

## **BAB 2**

### **LANDASAN TEORI**

#### **2.1 Tanda Tangan**

Tanda Tangan atau dalam Bahasa Inggris *signature* yang juga berasal dari kata latin *signare* yang berarti “tanda” atau paraf merupakan tulisan tangan atau goresan tinta dari tangan, kadang-kadang diberi gaya tulisan tertentu dari nama seseorang atau tanda identifikasi lainnya yang ditulis pada dokumen sebagai suatu bukti dari identitas dan kemauan [1]. Menurut Kamus Besar Bahasa Indonesia (KBBI) tanda tangan sebagai lambang nama yang dituliskan dengan tangan oleh orang itu sendiri sebagai penanda pribadi [11]. Tanda tangan adalah hasil proses menulis seseorang yang bersifat khusus sebagai substansi simbolik [2]. Tanda tangan merupakan bentuk yang paling banyak digunakan untuk identifikasi seseorang. Contoh-contoh tanda tangan setiap orang umumnya identik namun tidak sama, artinya tanda tangan seseorang sering berubah-ubah setiap waktu. Perubahan ini menyangkut posisi, ukuran maupun faktor tekanan tanda tangan. Pada kenyataannya, perubahan-perubahan tersebut dipengaruhi oleh waktu, umur, kebiasaan dan keadaan mental tertentu [2].

#### **2.2 Citra Digital**

Citra Digital merupakan gambar dua dimensi yang bisa ditampilkan pada layar computer sebagai himpunan atau diskrit nilai digital yang disebut pixel/ picture elements. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi [12].

##### **2.2.1 Cropping**

Cropping merupakan mengubah ukuran besarnya citra kedalam pixel. Adakalanya ukurannya berubah menjadi lebih kecil dari file aslinya dan sebaliknya[13]. Resize mengubah resolusi atau ukuran horizontal dan vertical suatu citra (gambar).

### 2.2.2 RGB

RGB adalah suatu model warna yang terdiri atas 3 buah warna yaitu: merah (Red), hijau (Green), dan biru (Blue), yang ditambahkan sebagai cara untuk menghasilkan bermacam-macam warna. Model RGB ini merupakan model yang banyak digunakan, salah satunya adalah monitor. Pada model ini untuk merepresentasikan gambar menggunakan 3 buah komponen warna tersebut. Selain model RGB terdapat 3 komponen yaitu,  $r, g, b$  yang merepresentasikan presentase dari sebuah piksel pada citra digital.

### 2.2.3 Grayscale

Dalam fotografi dan komputasi, grayscale atau greyscale dari suatu gambar digital adalah gambar dimana nilai setiap pixel sampel tunggal, yang memiliki informasi intensitas. Gambar ini yang juga dikenal sebagai hitam-putih, secara eksklusif terdiri dari warna abu-abu, bervariasi dari hitam di intensitas paling lemah sampai putih di intensitas terkuat [14]. Pada grayscale nilai intensitas tersebut dapat diseragamkan dengan suatu fungsi. Berikut ini adalah rumus konversi citra berwarna (RGB) menjadi nilai intensitas grayscale [15].

$$G = wR.Red + wG.Green + wB.Blue \quad (2.1)$$

Keterangan :

$G(x,y)$  = Matriks grayscale koordinat piksel.

Red = komponen nilai merah (Red) dari suatu titik piksel

Green = komponen nilai hijau (Green) dari suatu titik piksel

Blue = komponen nilai biru (Blue) dari suatu titik piksel

$wR, wG, wB$  = bobot setiap nilai elemen warna.

Persamaan 2.1 di atas merupakan salah satu rumus yang digunakan untuk mengkonversi citra berwarna menjadi grayscale. Pada citra grayscale, setiap piksel adalah representasi derajat intensitas atau keabuan. Terdapat 256 jenis derajat keabuan pada citra grayscale. Mulai dari putih, kemudian semakin gelap sampai

warna hitam. Oleh karena terdapat kemungkinan 256 derajat keabuan, maka setiap piksel pada grayscale disimpan 1 byte dalam memory (8 bits).

