BAB 2
TINJAUAN PUSTAKA

2.1 Citra

Citra atau image merupakan istilah lain dari gambar yang merupakan informasi berbentuk visual. Suatu citra adalah fungsi intensitas 2 dimensi f(x,y), dimana x dan y adalah koordinat sparsial dan f pada titik (x,y) merupakan tingkat kecerahan suatu citra pada suatu titik [5]. Citra terbagi menjadi 2 jenis yaitu citra analog dan citra digital.

2.1.1 Citra Analog

Citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak di kertas foto, lukisan, pemandangan alam, hasil CT scan, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya [6].

2.1.2 Citra Digital

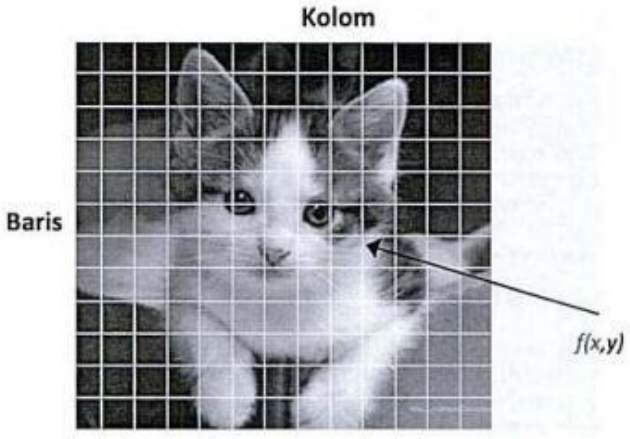
Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/diskrit nilai digital yang disebut pixel/picture elements. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahata pada bidang dua dimensi [6]. Suatu citra dapat didefinisikan sebagai fungsi f(x,y) berukuran M baris dan N kolom, dengan x dan y adalah koordinat spasial, dan amplitudo f di titik koordinat (x,y) dinamakan intensitas atau tingkat keabuan dari citra pada titik tersebut. Apabila nilai x,y dan nilai amplitudo f secara keseluruhan berhingga (*finite*) dan bernilai diskrit maka dapat dikatakan bahwa citra tersebut adalah citra digital [4]. Citra digital berukuran M x N dinyatakan dengan matriks M x N.

M = Jumlah Baris $0 \leq y \leq M-1$

N = Jumlah Baris $0 \leq x \leq N-1$

Citra digital dipresentasikan sebagai matriks berukuran M x N dimana M = baris dan N = kolom sebagai berikut

$$f(x,y) = \begin{matrix} f(0,0) & f(0,1) & f(0,N-1) \\ f(1,0) & f(1,1) & f(1,N-1) \\ f(M-1,0) & f(M-1,1) & f(M-1,N-1) \end{matrix}$$



Gambar 2. 1 Citra Digital

2.2 Pengolahan Citra Digital

Pengolahan citra digital artinya melakukan pemrosesan gambar berdimenasi dua melalui komputer digital. Pengolahan citra adalah isitlah umum untuk berbagai teknik yang keberadaannya untuk memanipulasi dan memodifikasi citra dengan berbagai cara [6]. Foto adalah contoh gambar berdimensi dua yang dapat diolah dengan mudah. Setiap foto dalam bentuk citra digital dapat diolah menggunakan perangkat lunak tertentu. Misalnya citra yang gelap dapat diolah supaya menjadi lebih terang, merubah format warna menjadi grayscale, resize citra dan lain sebagainya.

2.3 *Pattern Recognition*

Pattern Recognition (PR) atau pengenalan Pola, merupakan salah satu cabang AI (*artifical intelegence)* [6], yang mempelajari pembuatan sebuah sistem untuk dapat mengenali suatu pola tertentu. Menurut Ursa Majorsy, Pola dalam hal ini merujuk pada pengertian suatu komposisi stimulus penginderaan yang kompleks yang dapat dikenali oleh manusia sebagai pengamat sebagai suatu kelompok objek. Rekognisi pola merupakan proses pengenalan kembali terhadap pola yang pernah

dikenal. Oleh karena itu, jika kita melihat wajah teman kita atau mendengar lagu kesukaan kita, kita dapat mengenal masing-masing persepsi tersebut sebagai sesuatu yang sebelumnya telah dialami.

