

BAB 2

LANDASAN TEORI

2.1 Profil Instansi

Menurut Ketua Saung Budaya, Muhammad Hafidz Saung Budaya merupakan tempat atau sanggar budaya yang bertempat pada salah satu Universitas dikota bandung tepatnya Universitas Komputer Indonesia. Sadaya didirikan pada tanggal 25 Februari 2008. Selain itu tujuan dari saung budaya yaitu tersedianya seni pertunjukan tradisional sebagai sarana apresiasi masyarakat, meningkatkan pneranan seni tradisional. Dengan visi yaitu memperkenalkan kebudayaan dan kesenian tradisional dan melestarikan dan menjadi wadah pembinaan dan pengembangan seni musik dan seni tradisional. Dan mempunyai misi yaitu sebagai generasi muda penerus bangsa yang mempunyai kewajiban dalam memajukan budaya Indonesia, mengajarkan seni musik dan seni tari tradisional. Sadaya sendiri mempunyai 5 divisi, terdiri dari divisi angklung, divisi tari, divisi rampak kendang, divisi perkusi, dan divisi gamelan.

2.1.1 Logo Instansi

Logo dari Saung Budaya ditunjukkan pada gambar 2.



Gambar 2.1 Logo Saung Budaya
(Sumber : AD&ART SADAYA UNIKOM)

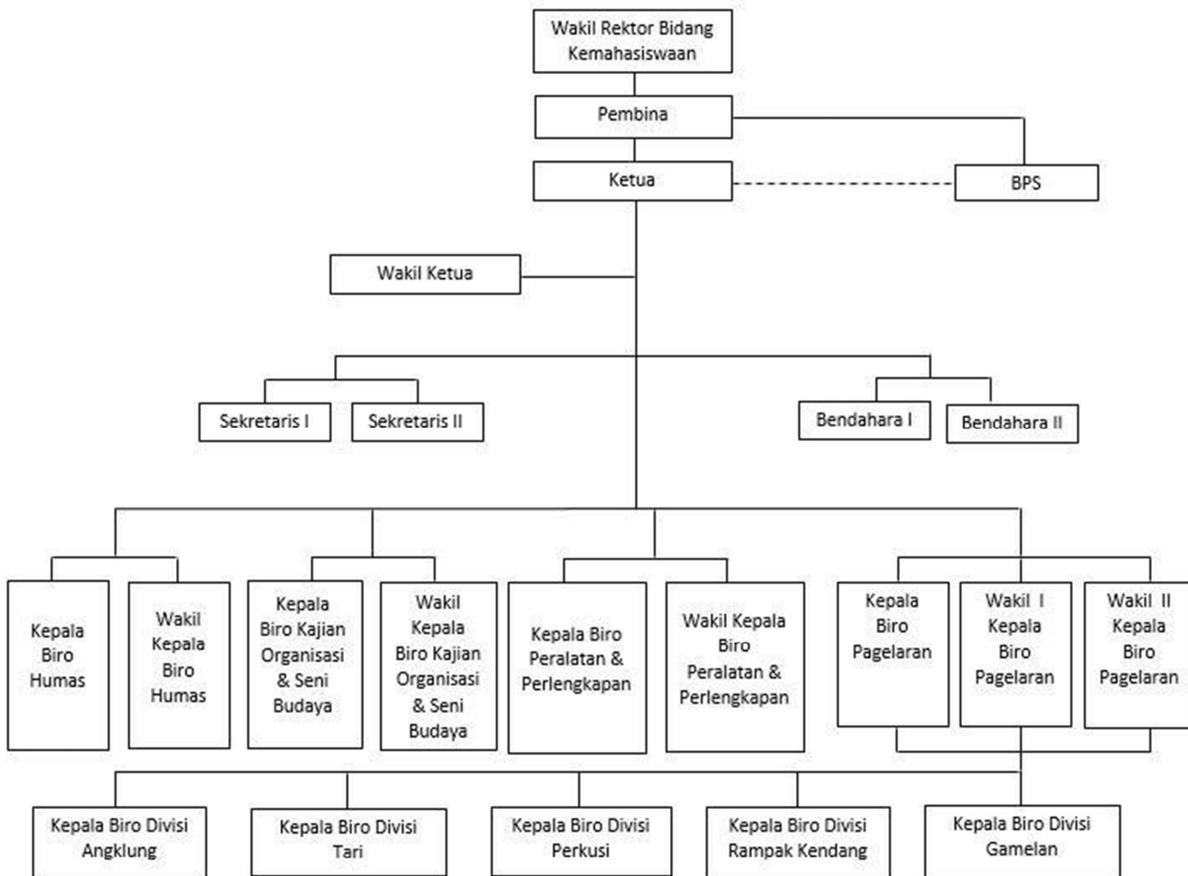
Arti dari logo sadaya :