### 2.3 Moment Invariant

Moment invariant ini digunakan sebagai fitur ekstraksi citra. Moment dapat memberikan karakteristik suatu objek yang secara unik merepresentasikan bentuknya. Pengenalan bentuk invariant dilakukan dengan klasifikasi dalam ruang fitur moment invariant multi dimensi. Beberapa teknik telah dikembangkan tentang penurunan fitur invariant dari moment objek untuk representasi dan pengenalan objek. Teknik ini dibedakan oleh definisi momentnya. Seperti jenis data yang dieksplotasi dan metode untuk menurunkan nilai invariant dari moment citra [16].

Moment Invariant terdapat 7 nilai descriptor bentuk, yang dihitung dari moment pusat melalui tiga derajat yang bebas terhadap translasi, skala dan arah objek [16]. Invariant translasi dicapai dengan menghitung moment yang dinormalisasi dengan pusat gravitasi sehingga pusat dari masa distribusi berada pada moment pusat. Moment Invariant ukuran diturunkan dari invariant aljabar tapi moment ini dapat diperlihatkan dari hasil penyederhanaan momen ukuran. Dari nilai order dua dan tiga moment pusat yang ternormalisasi, 7 moment invariant dapat dihitung yang juga bebas rotasi. Moment digunakan untuk membentuk moment invariant yang didefinisikan secara kontinu namun untuk implementasi praktis, moment dihitung secara diskrit. Diberikan sebuah fungsi  $f(x,y)$ , moment didefinisikan oleh gambar 2.2:

$$M_{pq} = \iint x^p y^q f(x,y) dx dy \quad (2.3)$$

$M_{pq}$  merupakan moment dua dimensi dari fungsi  $f(x,y)$ . Order moment adalah  $(p + q)$  dimana  $p$  dan  $q$  adalah bilangan asli. Untuk implementasi di dalam bentuk digital, maka persamaan ini menjadi:

$$M_{pq} = \sum_x \sum_y x^p y^q f(x,y) \quad (2.4)$$

Untuk menormalisasi invariant translasi dalam bidang citra, centroid citra digunakan menentukan moment pusat. Koordinat pusat grafitasi dari citra dihitung dengan menggunakan persamaan (2.4) dan diberikan oleh:

$$\bar{x} = \frac{M_{10}}{M_{00}} \quad \bar{y} = \frac{M_{01}}{M_{00}} \quad (2.5)$$

Selanjutnya, moment pusat dapat ditentukan secara diskrit seperti berikut:

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q \quad (2.6)$$

Moment selanjutnya dinormalisasi untuk efek perubahan skala dengan menggunakan rumus sebagai berikut:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^\gamma} \quad (2.7)$$

Dimana factor normalisasi  $\gamma = \left(\frac{p+q}{2}\right) + 1$ . Dari moment pusat normalisasi, 7 nilai dapat dihitung dan ditentukan dengan:

