

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Wanadri

Profil instansi ini dimaksudkan untuk mengetahui bagaimana keadaan dari penelitian yang dilakukan di Wanadri Bandung mulai dari sejarah, motto, visi dan misi, logo, serta struktur organisasi.

2.1.1 Sejarah Wanadri

Wanadri merupakan perhimpunan Penempuh Rimba dan Pendaki Gunung Wanadri didirikan pada tanggal 17 Januari 1964 oleh 6 orang pemuda “bekas PANDU”, yang kemudian dikenal dengan sebagai Angkatan Pendiri. Harry Hardiman Soebari (FE Unpad -W-001 Pen), Ronny Nurzaman (ITB -W-002 Pen), Bambang Pramono (Unpar-W-003 Pen), Satria Widjaja Somantri (W-004 Pen), Eddy Achmad Fadilah, dan Achmad Hidayat.

Nama Wanadri sendiri berasal dari bahasa Sanskerta, wana yang berarti hutan dan adri yang berarti gunung. Wanadri secara harafiah berarti gunung di tengah-tengah hutan. Pada dasarnya, Wanadri adalah kumpulan sekelompok orang yang mencintai kehidupan di alam bebas. Organisasi Wanadri yang dapat dipandang sebagai suatu “masyarakat Wanadri” adalah organisasi yang memiliki aturan dan norma yang khas, baik tertulis maupun tidak tertulis, yang senantiasa berlaku dan dihormati oleh anggotanya. Dengan berbekal visi akan pemuda-pemudi Indonesia yang tangguh, tabah, serta percaya pada kekuatan diri sendiri, Wanadri kemudian diresmikan pada tanggal 16 Mei 1964. Bersamaan dengan peresmian tersebut dikukuhkan juga angkatan kedua dalam organisasi Wanadri, yakni Angkatan Pelopor yang terdiri dari 25 anggota.

2.1.2 Motto

Motto Wanadri adalah Tak ada gunung yang tinggi, rimba belantara, jurang curam dan lautan serta angkasa yang tak dapat dijelajahi oleh Wanadri.

2.1.3 Visi dan Misi Wanadri

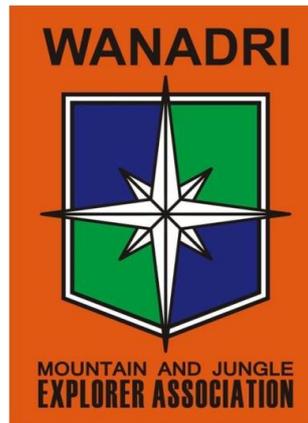
Visi Wanadri adalah menjadi sebuah organisasi pendidikan untuk mendidik manusia, khususnya anggotanya, agar mempunyai nilai-nilai yang terkandung dalam Janji dan Hakikat Wanadri.

Adapun Misi dari Wanadri sebagai berikut :

1. Membentuk manusia yang Mandiri, Ulet dan Tabah.
2. Menjadikan manusia yang berpegang teguh terhadap Pancasila.
3. Percaya kepada kekuatan diri sendiri.

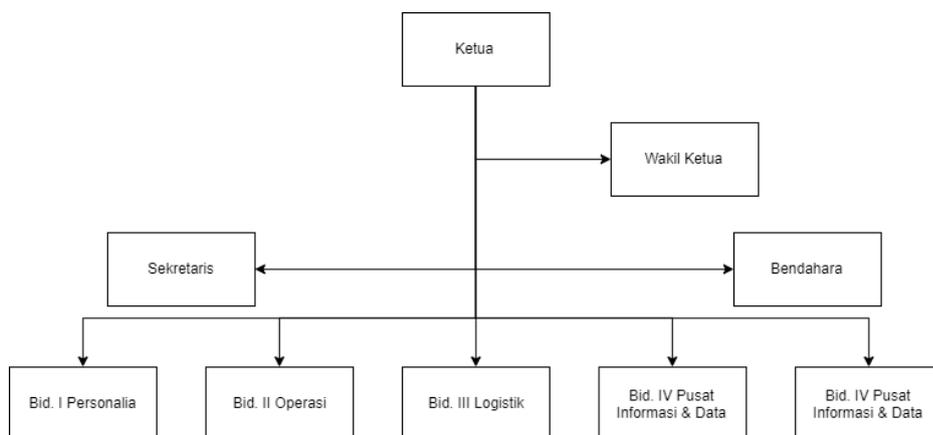
2.1.4 Logo Wanadri

Logo Wanadri dapat dilihat pada gambar 2.1.