1. Kata SADAYA merupakan singkatan dari Saung Budaya dimana kata Saung diambil dari petikan Bahasa Sunda yang berarti sebuah Gubuk atau tempat dimana orang – orang melakukan sebuah kegiatan.
2. Lingkaran hitam melambangkan ruang lingkup persaudaraan yang merangkul budaya Indonesia

3. Dua orang penari menggambarkan bahwa putra putri Indonesia memiliki ciri khas dalam seni tradisional.
4. Gong dan tifa menggambarkan suatu ciri khas alat musik tradisional Indonesia.
5. Merah dan putih menggambarkan suatu rasa nasionalisme terhadap bangsa dan tanah air.
6. Bintang mewakili martabat bangsa Indonesia dalam melestarikan seni dan budaya tradisional dimata dunia.
7. Warna emas melambangkan suatu kedudukan tertinggi dalam menumbuhkan nasionalisme.

2.1.2 Struktur Organisasi

Struktur organisasi dari Saung Budaya ditunjukkan pada gambar 2.2



Gambar 2.2 Struktur Organisasi Saung Budaya

(Sumber : AD&ART SADAYA UNIKOM)

2.2 Landasan Teori

Berikut adalah teori – teori yang digunakan dan yang berhubungan dengan penelitian yang dilakukan.

2.2.1 Noise

Menurut Agus kurniawan, Noise merupakan suatu sinyal pengganggu yang menyebabkan suatu sinyal rusak ataupun terganggu. Gejala – gejala alam yang menimbulkan kebisingan misalnya badai listrik, semburan api matahari (*solar flare*) dan sabuk – sabuk radiasi tertentu diruang angkasa. Satu – satunya cara yang efektif untuk mengurangi kebisingan semacam itu adalah dengan penempatan dan pengarahannya kembali antena penerima dimana untuk menerima kebisingan tersebut seminimal mungkin, sementara sinyal yang diterima tidak banyak berkurang. Selain itu ada juga sumber – sumber kebisingan alami, dalam peralatan – peralatan elektronik, sumber ini dikatakan mendasar karena merupakan bagian yang tak dapat dihindari. Kebisingan tersebut ternyata tunduk pada hukum – hukum fisika tertentu, dan pengertian ini memungkinkan dirancangnya peralatan – peralatan, dimana pengaruh kebisingan dapat dibuat seminimum mungkin. Agar sinyal informasi yang digunakan dapat dideteksi dan dihasilkan kembali dengan memuaskan pada sisi penerima dari suatu sistem, maka daya dari suatu sinyal yang dikehendaki tersebut haruslah lebih besar dari pada gangguan berisik (*noise*) yang ada.

2.2.2 Multimedia

Menurut Vaughan, Multimedia adalah kombinasi dari sebuah text, gambar, suara, animasi, dan video yang disampaikan kepada kita dan dimanipulasi secara digital. Elemen – elemen multimedia yaitu :

1. Text.

Setiap kata dapat memiliki beberapa makna, oleh sebab itu saat akan menggunakan text, penting untuk memperhatikan akurasi dan keringkasan pada kata yang akan dipilih. Kata dan simbol dalam bentuk apapun, baik tertulis ataupun lisan adalah bentuk yang paling umum dalam komunikasi. Mereka menyampaikan makna yang paling akurat dan detail. Karena itu mereka menjadi aspek penting pada elemen multimedia seperti judul, menu, navigasi dan konten.

2. Gambar

Gambar biasanya dapat diubah ke berbagai ukuran, diwarnai, diberi pola, dibuat transparan, ditempatkan didepan atau dibelakang objek, dan dibuat terlihat maupun tidak terlihat. Gambar terdiri dari dua tipe, yaitu bitmap yang biasanya disebut raster, digunakan untuk foto realistic dan gambar sulit yang memerlukan detail. Tipe lainnya yaitu vector yang

digunakan untuk garis, kotak, lingkaran, polygon, dan bentuk – bentuk lainnya yang dapat diekspresikan secara matematis seperti jarak, kordinat dan sudut.

