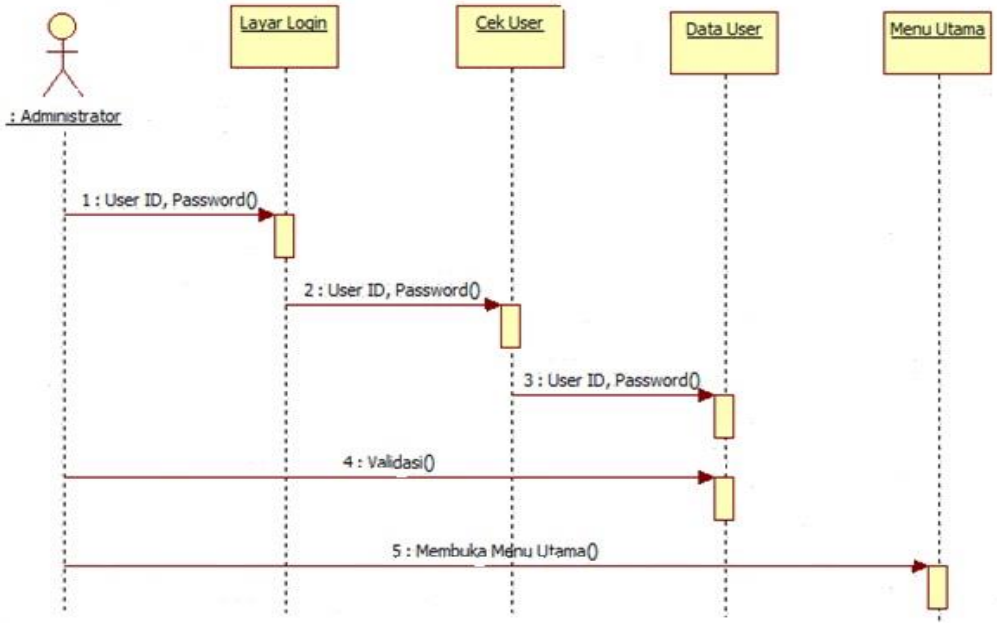
Diagram juga menunjukkan properti dan operasi sebuah kelas dan batasan-batasan yang terdapat dalam hubungan-hubungan objek tersebut.



Gambar 2.6 Contoh Class Diagram [23]

2.8.4 Sequence Diagram

Sequence diagram merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan, *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu dan objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2.7 Contoh Sequence Diagram [23]