2.3.1 Penerapan *Pattern Recognition*

Seperti yang telah disebutkan di atas bahwa AI merupakan salah satu cabang Ilmu Komputer. Tapi karena kompleksitas area AI maka dibuatkan sub-sub bagian yang dapat berdiri sendiri dan dapat berdiri sendiri dan saling bekerja sama dengan sub bagian lain atau dengan disiplin ilmu lain. Berikut ini beberapa cabang ilmu sub bagian dari AI :

1. *Character Recognition*

   Salah satu era pengenalan pola yang secara umum menangani permasalahan otomatisasi dan informasi. Sistem OCR mempunyai *front end device* yang terdiri dari lensa scan, document, data transport dan sebuah detector.

2. *Speech Recognition*

   Pengenalan pola yang secara umum telah banyak dikembangkan saat ini. Sistem ini mengijinkan kita untuk berkomunikasi antara manusia dengan memasukan data ke komputer. Meningkatkan efisiensi industri manufaktur, mengontrol mesin, dan sebagainya.

3. *Face Recognition*

   Salah satu pengaplikasian sistem pengenalan pola yang ditunjukan kepada pengenalan pola wajah, *face recognition* adalah sebuah sistem yang mengenali image wajah manusia yang biasa digunakan untuk otomatisasi dan security sebuah industri.

4. *Machine Vision*

   Sebuah sistem mesin yang menggunakan pengenalan pola sebagai dasar dari tahapannya. Mesin ini menangkap sebuah atau sekelompok objek dengan kamera dan selanjutnya dianalisa untuk di deskripsikan.

2.4 Konvolusi

Konvolusi adalah sebuah proses dimana citra dimanipulasi dengan eksternal mask untuk menghasilkan citra yang baru. Dimana eksternal mask dinyatakan dalam bentuk matriks yang ukurannya lebih kecil dari ukuran citranya [8]. Konvolusi pada operasi 2 dimensi dapat didefinisikan pada persamaan berikut:

$$I'(\hat{g}, \hat{f}) = \sum_{i=-n}^{n} \sum_{j=-m-1}^{m} h(i,j) I(n,m) \quad (2.1)$$

Operasi konvolusi dilakukan dengan cara menggeser mask *pixel* per *pixel* pada citra. Dimulai dari *pixel* awal (0,0) ke *pixel* (0,1) dan seterusnya hingga akhir dari pixel. Hasil konvolusi akan disimpan dalam matriks baru.

2.5 Resistor

Resistor adalah komponen elektronika berjenis pasif yang mempunyai sifat menghambat arus listrik. Satuan nilai dari resistor adalah ohm (Ω). Resistor merupakan komponen elektronika dua kutub yang didesain untuk menahan arus listrik. Nilai tengangan terhadap resistansi berbanding dengan arus yang mengalir [1].

2.5.1 Resistor Tetap

Resistor tetap merupakan suatu resistor yang nilai resistansinya tidak dapat diubah. Resistor tetap ini memiliki nilai resistansi yang tertulis pada badan resistor dengan menggunakan kode warna dan kode angka. Resistor ini digunakan sebagai penghambat arus listrik secara permanen dalam rangkaian elektronika. Berikut ini adalah nilai kode warna yang terdapat pada masing masing cincin:

Tabel 2. 1 Kode Warna Resistor

Warna Ring ke-	1	2	3	4
Hitam	0	0	10^0	
Coklat	1	1	10^1	
Merah	2	2	10^2	
Jingga	3	3	10^3	

Kuning	4	4	10^4	
Hijau	5	5	10^5	
Biru	6	6	10^6	
Ungu	7	7	10^7	
Abu-Abu	8	8	10^8	
Putih	9	9	10^9	
Emas	-	-	10^{-1}	5%
Perak	-	-	10^{-2}	10%
Tak Berwarna	-	-	-	20%



Gambar 2. 2 Resistor Tetap

2.6 *Template Matching*

Template matching merupakan salah satu teknik dalam pengolahan citra digital yang berfungsi untuk mencocokan tiap-tiap bagian dari suatu citra dengan citra yang menjadi *template* (acuan) [7]. Pada dasarnya proses dilakukan dengan cara memasukan citra yang mengandung *template* tertentu kemudian dibandingkan dengan citra pada database. *Template* ditempatkan pada pusat bagian citra yang akan dibandingkan dan dihitung seberapa banyak titik yang paling sesuai dengan *template.* Langkah ini diulangi terhadap citra masukan [8].