3. Suara

Suara adalah elemen multimedia yang paling sensual. Suara merupakan cara berkomunikasi yang berarti pada setiap Bahasa, mulai dari bisikan sampai teriakan. Bagaimana seseorang menggunakan kekuatan dari suara dapat membuat perbedaan dari presentasi multimedia yang biasa – biasa saja dan presentasi yang spektakuler.

4. Animasi

Animasi adalah objek yang benar – benar bergerak melewati kedalam maupun keluar dari layar. Animasi membuat suatu presentasi menjadi hidup.

5. Video

Video adalah elemen multimedia yang dapat mengejutkan penonton pada sebuah acara atau menarik perhatian murid pada pelajaran berbasis komputer. Video digital adalah elemen multimedia yang paling menarik dan merupakan alat untuk membawa pengguna komputer lebih dekat ke dunia nyata. Dari semua elemen multimedia, video adalah yang paling banyak ‘meminta’ *resource* dari komputer melalui memori, *storage*, dan performa (Vaughan, 2010).

2.2.3 Android

Menurut Abdul Ajiz, Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, Konsorsium dari perusahaan – perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat selular. Berikut versi yang telah dikeluarkan oleh Android :

1. Alpha

Dalam hal ini (Android 1.0) merupakan versi pertama dari sistem operasi Android oleh Google. Ini mempunyai fungsi dasar dengan browser sederhana dan aplikasi Google seperti Gmail, Maps, dan Youtube.

2. Beta

Android 1.1 beberapa fungsi ditambahkan, API berubah dari level 1 pada android 1.0 menjadi level 2. Mendukung lampiran dengan MMS. Android versi Beta ini diluncurkan pada tanggal 5 November 2007, sedangkan untuk

software Development Kit (SDK) nya diluncurkan pada tanggal 12 November 2007.

3. Cupcake

Cupcake adalah versi kedua Android dengan fitur baru serta API kerangka kerja Android yang diperbarui. Cupcake diluncurkan pada tanggal 30 April 2009. Pembaruan pada versi ini termasuk beberapa fitur baru dan UI.

4. Donut

Versi Android 1.6 dijuluki sebagai Donut. Dalam versi ini Google menambahkan dukungan teknologi baru CDMA/EVDO, dukungan layar WVGA dan mesin sintesis pengucapan yang memungkinkan aplikasi android tertentu mampu mengucapkan text.

5. Éclair

Android versi 2.0-2.1 muncul dengan nama panggilan sebagai Éclair. Dengan lebih banyak fungsi dan fitur. Éclair ini diluncurkan dengan fitur Bluetooth 2.1, wallpaper hidup, dukungan HTML 5, kemampuan untuk mencari sms dan mms, dukungan flash, zoom digital, dan fitur kamera lainnya.

6. Froyo

Android versi 2.2 – 2.3 diperkenalkan dengan fungsi dan aplikasi USB tethering dan WIFI hotspot, dukungan adobe flash, peningkatan kecepatan dan kinerja aplikasi dengan fitur – fitur baru.

7. Gingerbread

Android versi 2.3 – 2.7 diperkenalkan dengan antarmuka pengguna yang diperbarui yang memberikan lebih banyak kemudahan untuk digunakan. Fiturnya seperti sensor, beberapa kamera (depan dan belakang), keyboard virtual, saran text yang lebih baik, kemampuan input suara dan tekan terus kemampuan copy paste.

8. Honeycomb

Android Honeycomb ini dirancang untuk layar besar seperti tablet sehingga elemen antarmuka seperti keyboard virtual dioptimalkan untuk layar yang lebih besar. Layar beranda dioptimalkan, tab diperkenalkan di browser dengan metode penyamaran tambahan dan obrolan video dan talk didukung.

9. Ice Cream Sandwich

Android versi ini hadir pada tahun 2011, membawa semua tampilan baru, memberikan kemudahan lebih kepada pengguna seperti pengguna dapat dengan cepat menggesek untuk menutup aplikasi, tata letak galeri baru dan editor foto built in.

10. Jelly Bean

Google membuat sistem operasi lebih responsif dengan Jelly Bean dan memperkenalkan berbagai fitur dengan Android Bream. Keunggulan utama dari versi ini adalah antarmuka pengguna yang lebih halus.