Gambar 2. 1 Logo Wanadri

2.1.5 Struktur Organisasi



Gambar 2. 2 Struktur Organisasi Wanadri

2.2 Landasan Teori

Landasan teori adalah penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan sistem yang akan dibangun. Konsep dan teori tersebut diantaranya Gamifikasi, *Six Steps To Gamification*, *Android*, *Firestore*, Standar Operasional Prosedur, *multimedia*, OOP dan UML.

2.2.1 Gamifikasi

Gamifikasi adalah penggunaan elemen-elemen *game* dan teknik *design game* dalam konteks *non-game*. Elemen *game* yaitu seperti poin, lencana, tingkatan, narasi dan sebagainya, tapi seiring perkembangannya inti dari gamifikasi saat ini adalah bagaimana untuk membangun motivasi [5]. Belakangan ini, gamifikasi digunakan secara luas di bidang *non game*/hiburan, melainkan digunakan juga untuk pendidikan dan dunia bisnis. Misalnya, aplikasi *game* yang disertakan pada *Tab* atau *iPad* untuk membantu anak-anak dalam belajar berhitung dan menulis sehingga sangat membantu orang tua dalam mengajar anak-anak mereka matematika dan menulis.

Gamifikasi *Framework* merupakan kumpulan kode *program* yang bersifat *universal* digunakan oleh pengembang untuk memudahkan gamifikasi yang akan dibangun. Gamifikasi *Framework* terdiri dari kumpulan program, *class library*, *API*, dan komponen pendukung lainnya yang memungkinkan pengembang untuk mengembangkan perangkat lunak dengan menggunakan *Framework* ini.

Gamifikasi digunakan untuk menarik dan memotivasi orang untuk menggunakan produk dan gamifikasi juga digunakan untuk mempengaruhi perilaku. Penggunaan Gamifikasi sangat efektif untuk mengubah cara pekerjaan yang membosankan, menjadi jauh lebih menyenangkan untuk dilakukan. Penggunaan gamifikasi dapat diterapkan menggunakan berbagai gamifikasi *framework* tergantung kepada tipe gamifikasi seperti apa yang akan dibangun.

2.2.2 Six Steps to Gamification

Metode 6D (*Six steps to Gamification*) adalah sebuah metode yang dikembangkan oleh Werbach and Hunter dan merepresentasikan tahapan-tahapan gamifikasi dalam sebuah *framework* yang mempunyai awalan huruf D yaitu :

1. *DEFINE* (Menunjukkan tujuan yang spesifik)

Untuk membuat sebuah gamifikasi yang baik maka diperlukan untuk membuat tujuan spesifik dari produk yang akan dibuat. Ada pun untuk langkah-langkah berikutnya dari define:

- a. Membuat semua daftar kemungkinan tujuan.
 - b. Menurutkan dan memilih tujuan yang paling atas dari daftar yang sudah dibuat.
 - c. Pastikan tujuan yang dibuat itu benar-benar penting.
2. *DELINEATE* (Menggambarkan perilaku yang ingin dicapai dengan Gamifikasi)

Menentukan seperti apa hasil yang diinginkan bagi user kedepannya setelah menggunakan aplikasi tersebut, dan bagaimana cara kita untuk mengukur *user* yang telah menggunakan aplikasi tersebut. Ada pun untuk langkah-langkah berikutnya dari *delineate* :

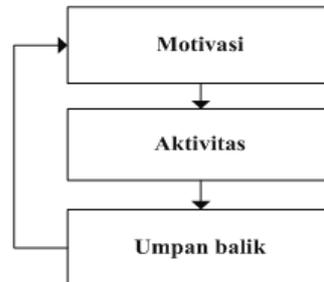
- a. Menentukan apa saja yang akan dilakukan oleh user.
 - b. Memprioritaskan siapa yg akan menggunakan aplikasi ini.
 - c. Membuat poin-poin dari untuk setiap yang dilakukan user.
3. *DESCRIBE* (Menjelaskan seperti apa *user* Gamifikasi)

Didalam *describe* harus menjelaskan *user* yang akan menggunakan aplikasi ini seperti skill dalam bermain *game* apakah *user* terbiasa dalam bermain *game* atau biasa saja bahkan tidak terbiasa. Berikan fitur-fitur untuk *user* yang akan bermain *game* ini untuk menaikkan *mood user*, contoh seperti Fitur kemajuan dan prestasi *user*. Yang paling penting seperti memberikan *nickname* dan sebuah *avatar* untuk *user*.