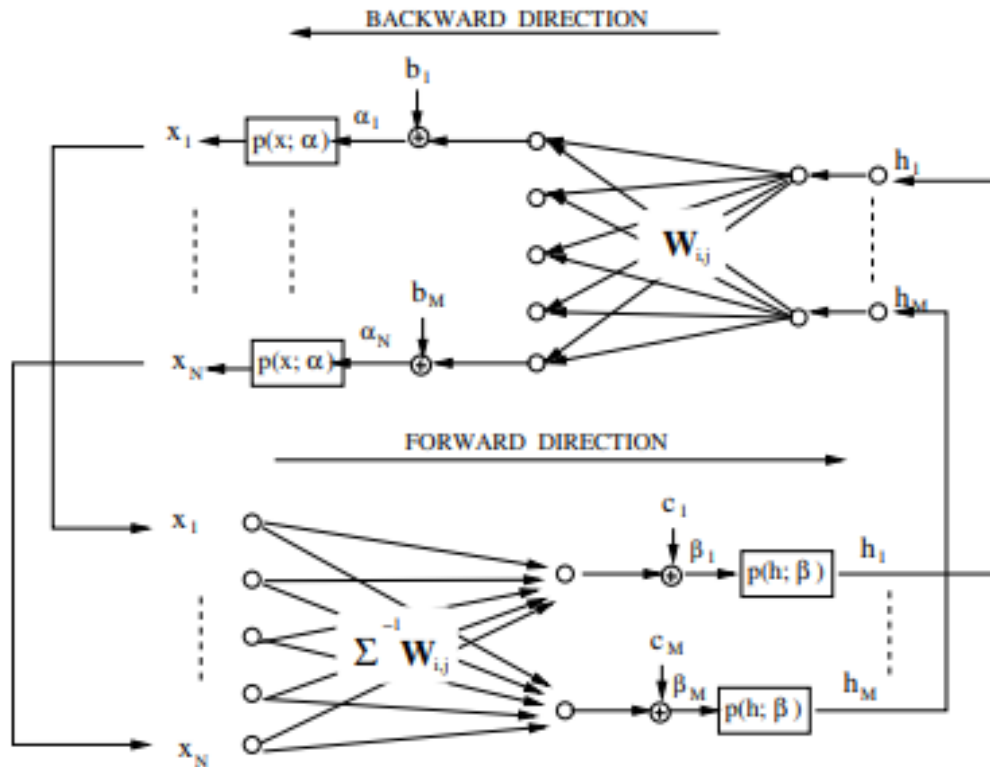
$$\begin{aligned} \phi_1 &= \eta_{20} + \eta_{02} \\ \phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\ \phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (\eta_{03} - 3\eta_{21})^2 \\ \phi_4 &= (\eta_{30} - \eta_{12})^2 - (\eta_{03} + \eta_{21})^2 \\ \phi_5 &= (3\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\ &\quad (\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\ \phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} - \eta_{12})(\eta_{21} + \eta_{03}) \\ \phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{12} - \eta_{30})(\eta_{21} \\ &\quad + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \dots\dots\dots (2.8) \end{aligned}$$

Tujuh moment invariant ini,  $\phi_i, i = 1, 2, \dots, 7$  dilakukan oleh Hu [16], diperlihatkan menjadi bebas terhadap rotasi. Akan tetapi nilai-nilai tersebut dihitung berdasarkan batasan luar (*boundary*) dan daerah bagian dalam (*interior region*).

## 2.4 Restricted Boltzmann Machine

Pada algoritma Restricted Boltzmann Machine ini digunakan sebagai klasifikasi citra setelah di ekstraksi oleh metode moment invariant. RBM merupakan jaringan saraf yang bersifat *stochastic* (jaringan saraf yang berarti memiliki unit neuron berupa aktivasi biner yang bergantung pada neuron-neuron yang saling terhubung, sedangkan *stochastic* berarti aktivasi yang memiliki unsur probabilistik) yang terdiri dari dua binary unit yaitu visible layer merupakan state yang akan diobservasi dan hidden layer merupakan feature detectors serta unit bias[8]. Selanjutnya masing-masing visible unit terhubung ke semua hidden unit yang diwakili oleh array bobot, sehingga setiap hidden unit juga terhubung ke semua visible unit dan unit bias terhubung ke semua visble unit dan semua hidden unit. Untuk memudahkan proses pembelajaran, jaringan dibatasi sehingga tidak ada visible unit terhubung ke visible unit lain dan hidden unit terhubung dengan hidden unit lainnya.

Dalam melatih RBM, sampel data yang digunakan sebagai masukan untuk RBM melalui neuron visible, dan kemudian jaringan sampel bolak-balik antara neuron visible dan hidden. Tujuan dari pelatihan adalah untuk pembelajaran koneksi bobot pada visible atau hidden dan bias aktivasi neuron sehingga RBM belajar untuk merekonstruksi data input selama fase dimana sampel neuron visible dari neuron hidden. Setiap proses sampling pada dasarnya berupa perkalian matriks-matriks antara sekumpulan sampel pelatihan dan matriks bobot, diikuti dengan fungsi aktivasi neuron, yaitu fungsi sigmoid (persamaan 2.9). Sampling antara lapisan hidden dan visible diikuti oleh modifikasi parameter (dikontrol oleh learning rate) dan diulang untuk setiap kelompok data dalam training set, dan untuk state sebanyak yang diperlukan untuk mencapai konvergensi.