11. Kitkat

Android kitkat hadir pada tahun 2013 dengan API level 19. Kitkat ini memiliki pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan, webview berbasis chromium dan peningkatan tampilan mode layar penuh.

12. Lollipop

Android versi 5.0-5.1 muncul dengan peningkatan RAM dan manajemen baterai. Desain antarmuka yang dinamakan Material Design. Selain itu versi yang satu ini juga memiliki fitur “factory reset protection” sehingga jika smartphone hilang, dia tidak akan bisa diriset ulang tanpa memasukan id google dan password.

13. Marshmallow

Android versi Marshmallow dirilis pada tahun 2015, muncul dengan baterai yang lebih cerdas dan mode tidur. Pengaturan privasi yang lebih baik, lebih mudah untuk meng-upgrade ponsel, sensor sidik jari, dan dibangun pesan suara visual.

14. Nouget

Android versi dibuat resmi pada tahun 2016 dengan emoji yang diperbarui, 72 tampilan multi jendela yang baru ditambahkan, baterai pintar dengan mode penghemat data, realitas visual yang lebih aman dan berkualitas tinggi dengan dimensi baru.

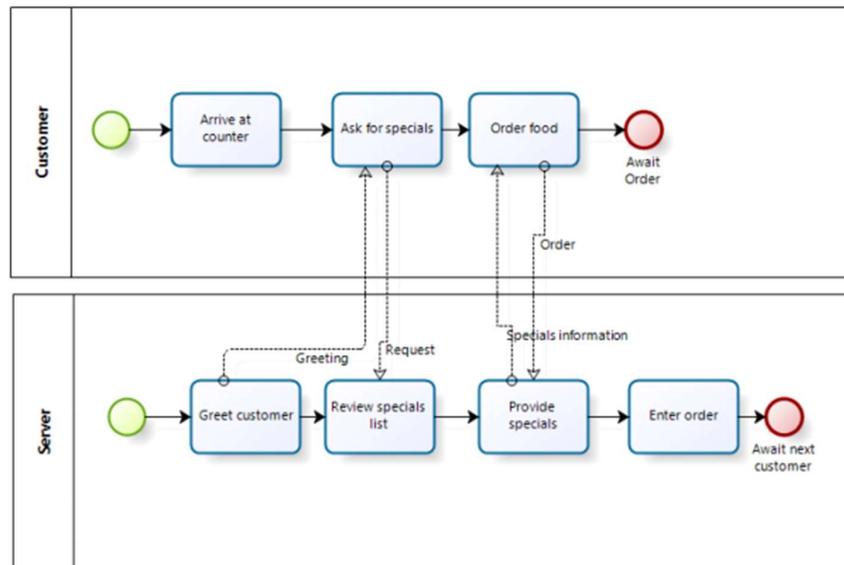
15. Oreo

Android versi oreo pertama kali rilis sebagai review pengembang pada tanggal 21 maret 2017. Android oreo mengusung fitur baru yang membuat smartphone lebih cepat, pintar dan powerful dibandingkan dengan versi android sebelumnya. Beberapa fitur unggulan versi ini adalah background limit, Google play protect, emoji baru, dan picture in picture. (Abdul Ajiz, 2018).

2.2.4 BPMN (*Business Process Modelling Notation*)

Menurut Rahmawati, BPMN merupakan singkatan dari *Business Process Modelling Notation*. *Business Process Modelling* yaitu suatu metodologi yang dikembangkan *Business Process Modelling Initiative* (BPMI) dalam memodelkan proses bisnis. (Rahmawati, 2017)

Gambar BPMN seperti yang ditunjukkan gambar 2.3



Gambar 2.3 Contoh BPMN
(Sumber : Rahmawati, 2017)

Terdapat empat kategori dasar yang ada pada BPMN modelling antara lain :

1. *Flow Object* terdiri dari *event*, *activities*, dan *gateway*.
2. *Connecting Objects* biasa digunakan untuk menghubungkan *flow object* melalui beberapa jenis *arrows*.
3. *Swimlanes* terdiri dari *pool* dan *lane*.
4. *Artifacts* dapat dimasukkan kedalam model dimana model tersebut dianggap sesuai dalam rangka untuk menampilkan informasi lebih lanjut terkait seperti data yang diproses atau komentar – komentar lain.