4. *DEVISE* (Merencanakan siklus Sistem Gamifikasi)

Aktivitas pengguna disebabkan adanya motivasi yang selanjutnya menghasilkan sebuah umpan balik dari sistem, seperti pemberian poin. Umpan balik tersebut memotivasi pengguna untuk melakukan aktivitas selanjutnya, begitu seterusnya. Umpan balik merupakan salah satu bagian yang membuat *game* begitu efektif sebagai motivator. Setiap aktivitas akan menghasilkan umpan balik yang dapat langsung diterima pengguna. Secara *virtual*, semua komponen *game* dapat

dilihat sebagai bentuk umpan balik. Sebagai contoh, penggunaan poin dapat digunakan sebagai umpan balik untuk menunjukkan performa pengguna.



Gambar 2. 3 Siklus aktivitas pada *engagement loops*

5. *DON'T FORGET THE FUN* (Harus melibatkan aspek yang menyenangkan)

Poin ke-lima adalah *don't forget the fun*, jadi *user* harus menjadi salah satu aspek menyenangkan, aspek tersebut adalah sebagai berikut :

- a. *Hard Fun* – menyenangkan untuk menyelesaikan (ketagihan).
- b. *Easy Fun*
- c. *Experimental Fun* – mencoba sesuatu hal yang baru, mempelajari hal yang baru tersebut.
- d. *Social Fun* – berinteraksi dengan yang lain.

6. *DEPLOY* (Membangun *tools* yang akan diterapkan sistem Gamifikasi)

Dalam poin terakhir ini semua yang direncanakan haruslah ada dalam proses pembangunan *tools* yang akan diterapkan sistem gamifikasi, seperti karakter, achievement, leaderboard dsb, sehingga aplikasi tersebut akan memenuhi semua poin-poin yang sebelumnya.[6].

2.2.3 Android

Android adalah sistem operasi yang berbasis Linux untuk telepon seluler seperti telepon pintar dan komputer tablet. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk ponsel.

2.2.3.1 Versi Android

Versi Android yang telah dikeluarkan hingga 20 Oktober 2019 adalah sebagai berikut :

1. Android versi 1.1

Pertama kali platform Android diluncurkan pada tahun 2009, setahun sebelumnya Google telah merilis versi beta yang diperkenalkan kepada khalayak ramai. Android versi pertama dikenal dengan Android 1.1. Berikut adalah fitur-fitur yang tersedia:

- a. Mampu menyimpan *attachment* dalam pesan
- b. Waktu default *screen in-call* lebih lama bila menggunakan *speakerphone* plus kemampuan untuk menampilkan atau menyembunyikan dialpad.

2. Android versi 1.5 (Cupcake)

Pada tanggal 27 April 2009 Google juga merilis Android versi Cupcake. Cupcake adalah versi Android yang memulai tradisi penamaan kue untuk rilis Android. Ini dapat menambahkan beberapa fitur dan peningkatan baru dibanding versi sebelumnya.

Berikut adalah fitur-fitur yang tersedia:

- a. Dukungan untuk tampilan aplikasi *widget-miniatur* yang bisa disematkan di aplikasi lain (seperti *home screen*) dan menerima update berkala
- b. Rekaman video ditambahkan ke kamera bersamaan dengan kemampuan untuk langsung mengupload *video* ke YouTube.

3. Android versi 1.6 (Donut)

Google merilis Android 1.6 Donut pada bulan September tahun 2009. Penambahan fitur terbesar adalah dukungan untuk CDMA yang memperluas pasar pengguna baru ke Android. CDMA adalah teknologi yang digunakan oleh jaringan mobile Amerika pada saat itu..

Berikut adalah fitur-fitur yang tersedia:

- a. Navigasi *Google Maps* ditambahkan bersamaan dengan dukungan navigasi satelit.
- b. Donut menyertakan fitur *gallery* untuk memperlancara penangkapan media
- c. Pengenalan *Search Box*

d. *Toggling* cepat antara kamera

4. Android versi 2.0 - 2.1 (Éclair)

Pada bulan Oktober 2009, sekitar setahun setelah peluncuran Android 1.0, Google merilis versi 2.0 dari OS, dengan nama Android Eclair. Versi ini adalah pertama yang menambahkan dukungan Text-to-Speech.