Nilai kesesuaian titik yang paling besar antara citra masukan dan citra *templ ate* menandakan bahwa *template* tersebut merupakan citra *template* yang paling se suai masukan.

Template Matching dapat dibagi antara 2 pendekatan, yaitu:

1. Pendekatan berbasis fitur

Pendekatan berbasis fitur menggunakan pencarian dan *template* gambar sepe rti tepi sebagai pembanding pengukuran matriks untuk menemukan lokasi *Templat e Matching* yang terbaik pada sumber gambar. Pendekatan ini dianggap sebagai pe ndekatan yang lebih efektif jika *template* gambar memiliki fitur yang kuat jika pen cocokan di pencarian gambar bisa diubah dengan cara tertentu. Karena pendekatan ini tidak mempertimbangkan keseluruhan *template* gambar, komputasi lebih efesie n ketika bekerja dengan gambar yang memiliki resolusi lebih besar [8].

2. Pendekatan berbasis template

Ketika sebagian besar *template* gambar merupakan gambar yang cocok, sebu ah pendekatan berbasis template mungkin efektif. Karena berbasis *template, templ ate matching* berpotensi memerlukan sample dengan mengurangi resolusi pencaria n dari *template* gambar oleh faktor yang sama dan melakukan operasi pada peramp ingan gambar yang dihasilkan [8].

2.7 *Normalized Cross-Correlation*

Normalisasi merupakan salah satu proses dalam tahap *preprocessing* yang dilakukan sebelum masuk ke proses pendeteksian. Proses untuk merubah atau menormalkan tingkat intensitas cahaya pada kedua citra tersebut, citra dipetakan pada *pixel* dengan ukuran tertentu sehingga memberikan representasi dimensi yang tetap. Proses yang digunakan pada tahap normalisasi ini mengggunakan konsep normalisasi sesuai dengan metode *Normalized Coss Corelation,* dimana dalam tahapan ini meliputi, menghitung *mean* (rata-rata) untuk citra uji dan citra template, dan menghitung nilai *standard deviation.*

Persamaan menghitung *mean* sebagai berikut :

$$\bar{x} = \frac{x1+x2+x3+\cdots+xn}{n} \tag{2.2}$$

Sedangkan persamaan untuk menghitung *standard deviation* sebagai berikut:

$$\hat{f} = \frac{\mathrm{f}-\bar{\mathrm{f}}}{\sqrt{\Sigma(\mathrm{f}-\bar{\mathrm{f}})^2}} \qquad \hat{\mathrm{g}} = \frac{\mathrm{g}-\bar{\mathrm{g}}}{\sqrt{\Sigma(\mathrm{g}-\bar{\mathrm{g}})^2}} \tag{2.3}$$

Keterangan :

f = Template

g = image target

$\bar{f}$ = *mean* dari f

$\bar{g}$ = *mean* dari g

$\hat{f}$ = normalisasi dari f

$\hat{g}$ = normalisasi dari g

Korelasi adalah alat penting dalam pengolahan citra, pengenalan pola, pendeteksian objek dan bidang lainnya. Korelasi antar dua sinyal adalah pendekatan standar untuk mendeteksi fitur dalam suatau citra. Penggunaan cross-correlation untuk pencocokan template termotivasi oleh ukuran jarak (Euclidean Distance).

$$d_{f,t}^2(u,v = \sum_x \sum_y [f(x,y) - t(x-u,y-v)]^2 \quad (2.4)$$

(perjumlahan di atas x,y dengan posisi fitur yang ada pada u,v) atau jika d^2 d iperluas maka

$$d_{f,t}^2(u,v = \sum_x \sum_y [f(x,y) - 2f(x,y)t(x-u,y-v) + t^2(x-u,y-v)]$$

Dimana nilai $\sum t^2(x-u,y-v)$ selalu bernilai konstan. Jika $\sum t^2(x,y)$ men dekati nilai konstan, maka Cross-Correlation

$$c(u,v) = \sum_x \sum_y [f(x,y)t(x-u,y-v)] \quad (2.5)$$

Untuk mendapatkan titik-titik yang sama antar citra, maka dapat digunakan *c orrelation-based*. Dalam *correlation-based* kita memotong citra pertama dengan u kuran tertentu pada bagian sudut kemudian membandingan *pixel* per *pixel* dengan citra yang kedua, dimana citra yang kedua tetap utuh, untuk mencari letak persama an dari kedua citra tersebut [8].