**Gambar 2.1** Arsitektur Restricted Boltzmann Machine

Arsitektur neural network RBM dapat dilihat pada gambar 2.1 jaringan ini terdiri atas 2 (dua) unit neuron pada lapisan *visible unit* ( $x_1, x_2, \dots, x_n$ ), 3 (tiga) neuron pada lapisan *hidden unit* ( $h_1, h_2, h_3, \dots, h_n$ ) dan 1 (satu) neuron bias pada lapisan *visible* dan 1 (satu) neuron bias pada lapisan *hidden*.

## 2.5 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 [24]. Pengujian seharusnya meliputi tiga konsep berikut.

1. Demonstrasi validitas perangkat lunak pada masing-masing tahap siklus pengembangan sistem.

2. Penentuan validitas sistem akhir dikaitkan dengan kebutuhan pemakai.
3. Pemeriksaan perilaku sistem dengan mengeksekusi sistem pada data sampel pengujian.

Pada dasarnya pengujian diartikan sebagai aktiitas yang dapat atau hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian seharusnya dilakukan dalam skala lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik dipengujian. Kategori pengujian dapat dikategorikan menjadi dua [24], yaitu :

1. Berdasarkan ketersediaan logik sistem, terdiri dari *Black box testing* dan *White box testing*.
2. Berdasarkan arah pengujian, terdiri dari Pengujian *top down* dan Pengujian *bottom up*.

### 2.5.1 Akurasi

Akurasi Menunjukkan kedekatan hasil pengukuran data sesungguhnya, presisi menunjukkan seberapa dekat perbedaan nilai pada saat dilakukan pengulangan pengukuran. Untuk mengukur hasil akurasi dari Restricted Boltzmann Machines (RBM). Maka dilakukan perhitungan dengan rumus berikut.

$$Akurasi = \frac{Jumlah\ yang\ diklasifikasi\ secara\ benar}{Total\ sample\ testing\ yang\ diuji} \times 100\% \quad (2.13)$$

## 2.6 Alat-alat dan Pemodelan Sistem

Pemodelan sistem dapat berarti menyusun suatu sistem yang baru untuk menggantikan sistem yang lama secara keseluruhan atau memperbaiki sistem yang sudah ada. Berikut adalah pemodelan yang digunakan dalam membangun sebuah sistem pada penelitian ini.

### 2.6.1 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James

Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam p-code (bytecode) dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (general purpose), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin [18]. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda. Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi.

### **2.6.2 Netbeans**

NetBeans adalah Integrated Development Environment (IDE) berbasiskan Java dari Sun Microsystems yang berjalan di atas Swing. Swing sebuah teknologi Java untuk pengembangan aplikasi Desktop yang dapat berjalan di berbagai macam platforms seperti Windows, Linux, Mac OS X and Solaris. Suatu IDE adalah lingkup pemrograman yang diintegrasikan kedalam suatu aplikasi perangkat lunak yang menyediakan pembangun Graphic User Interface (GUI), suatu text atau kode editor, suatu compiler atau interpreter dan suatu debugger. Netbeans merupakan software development yang Open Source, dengan kata lain software ini di bawah pengembangan bersama, bebas biaya NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra. Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama. Saat ini terdapat dua produk: NetBeans IDE dan NetBeans Platform. The NetBeans IDE adalah sebuah lingkungan pengembangan sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program.

Netbeans IDE ditulis dalam Java, namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE.



Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Tersedia juga NetBeans Platform sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan plug-in bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam Platform dan dapat juga digunakan untuk membuat kaskas dan solusi sendiri. Kedua produk adalah kode terbuka (open source) dan bebas (free) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi Common Development and Distribution License (CDDL). Kedua kelebihan dan kekurangan Netbeans kelebihan NetBeans GUI Builder: Salah satu yang menjadi kelebihan NetBeans GUI Builder adalah yang telah disebutkan diatas, yaitu gratis. Selain itu NetBeans GUI Builder sangat kompetebel dengan Swing karena memang langsung dikembangkan oleh Sun yang notabenenya sebagai pengembang Swing.