Berikut adalah fitur-fitur yang tersedia:

- a. Dukungan multi-touch ditambahkan ke Android.
- b. Dukungan untuk mencari di dalam pesan teks.
- c. Eclair membawa kotak masuk terpadu ke Android. Dukungan untuk beberapa akun Google telah ditambahkan.

5. Android versi 2.2.3 (Froyo)

Android versi terbaru kembali dirilis pada 20 Mei 2010. Google menamainya dengan Froyo. Pada android versi ini mulai dilengkapi dengan fitur *friendly* user seperti opsi untuk mematikan akses data pada jaringan seluler.

Berikut adalah fitur-fitur yang tersedia:

- a. *Hotspot WiFi portabel* untuk berbagi koneksi 3G perangkat dengan *gadget* lainnya.
- b. Setelan bergabung dengan kontak dan email untuk memback up ke *server* Google yang memungkinkan pembaca mengembalikan segalanya secara otomatis ke perangkat baru.
- c. *Flash* telepon juga bisa digunakan dalam *video*.

6. Android versi 2.3 - 2.3.7 (Gingerbread)

Gingerbread dirilis pada tahun 2010. Pada 13 September 2010, Google menunjukkan bahwa hanya 0,6 persen dari semua perangkat Android yang saat ini menjalankan beberapa versi Gingerbread. Fitur utama termasuk dukungan *NFC*, *SIP* untuk panggilan Internet.

Berikut adalah fitur-fitur yang tersedia:

- a. *UI overhaul* untuk menghindari screen *burn-in* dan meningkatkan daya tahan baterai.

- b. Dukungan kamera menghadap depan untuk panggilan *video*.
- c. *Download manager* untuk mengawasi *download* Pembaca.
- d. Peningkatan pada *keyboard* layar dengan cara pintas dan kursor untuk membantu *copy paste*.

7. Android versi 3.0 - 3.2.6 (Honeycomb)

Honeycomb diluncurkan pada bulan Mei 2011 pada dasarnya untuk memperluas Android untuk mendukung layar tablet. Versi Android ini paling diabaikan dari semua. Karena dirilis khusus untuk tablet dan tidak pernah sampai ke ponsel.

Berikut adalah fitur-fitur yang tersedia:

- a. Beberapa perbaikan UI memanfaatkan layar besar.
- b. Tombol perangkat keras dijatuhkan untuk mendukung tombol di layar.
- c. *Browser web* mengenalkan *tabbed browsing*.
- d. *Widget* yang lebih besar.
- e. Aplikasi seperti Gmail dan YouTube dirancang ulang untuk menggunakan layar besar.

8. Android versi 4.0 - 4.0.4 (Ice Cream Sandwich)

Android Ice Cream Sandwich dirilis pada bulan Oktober 2011, versi Android Ice Cream Sandwich menghadirkan sejumlah fitur baru bagi pengguna. Ini menggabungkan banyak fitur versi Honeycomb tablet saja dengan smartphone yang berorientasi pada Gingerbread.

Berikut adalah fitur yang menyertainya:

- a. Desain ulang terbesar ke Android dengan tema Holo.
- b. Pengalaman *browsing* lebih cepat.
- c. Ruang penyimpanan multi untuk aplikasi.
- d. Pengenalan wajah untuk membuka kunci telepon.

9. Android versi 4.1 - 4.3.1 (Jelly Bean)

Jellybean dimulai pada bulan Juni 2012 dengan merilis Android 4.1. Google dengan cepat merilis versi 4.2 dan 4.3, keduanya berada di bawah label Jelly Bean, masing-masing pada bulan Oktober 2012 dan Juli 2013.

Berikut adalah fitur yang menyertainya:

- a. *Google Now, tool* bantu yang menampilkan informasi yang relevan berdasarkan riwayat pencarian.
- b. *Project Butter* untuk mendukung *frame rate* yang lebih tinggi saat menggesek menu dan layar rumah.
- c. Mampu melihat foto dengan menggesek dari kamera untuk menuju ke *filmstrip*.
- d. *Widget* menyetel ulang diri mereka untuk menambahkan yang baru.
- e. Pemberitahuan fitur yang lebih banyak.
- f. Fitur gerak dan aksesibilitas baru.