Cross correlation merupakan perkalian antara dua buah matriks yang berbed a yang digunakan untuk mengkategorikan data berdasarkan nilai yang sama denga n rumus sebagai berikut:

$$NCC(f,g) = cfg(\hat{f}.\hat{g}) = \Sigma_{[i,j]} \hat{f}(i.j)\ \hat{g}(i.j) \quad (2.6)$$

f = citra template

g = citra target

cfg = korelasi antara citra template dan citra target

$\hat{f}$ = normalisasi dari citra template

ĝ = normalisasi dari citra target

2.8 Model Warna HSV

Pada pengolahan warna gambar, ada bermacam model warna. Model RGB (*red green blue*) merupakan model yang banyak digunakan. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain RGB terdapat juga model warna HSV dimana model ini terdapat 3 komponen yaitu *hue*, *saturation*, dan *value*. *Hue* adalah salah satu ukuran panjang gelombang yang terdapat pada warna dominan yang diterima oleh penglihatan, sedangkan *saturation* adalah ukuran banyaknya cahaya putih yang bercampur pada *hue.*

$$R' = \frac{R}{255} \tag{2.7}$$

$$G' = \frac{G}{255} \tag{2.8}$$

$$B' = \frac{B}{255} \tag{2.9}$$

$$H = \begin{cases} 0, & jika\ max = min \\ 60° \ x \ \left(\frac{(G-B)}{max-min}\ mod\ 6\right), & jika\ max = R \\ 60° \ x \ \left(\frac{(B-R)}{max-min} + 2\right), & jika\ max = G \\ 60° \ x \ \left(\frac{(R-G)}{max-min} + 4\right), & jika\ max = B \end{cases} \tag{2.10}$$

$$S = \begin{cases} 0, & jika\ max = min \\ \frac{max-min}{v} & \end{cases} \tag{2.11}$$

$$V = max \tag{2.12}$$

2.9 Kuantisasi Warna

Kuantisasi warna adalah proses untuk mereduksi jumlah warna yang mereprentasi suatu citra. Salah satu kombinasi bins yang baik dari segi performa serta beban komputasi yang ringan dan sering digunakan yaitu 72 bins (8 bins komponen *hue*, 3 bins komponen *saturation*, dan 3 bins komponen *value*). Pembagian rentang bins tersebut didefinisikan pada persamaan (2.13), (2.14) dan (2.15)

$$H = \begin{Bmatrix} 0\ jika\ h\ \in [316-20] \\ 1\ jika\ h\ \in [21-40] \\ 2\ jika\ h\ \in [41-75] \\ 3\ jika\ h\ \in [76-155] \\ 4\ jika\ h\ \in [156-190] \\ 5\ jika\ h\ \in [191-270] \\ 6\ jika\ h\ \in [271-295] \\ 7\ jika\ h\ \in [296-315] \end{Bmatrix} \quad (2.13)$$

$$S = \begin{Bmatrix} 0\ jika\ s\ \in [0-0.2] \\ 1\ jika\ s\ \in [0.2-0.7] \\ 2\ jika\ s\ \in [0.7-1] \end{Bmatrix} \quad (2.14)$$

$$V = \begin{Bmatrix} 0\ jika\ v\ \in [0-0.2] \\ 1\ jika\ v\ \in [0.2-0.7] \\ 2\ jika\ v\ \in [0.7-1] \end{Bmatrix} \quad (2.15)$$