2.2.5 UML (Unified Modelling Language)

Menurut B. Hariyanto UML (*Unified Modelling Language*) adalah Bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, merupakan banyak level abstrak, tidak bergantung pada proses pengembangan, tidak bergantung Bahasa dan teknologi, pemanduan beberapa notasi diberagam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas – kakas yang diintegrasikan lewat XML (XMI). Standar umum UML dikelola oleh OMG (*Object Management Group*).

Tujuan utama perancangan UML adalah :

1. Menyediakan Bahasa pemodelan visual yang ekspresif dan siap pakai untuk mengembangkan dan pertukaran model – model yang berarti.

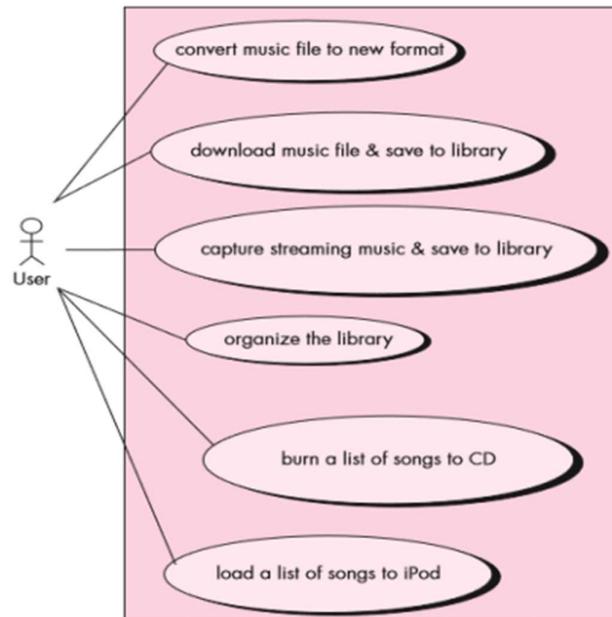
2. Menyediakan mekanisme perluasan dan spesifikasi untuk memperluas konsep – konsep inti.
3. Mendukung spesifikasi independen Bahasa pemrograman dan proses pengembangan tertentu.
4. Menyediakan basis formal untuk pemahaman Bahasa pemodelan.
5. Mendorong pertumbuhan pasar kaku berorientasi objek.
6. Mendukung konsep – konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, framework dan pattern.

Secara fundamental, UML berkaitan dengan penangkapan dan komunikasi pengetahuan.

Konsep – konsep yang diterapkan di UML adalah satu model berisi informasi mengenai sistem, model – model berisi elemen – elemen model seperti kelas – kelas, simpul – simpul, paket – paket, dan sebagainya. Satu diagram menunjukkan satu pandangan tertentu dari model. (B. Hariyanto, 2004).

2.2.5.1 Usecase Diagram

Menurut B. Hariyanto *Usecase Diagram* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem masing – masing diagram *usecase* menunjukkan sekumpulan *usecase* , aktor, dan hubungannya. Diagram *usecase* penting untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan kebutuhan perilaku sistem. Diagram – diagram *usecase* merupakan pusat pemodelan perilaku sistem, subsistem, dan kelas. Contoh *usecase diagram* dapat dilihat pada gambar 2.4



Gambar 2.4 Contoh Usecase Diagram

(Sumber : R.S. Pressman, 2010)

Usecase adalah interaksi antara aktor eksternal dan sistem, hasil yang dapat diamati oleh aktor, berorientasi pada tujuan, dideskripsikan di diagram *usecase* dan teks. Diagram *usecase* melibatkan :

1. Sistem yaitu sesuatu yang hendak dibangun.
2. Aktor, entitas – entitas luar yang berkomunikasi dengan sistem.
3. *Usecase* adalah fungsionalitas dipersepsi oleh aktor.
4. Reliasi adalah relasi antar aktor dengan *usecase*.

Diagram *usecase* digunakan untuk mendeskripsikan apa yang seharusnya dilakukan oleh sistem. Diagram *usecase* menyediakan cara mendeskripsikan pandangan eksternal terhadap sistem dan interaksi – interaksinya dengan dunia luar. Dengan cara ini, diagram *usecase* menggantikan diagram konteks pada pendekatan konvensional.