10. Android versi 4.4 (Kitkat)

Android KitKat dirilis pada November 2013 ,Android 4.4 adalah satu-satunya versi OS yang benar-benar menggunakan nama sepotong permen.dan menjadi salah satu versi Android yang paling disukai oleh pengguna Smartphone di dunia.

KitKat memiliki fitur yang istimewa dari OS Android sebelumnya.

Berikut adalah fitur yang menyertainya:

- a. *Immersive mode* untuk konsumsi konten yang lebih baik.
- b. *Bar navigasi* yang lebih baik untuk masuk dan keluar dari mode *Immersive*.
- c. Dukungan widget layar kunci.
- d. Dialer baru dengan fitur *Caller ID*.
- e. Wallpaper layar penuh.
- f. Emoji keyboard untuk *emoticon*.
- g. Aplikasi *Hangouts* dan perpesanan terpadu
- h. Dukungan *cloud print* yang lebih baik.
- i. Integrasi *Google Now* yang lebih cerdas dan *handsfree*.

11. Android versi 5.0 (Lollipop)

Android 5.0 Lollipop pertama kali diperkenalkan pada Mei 2014.,Android lollipopmerupakan perancangan ulang terbesar untuk

Android. Smartphone Google Nexus 6, bersama dengan tablet Nexus 9-nya, merupakan perangkat pertama yang memiliki Lollipop yang telah terpasang sebelumnya. Berikut adalah fitur yang dimilikinya:

- a. Dukungan pengaturan cepat yang lebih baik.
- b. Masa pakai baterai yang disempurnakan dengan mode *Battery*
- c. Layar kunci baru.
- d. Fitur *Smart Lock* melalui Layanan Google Play.
- e. Mode tamu untuk berbagi perangkat.
- f. Pemasangan tombol.

12. Android versi 6.0 (Marshmallow)

Android 6.0 (Marshmallow) Di rilis pada tahun 2015. Ini Perangkat pertama yang dikirim bersama Marshmallow yang telah terpasang sebelumnya adalah smartphone Google Nexus 6P dan Nexus 5X, dengan tablet Pixel C-nya. Tujuan marshmallow memoles sudut kasar dan membuat versi Lollipop lebih baik lagi.

Berikut adalah fitur yang dimilikinya:

- a. Dukungan sidik jari resmi untuk perangkat.
- b. Dukungan untuk pembayaran seluler melalui Android *Pay*.
- c. Model perizinan yang lebih baik untuk aplikasi.
- d. Google *Now* di Tap.
- e. *Deep* menghubungkan Apps.

13. Android versi 7.0 (Nougat)

Android 7.0 (Nougat) Dirilis pada Tahun ,2016. Sebelum Nougat terungkap "Android N" dirujuk secara internal oleh Google sebagai "New York Cheesecake".

Berikut adalah fitur yang dimilikinya:

- a. Doze on the Go untuk waktu siaga yang lebih baik lagi.
- b. Multi Window untuk penggunaan dua aplikasi secara bersamaan.
- c. Aplikasi Setelan yang Lebih Baik.
- d. Hapus semua di layar aplikasi baru-baru ini.

- e. Balas langsung ke pemberitahuan.
- f. Notifikasi dibundel.
- g. Pengaturan Cepat akan mengubah kustomisasi.

14. Android versi 8.0 (Oreo)

Pada bulan Maret 2017, Google Rilis Android 8.0 Oreo, bulan Agustus, Google mengkonfirmasi Oreo akan menjadi nama publik untuk Android 8.0. Adapaun sekarang versi ini adalah semua yang baru yang dimilikinya, diantaranya adalah sebagai berikut:

- a. Pemberitahuan untuk prioritas dan kategorisasi yang lebih baik.
- b. Pengelolaan warna lebih baik.
- c. Android O memiliki koleksi emoji baru yang telah didesain ulang.
- d. Waktu boot lebih cepat: Pada perangkat Pixel, sekarang bisa mengalami waktu *boot* dua kali lebih cepat dibandingkan dengan Nougat.
- e. Mengisi otomatis dan mengingat kata sandi dalam aplikasi.

Versi android yang akan digunakan dalam pembangunan aplikasi *Bushcraft* adalah android versi 5.0.