Setelah mendapatkan nilai kuatisasi, kemudian digabungkan menjadi 8x3x3 histogram dan mendapatkan vektor 72 dimensi. Kombinasi nilai h, s, v akan dinormalisasi menjadi 1 vektor. Ini akan mengkuantisasi seluruh ruang hsv menjadi 72 bagian warna dengan 1 dimensi. Persamaan untuk mengkuantisasi ruang warna hsv dapat dilihat pada persamaan (2.16)

$$G = 9H + 3S + V \quad (2.16)$$

2.10 UML (*Unified Modeling Language*)

Unified Modeling Language (UML) adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer. Di dalam pemodelan objek guna menyajikan sistem yang berorientasi pada objek dan pada orang lain, akan sangat sulit dilakukan jika pemodelan tersebut dilakukan dalam bentuk kode bahasa pemrograman [9].

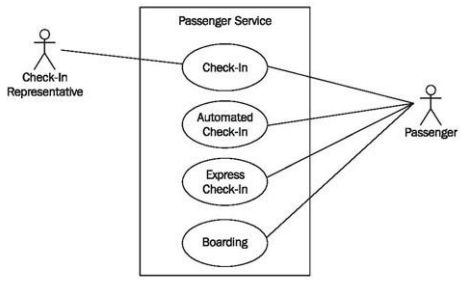
UML disebut sebagai bahasa pemodelan bukan metode. Bahasa pemodelan (sebagian besar grafik) merupakan notasi model yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk visualisasi, spesifikasi, pembentukan dan pendokumentasian. Alat-alat dari sistem perangkat lunak. UML biasanya disajikan dalam bentuk diagram atau

gambar yang meliputi *class* beserta atribut dan operasinya, serta hubungan antara kelas. UML terdiri dari banyak diagram diantaranya *use case diagram* , *activity diagram, class diagram,* dan *squence diagram*.

2.10.1 *Use Case Diagram*

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang sesungguhnya merupkan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya.

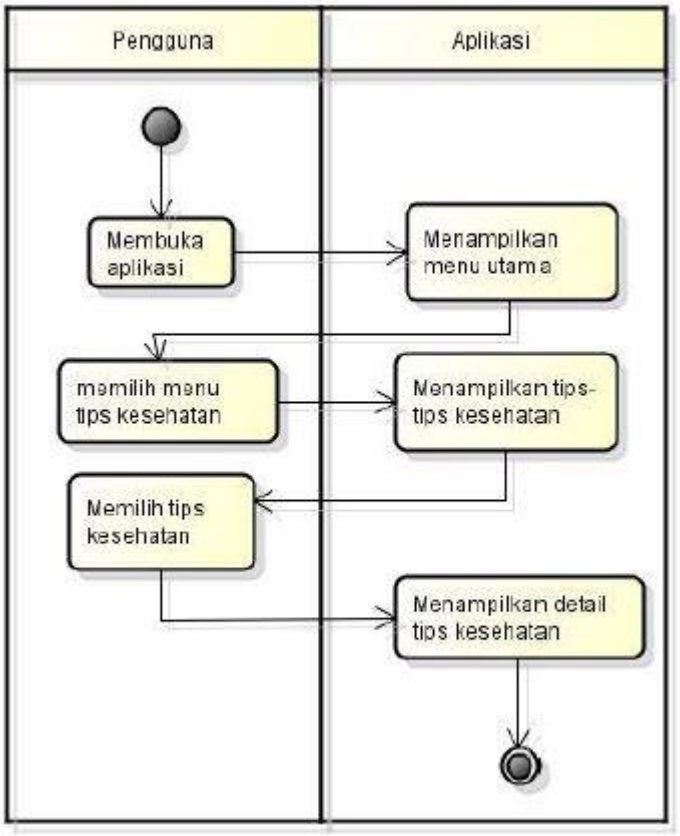
Use Case Diagram merupakan deskripsi lengkap tentang interaksi yang terjadi antara para akor dengan sistem [9]. Dalam hal ini, setiap objek yang berinteraksi dengan sistem merupakan aktor untuk sistem berprilaku kepada aktronya. Aktor dalam *use case diagram* digambarkan sebagai ikon yang berbentuk manusia, sementara *use case* digambarkan sebagai ikon yang berbentuk manusia, sementara *use case* digambarkan elips yang berisi nama *use case* yang bersangkutan. Untuk mempermudah pemahaman, aktor biasanya dituliskan sebagai kata benda, sementara *use case* biasanya dituliskan dengan kata kerja.