Kekurangan NetBeans GUI Builder: NetBeans hanya mensupport 1 pengembangan Java GUI, yaitu Swing, yang padahal ada Java GUI yang dikembangkan oleh eclipse yang bernama SWT dan JFace yang sudah cukup populer. NetBeans mempatenkan source untuk Java GUI yang sedang dikerjakan dalam sebuah Generated Code, sehingga programmer tak dapat mengeditnya secara manual..

### **2.6.3 UML (Unified Modeling Language)**

UML (*Uniefied Modeling Language*) adalah notasi yang lengkap untuk membuat visualisasi model suatu sistem. Sistem berisi informasi dan fungsi, tetapi secara normal digunakan untuk memodelkan sistem komputer. Di dalam pemodelan objek guna menyajikan sistem yang berorientasi objek kepada orang lain, akan sangat sulit dilakukan dalam bentuk kode bahasa pemrograman [27].

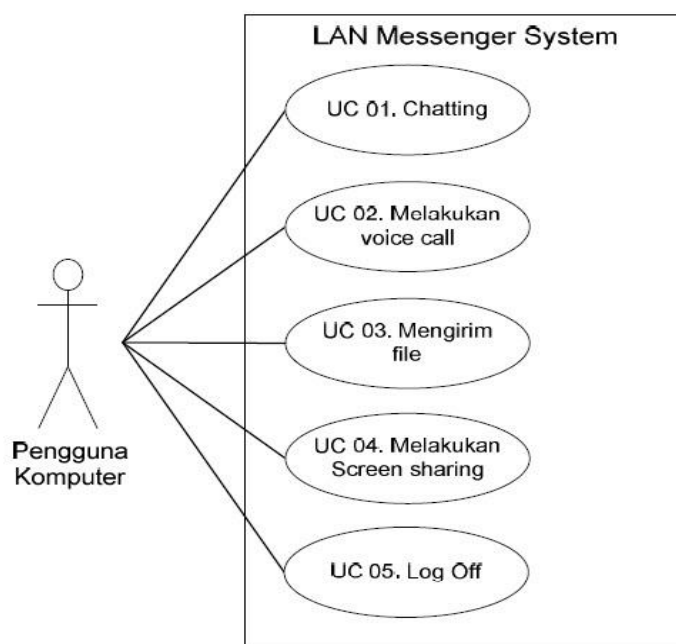
UML disebut sebagai bahasa pemodelan bukan metode. Bahasa pemodelan (sebagian besar grafik) merupakan notasi model yang digunakan untuk mendesain secara cepat. Bahasa pemodelan merupakan bagian terpenting dari metode. UML merupakan bahasa standar untuk penulisan *blueprint software* yang digunakan untuk 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 antar kelas. UML terdiri dari banyak diagram diantaranya *use case*, *diagram*, *activity diagram*, *class diagram*, dan *sequence diagram*.

#### 2.6.4 Use Case Diagram

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

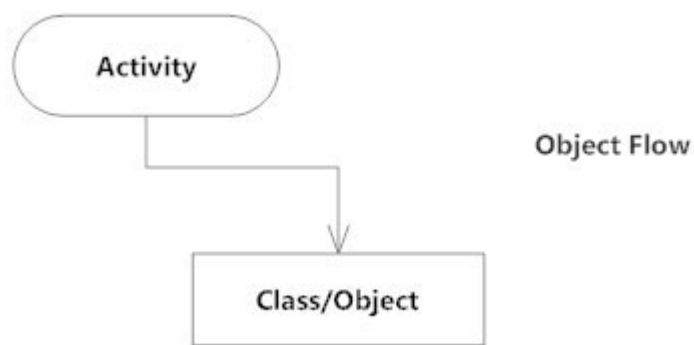
*Use case diagram* merupakan deskripsi lengkap tentang interaksi yang terjadi antara para aktor dengan sistem[27]. Dalam hal ini, setiap objek yang berinteraksi dengan sistem merupakan aktor untuk sistem, sementara *use case* merupakan deskripsi lengkap tentang bagaimana sistem berperilaku kepada aktornya. Aktor dalam *use case diagram* 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 2.2 Use Case Diagram