2.2.4 *Firestore*

Firestore adalah *API* yang disediakan google untuk penyimpanan dan penyelarasan data ke dalam aplikasi Android, iOS, atau *web*. *Realtime database* adalah salah satu fasilitas yang menyimpan data ke *database* dan mengambil data darinya dengan sangat cepat tetapi *Firestore* bukan hanya *realtime database*, jauh lebih dari itu. *Firestore* memiliki banyak fitur seperti *authentication*, *database*, *storage*, *hosting*, pemberitahuan dan lain-lain[14].

Dalam penelitian ini *Firestore* digunakan sebagai penyimpanan data anggota wanadri dari mulai data personal, materi, dan soal latihan.

2.2.5 *Standard Operating Procedure (SOP)*

SOP (Standard Operating Procedure) pada dasarnya merupakan pedoman yang berisi prosedur-prosedur operasional standar yang ada di dalam suatu organisasi yang digunakan untuk memastikan bahwa semua keputusan dan

tindakan, serta penggunaan fasilitas-fasilitas proses yang dilakukan oleh orang-orang di dalam organisasi yang merupakan anggota organisasi agar berjalan efektif dan efisien, konsisten, standar dan sistematis.[13]

Sehingga SOP menjadi sebuah mekanisme yang vital, apabila tidak dirancang dan dilaksanakan dengan baik, maka kegiatan organisasi tersebut kemungkinan besar akan menjadi kacau. SOP yang digunakan dalam penelitian ini berasal dari buku diktat yang digunakan di Wanadri.

2.2.6 *Object Oriented Programming (OOP)*

Object Oriented Programming (OOP) adalah istilah yang terdapat dalam bahasa pemrograman dengan menggunakan teknik yang berorientasi atau berbasis pada objek pada pembangunan sistem atau program aplikasi, dengan makna lain bahwa orientasi dalam pembuatan program tidak lagi menggunakan orientasi linear atau struktural melainkan berorientasi pada objek-objek yang terpisah [9].

2.2.7 *Unified Modeling Language (UML)*

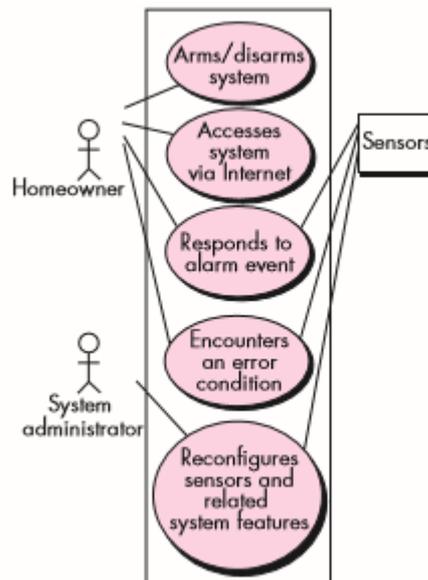
Unified Modeling Language (UML) merupakan bahasa pemodelan visual yang digunakan dalam menentukan, memvisualisasikan, membangun, dan mendokumentasikan artefak dalam sebuah sistem perangkat lunak yang akan dibangun [3]. UML digunakan dalam untuk memahami sebuah rancangan, konfigurasi dan memelihara kontrol dalam sistem informasi. UML menangkap informasi tentang struktur statis dan perilaku dinamis untuk suatu sistem. Suatu sistem yang dimodelkan sebagai kumpulan objek diskrit yang berinteraksi untuk melakukan pekerjaan yang pada akhirnya menguntungkan pengguna. UML bukan merupakan sebuah bahasa pemrograman namun merupakan alat yang menyediakan sebuah *generator* kode dari UML ke berbagai bahasa pemrograman.

Pada dasarnya, setiap orang memiliki pendapat masing-masing mengenai UML. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang bangun perangkat lunak efektif [7].

2.2.7.1 *Use Case Diagram*

Use Case merupakan salah satu bagian dalam UML yang menggambarkan *external view* dari sistem yang akan kita buat modelnya. Pooley mengatakan bahwa

model *use case* dapat dijabarkan dalam diagram *use case*, tetapi perlu diingat, diagram tidak identik dengan model karena model lebih luas dari *diagram* [8]. *Use case* bekerja dengan cara mendeskripsikan tipe interaksi antara *user* sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai serta dapat digunakan untuk membentuk perilaku sistem yang akan dibuat.