Pemodelan ini biasa dilakukan lewat proses berulang interaksi antara pengembang dan pemakai untuk memperoleh spesifikasi kebutuhan yang sama – sama disepakati. pemodelan *usecase* diciptakan oleh Ivar Jacobson. Tujuan utama pemodelan *usecase* adalah :

1. Memutuskan dan mendeskripsikan kebutuhan – kebutuhan fungsional sistem.
2. Memberikan deskripsi jelas dan konsisten dari apa yang seharusnya dilakukan, sehingga model *usecase* digunakan di seluruh proses

pengembangan untuk komunikasi dan menyediakan basis pemodelan berikutnya yang mengacu sistem harus memberikan fungsionalitas yang dimodelkan pada *usecase*.

3. Menyediakan basis untuk melakukan pengujian sistem yang memverifikasi sistem. Menguji apakah sistem telah memberikan fungsionalitas yang diminta.
4. Menyediakan kemampuan melacak kebutuhan fungsionalitas menjadi kelas – kelas dan operasi – operasi actual di sistem. Untuk menyederhanakan perubahan dan ekstensi ke sistem dengan mengubah model *usecase* dan kemudian melacak *usecase* yang dipengaruhi ke perancangan dan implementasi sistem.

Elemen pada diagram *usecase* adalah :

1. Aktor

Aktor merupakan pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. Aktor adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu siapa atau apa yang menggunakan sistem. Yang dimaksud dengan berinteraksi adalah aktor mengirim atau menerima pesan ke atau dari sistem, atau bertukar informasi dengan sistem. Dalam perspektif aktor, *usecase* melakukan sesuatu yang berharga bagi aktor. Aktor adalah tipe kelas bukan instan. Aktor mempresentasikan peran bukan pemakai individu dari sistem. Aktor mempunyai nama. Nama yang dipilih harus menyatakan peran aktor.

Usecase selalu diawali dengan aktor, yang mengirim pesan disebut stimulus. Ketika satu *usecase* dilakukan, *usecase* dapat mengirim pesan ke satu aktor atau lebih.

2. *Usecase*

Usecase adalah cara spesifik penggunaan sistem oleh aktor. Ciri –ciri dari *usecase* yaitu :

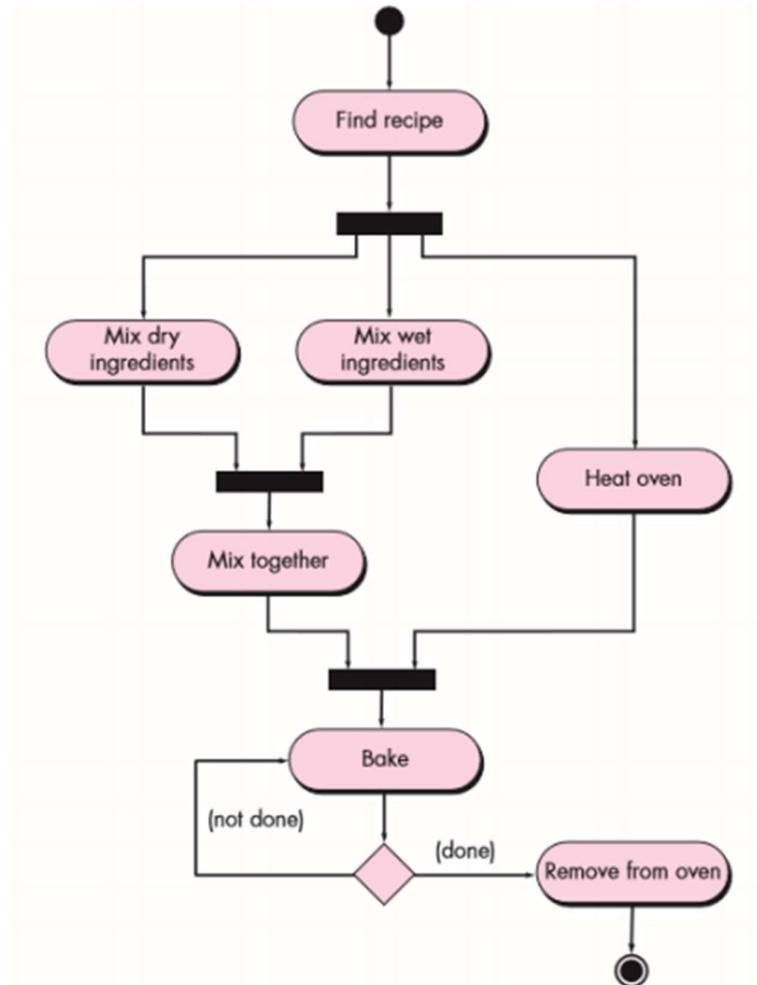
- a. Pola perilaku yang harus dipenuhi oleh sistem.
 - b. Sekuen transaksi terhubung yang dilakukan aktor dan sistem.
 - c. Memberikan sesuatu yang berharga bagi aktor.
3. Hubungan ketergantungan generalisasi dan asosiasi
Keterhubungan antara *usecase* dengan *usecase* lain berupa generalisasi antara *usecase* yaitu :
 - a. *Include*, perilaku *usecase* merupakan bagian dari *usecase* yang lain.
 - b. *Extend*, perilaku *usecase* memperluas perilaku *usecase* yang lain.
 (B. Hariyanto, 2004).