### 2.6.5 Activity Diagram

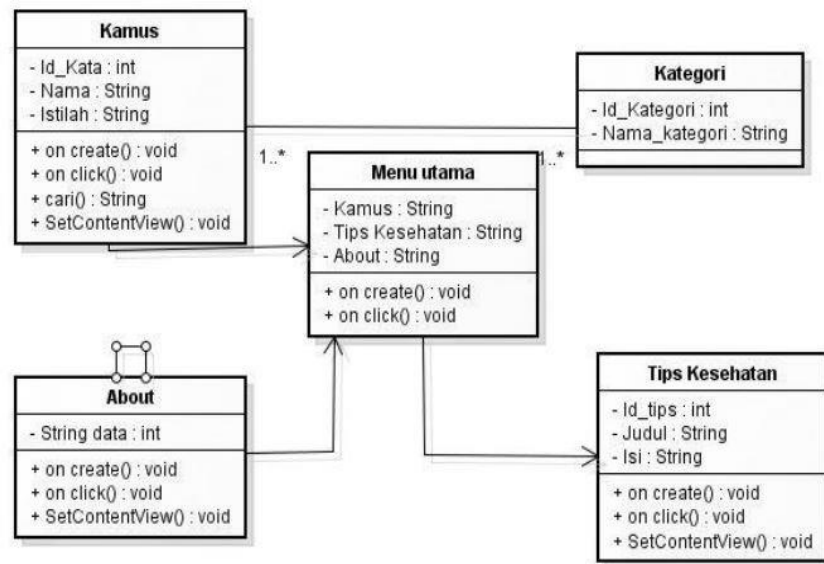
*Activity Diagram* pada dasarnya menggambarkan skenario secara grafis. *Activity diagram* cukup serupa dengan diagram alur (*flow chart*), yang membedakan hanya *swimlane* yang menunjukkan suatu *state* berada pada objek/kelas tertentu. Keunggulan dari *activity diagram* adalah lebih mudah dipahami dibanding skenario. Selain itu, dengan menggunakan *activity diagram*, kita dapat melihat dibagian manakah sistem dari suatu skenario akan berjalan [27].



**Gambar 2.3 Activity Diagram**

### 2.6.6 Class Diagram

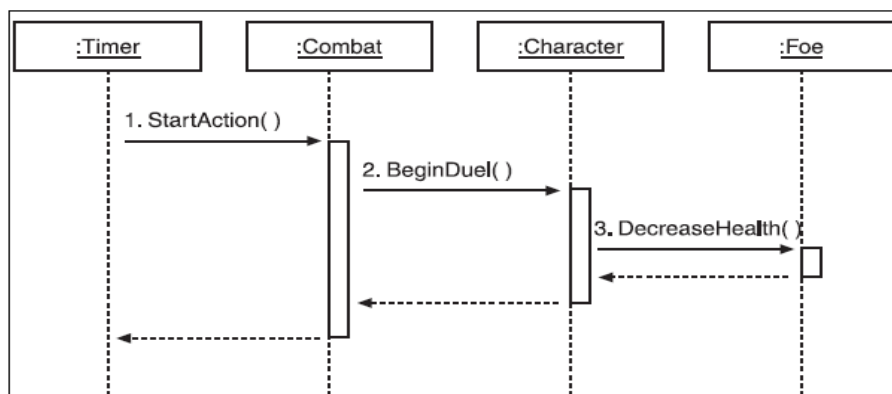
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. *Class diagram* menggambarkan struktur dan deskripsi kelas, package, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi dan sebagainya.



**Gambar 2.4 Class Diagram**

### 2.6.7 Sequence Diagram

Diagram sekuen menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek[27]. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.



**Gambar 2.5 Sequence Diagram**