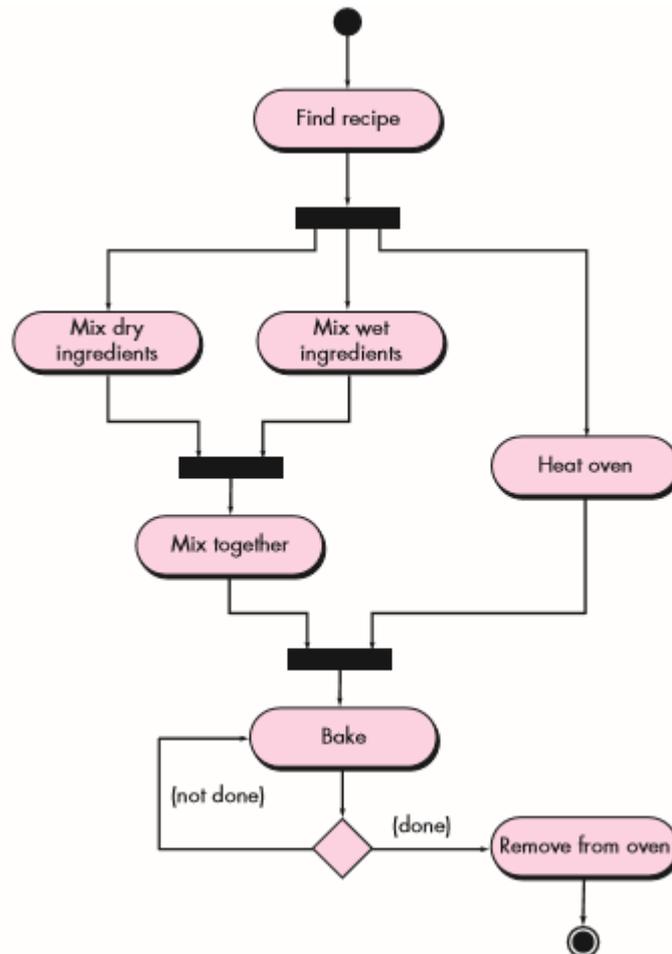
Gambar 2. 4 Contoh *Use Case*

2.2.7.2 Activity Diagram

Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan state diagram khusus, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*).

Oleh karena itu *activity diagram* tidak menggambarkan *behavior internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari *level* atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktivitas proses bisnis.

Struktur *diagram* ini mirip *flowchart* atau *Data Flow Diagram* pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *use case* pada *use case diagram*.



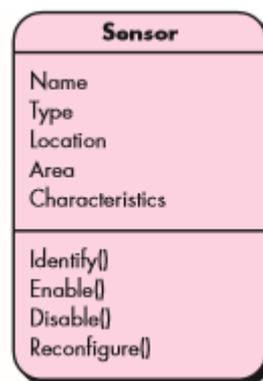
Gambar 2. 5 Contoh *Activity Diagram*

2.2.7.3 *Class Diagram*

Pemodelan berbasis kelas pada dasarnya memperlihatkan objek-objek yang akan dimanipulasi oleh sistem/perangkat lunak, memperlihatkan operasi-operasi (juga dinamakan metode-metode atau layanan-layanan) yang akan diterapkan pada objek-objek untuk menghasilkan imbas tertentu pada manipulasi objek, memperlihatkan relasi-relasi antar objek (yang beberapa di antaranya bersiat

bersifat hierarkis), serta memperlihatkan kolaborasi-kolaborasi yang terjadi di antara kelas-kelas yang didefinisikan.

Elemen-elemen model berbasis kelas mencakup di dalamnya elemen-elemen kelas-kelas dan objek-objek, atribut-atribut, operasi-operasi, model tanggung jawab kelas (*Class Responsibility Collaborator* (CRC]), diagram-diagram kolaborasi, dan paket-paket (packages) [11].