2.2.5.2 Activity Diagram

Menurut B. Hariyanto *Activity Diagram* adalah diagram *flowchart* yang diperluas, yang menunjukkan aliran kendali suatu aktivitas ke aktivitas

lain. Diagram ini digunakan untuk memodelkan aspek dinamis sistem. Aktivitas adalah eksekusi nonatonik yang berlangsung di *state machine*. Diagram aktivitas mendeskripsikan aksi – aksi dan hasilnya. Diagram aktivitas berupa operasi – operasi dan aktivitas – aktivitas di *usecase*. (B. Hariyanto, 2004).

Contoh *Activity Diagram* dapat dilihat pada gambar 2.5



Gambar 2.5 Contoh Activity Diagram

(Sumber : R.S. Pressman, 2010)

Diagram Aktivitas dapat digunakan untuk :

1. Pandangan dalam yang dilakukan di operasi
2. Pandangan dalam bagaimana objek – objek bekerja
3. Pandangan dalam di aksi – aksi dan pengaruhnya pada objek – objek

4. Pandangan dalam dari suatu *usecase*
5. Logika dari proses bisnis.

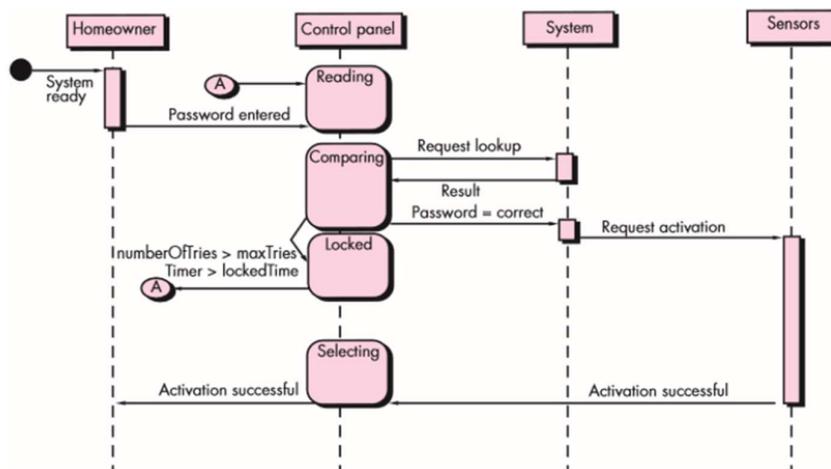
Diagram aktivitas merupakan jenis khusus dan diagram *statechart*. *State* adalah aksi – aksi yang menuju state berikutnya setelah selesai aksi sebelumnya.

Diagram berfokus pada aktivitas – aktivitas, potongan – potongan dari proses yang belah jadi (mungkin) berkorespondensi dengan metode – metode atau fungsi – fungsi anggota dan pengurutan dari aktivitas – aktivitas ini. Hal ini serupa dengan *flowchart*, namun diagram aktivitas berbeda terutama karena diagram aktivitas secara eksplisit mendukung aktivitas – aktivitas parallel dan sinkronisasi aktivitas – aktivitas tersebut.

2.2.5.3 Sequence Diagram

Menurut B. Hariyanto *Sequence Diagram* yaitu mendeskripsikan komunikasi diantara objek – objek meliputi pesan – pesan yang ada dan urutan pesan tersebut muncul.