Gambar 2. 3 Contoh *Use Case Diagram*

2.10.2 *Activity Diagram*

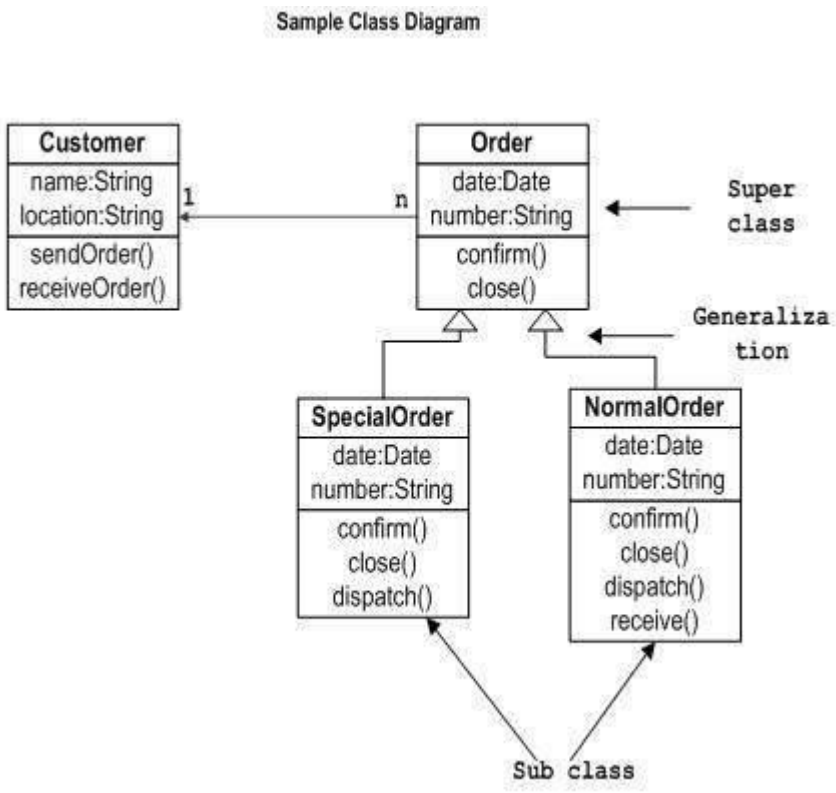
Activity Diagram pada dasarnya menggambarkan skenario secara grafis. *Activity Diagram* cukup serupa dengan diagram alur (*flow chart*), yang membedakan hanya *swimlane* yang menunjukkan suatu *state* berada pada objek/kelas tertentu. Keunggulan dari *activity diagram* adalah lebih mudah dipahami dibanding skenario. Selain itu, dengan menggunakan *activity diagram*, kita dapat melihat dibagian manakah sistem dari suatu skenario akan berjalan [9].