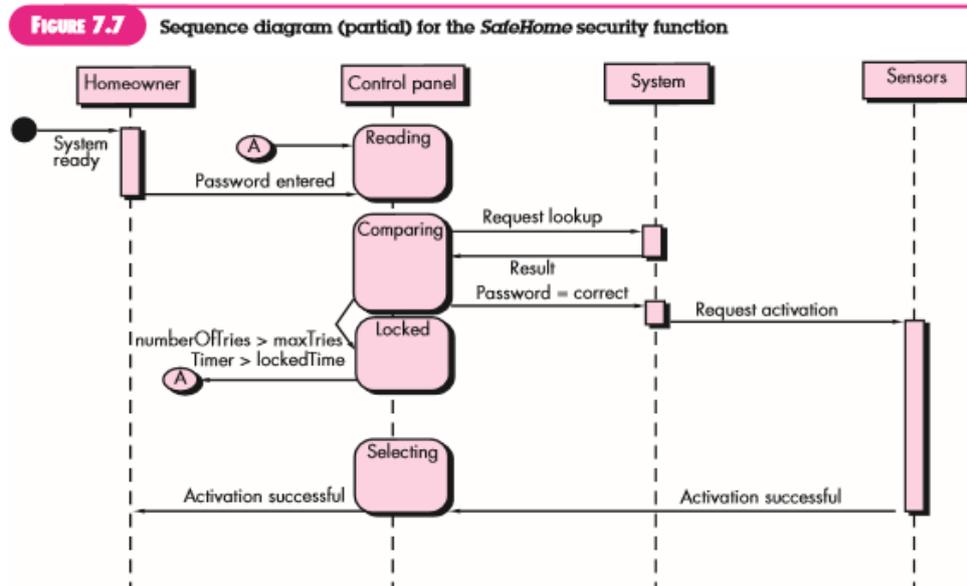


Gambar 2. 6 Contoh *Class Diagram*

2.2.7.4 *Sequence Diagram*

Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi *vertikal* (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline vertikal*. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya.



Gambar 2. 7 Contoh *Sequence Diagram*

2.2.8 Adobe Photoshop

Adobe Photoshop adalah *software* grafis berbasis *Bitmap* (pixel). Yang biasa dipakai untuk mengedit foto, membuat ilustrasi bahkan desain *web*. Sehingga banyak digunakan di studio foto, percetakan, *production house*, biro arsitektur, pabrik tekstil dan bidang yang berkaitan dengan Teknologi Informasi (IT).[16]

Software sejenis antara lain *Paint for Windows*, *Gimp*, *CorelPaint*, *Photo Impact*, *Photostudio*, *Photostyler*, *ACDSee*, *Paintbrush* dll. Namun semuanya kalah populer dibanding *Photoshop*. Sehingga umumnya desainer profesional wajib menguasai *Adobe Photoshop*, dalam penelitian ini *photoshop* digunakan dalam membuat asset *background* dan *button*.

2.2.9 Pengujian *Blackbox*

Black-box testing atau pengujian kotak hitam adalah cara atau metode untuk menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode *program*. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan, dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan. Pengujian *Black-box* dilakukan dengan membuat kasus uji yang bersifat mencoba semua fungsi dengan memakai perangkat lunak apakah sesuai

dengan spesifikasi yang dibutuhkan. Kasus uji yang dibuat untuk melakukan pengujian *Black-box* harus dibuat dengan kasus benar dan kasus salah.[15]

Tabel 2. 1 Contoh pengujian *Black-box*

No	Data Masukan	Yang Diharapkan	Hasil Yang Keluar	Keterangan
1	Pilih button(tombol) Login (id: btnLogin)	Login berhasil dan menampilkan menu anggota / pemateri	Login berhasil dan menampilkan menu anggota / pemateri	[√] Diterima [] Ditolak
2	Pilih button(materi) Materi (id: btnMateri)	Berhasil memilih dan menuju materi yang dipilih	Berhasil memilih dan menuju materi yang dipilih	[√] Diterima [] Ditolak
3	Pilih button(latihan) Latihan (id: btnLatihan)	Berhasil memilih dan menuju menu Latihan	Berhasil memilih dan menuju menu Latihan	[√] Diterima [] Ditolak
4	Pilih button(Achievement) Achievement (id:btnAchievement)	Berhasil memilih menu achievement dan menampilkan achievement yang telah didapatkan	Berhasil memilih menu achievement dan menampilkan achievement yang telah didapatkan	[] Diterima [√] Ditolak
5	Pilih button(Tambah Data) Tambah Data (id:btnTambahdata)	Berhasil menambahkan data anggota baru	Berhasil menambahkan data anggota baru	[√] Diterima [] Ditolak
6	Menampilkan Score (id:score)	Berhasil menampilkan score saat melakukan latihan	Berhasil menampilkan score saat melakukan latihan	[√] Diterima [] Ditolak

Pengujian *black-box* sendiri berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya :

- a. fungsi-fungsi yang salah atau hilang
- b. kesalahan interface
- c. kesalahan dalam struktur data atau akses database eksternal
- d. kesalahan performa
- e. kesalahan inisialisasi dan terminasi.