Gambar 2. 4 Contoh Activity Diagram

2.10.3 *Class Diagram*

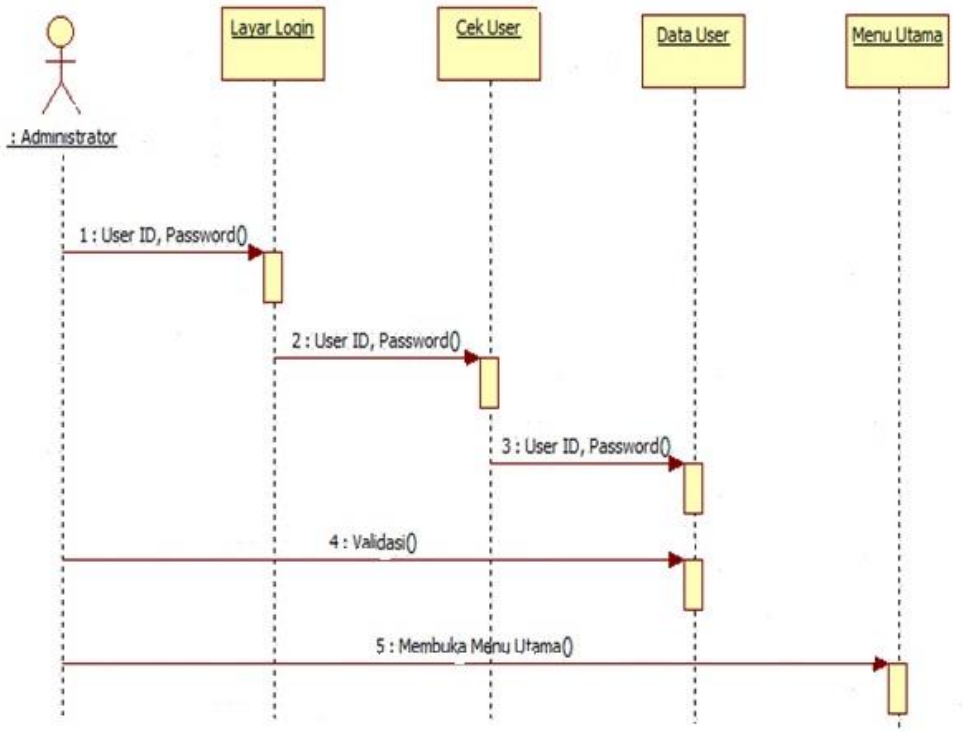
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class Diagram* menggambarkan struktur dan deskripsi kelas, package, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan sebagainya.



Gambar 2. 5 Contoh *Class Diagram*

2.10.4 *Sequence Diagram*

Diagram seakan menggambarkan kelakuan/perilaku objek pada *use case*, dengan mengdeskripsikan waktu hidup objek dan message yang dikirimkan dan diteruma antar objek [9]. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Oleh karena itu untuk menggambarkan diagram skuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



Gambar 2. 6 Contoh Sequence Diagram

2.11 Bahasa pemrograman

Bahasa pemrograman diperlukan untuk menjalankan instruksi-instruksi apa yang harus dilakukan komputer. Penelitian ini menggunakan bahasa pemrograman Java untuk mengembangkan aplikasi dan bahasa pemograman SQL untuk mengelola *database.*

2.11.1 Java

Bahasa Java dikembangkan oleh sebuah tim yang diketuai oleh James Gosling di Sun Microsystem. Java awalnya dikenal dengan Oak, yang didesain pada tahun 1991 untuk chip-chip yang tertanam pada peralatan-peralatan elektronik. Pada tahun 1995, diberi nama baru Java, yang didesain ulang untuk mengembangkan aplikasi-aplikasi internet. Java telah menjadi sangat populer. Perkembangannya yang sangat cepat dan penerimaannya di kalangan pengguna dapat dijejak dari karakteristik perancangannya, khususnya dari janji pengembang Java bahwa begitu Anda menciptakan suatu program, maka Anda bisa menjalankannya di mana saja [10].

Java memiliki banyak fitur, bahasa pemrograman bertujuan-umum yang dapat digunakan untuk mengembangkan aplikasi-aplikasi tingkat tinggi. Saat ini, Java tidak lagi hanya digunakan untuk pemrograman Web, tetapi juga aplikasi-aplikasi *standalone* bebas *platform* pada *server*, *desktop*, dan divais-divais bergerak (*mobile*) [10]. Bahasa Java juga telah digunakan untuk mengembangkan kode dalam berkomunikasi dan mengendalikan robot di Mars. Banyak perusahaan yang sebelumnya meremehkan keunggulan Java sekarang malah menggunakannya untuk mengembangkan aplikasi-aplikasi terdistribusi yang dapat diakses oleh banyak konsumen melalui internet.

Java merupakan bahasa pemograman yang tangguh terbukti handal pada banyak aplikasi. Terdapat tiga edisi java, yaitu :

1. Java SE (*Standar Edition*)
   Digunakan untuk mengembangkan aplikasi-aplikasi pada sisi client atau *applet.*
2. Java EE (*Enterprise Edition*)
   Digunakan untuk mengembangkan aplikasi-aplikasi pada sisi *server*, seperti *Java Servlets* dan *Java server pages.*
3. Java ME (*Micro Edition*)
   digunakan untuk mengembangkan aplikasi-aplikasi untuk *device* bergerak, seperti telepon genggam.