Contoh *Sequence Diagram* dapat dilihat pada gambar 2.4.3



Gambar 2.6 Contoh Sequence Diagram

(Sumber : R. S. Pressman, 2010)

Diagram sekuen digunakan untuk :

1. *Overview* perilaku sistem.
2. Menunjukkan objek – objek yang diperlukan.
3. Mendokumentasikan scenario dari suatu diagram *usecase*
4. Memeriksa jalur – jalur pengaksesan.

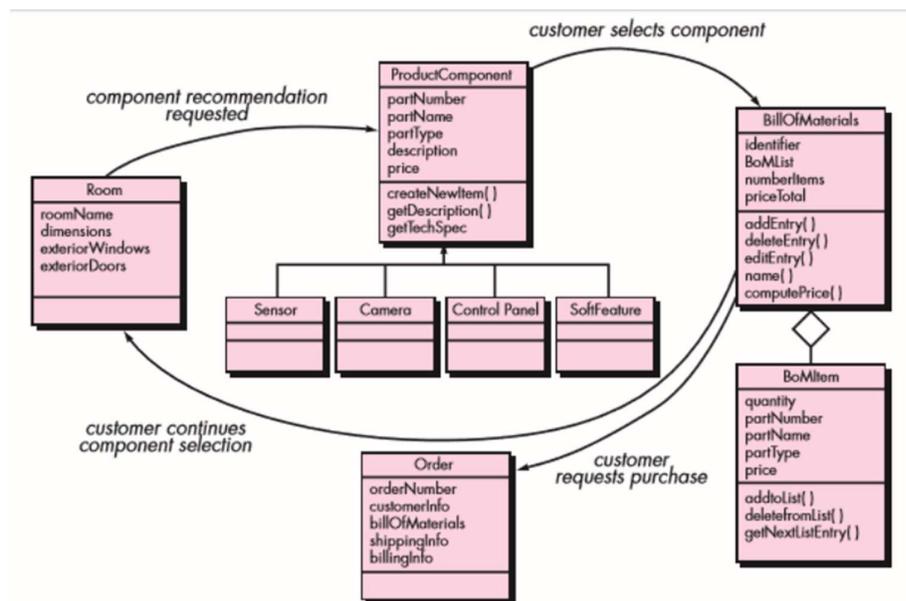
Diagram sekuen digunakan untuk memodelkan scenario penggunaan. Skenario penggunaan adalah barisan kejadian yang terjadi selama satu eksekusi sistem. Cakupan skenario dapat beragam, dari mulai semua kejadian di sistem atau hanya kejadian pada objek – objek tertentu. Skenario menjadi rekaman historis eksekusi sistem atau gagasan eksperimen eksekusi sistem yang diusulkan.

Diagram sekuen menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukkan barisak kejadian, bukan pewaktuan nyata. Kecuali untuk sistem waktu nyata yang mengharuskan konstrain barisan kejadian. (B. Hariyanto, 2004).

2.2.5.4 Class Diagram

Menurut B. Hariyanto *Class Diagram* merupakan diagram paling umum dipakai di semua pemodelan berorientasi objek. Pemodelan kelas merupakan pemodelan paling utama di pendekatan berorientasi objek. Pemodelan kelas menunjukkan kelas – kelas yang ada di sistem dan berhubungan antar kelas – kelas itu, atribut – atribut dan operasi – operasi di kelas – kelas. (B. Hariyanto, 2004).

Contoh *Class Diagram* dapat dilihat pada gambar 2.4.4



Gambar 2.7 Contoh Class Diagram

(Sumber : R.S. Pressman, 2010)

Diagram kelas menunjukkan aspek statik sistem terutama untuk mendukung kebutuhan fungsional sistem. Kebutuhan fungsional berarti layanan – layanan yang harus disediakan sistem ke pemakai. Meskipun diagram kelas serupa dengan model data, namun kelas – kelas tidak hanya menunjukkan struktur informasi tapi juga mendeskripsikan perilaku. Salah satu maksud diagram kelas adalah untuk mendefinisikan fondasi bagi diagram – diagram lain dimana aspek – aspek lain dari sistem ditunjukkan. Kelas diagram dapat secara langsung diimplementasikan di Bahasa pemrograman berorientasi objek yang secara langsung mendukung bentukan kelas.

Elemen – elemen esensi di diagram kelas adalah sebagai berikut :

1. Kelas
Kelas merupakan elemen terpenting si sistem berorientasi objek. Kelas mendeskripsikan satu blok pembangun sistem. Kelas memiliki sejumlah fitur, kita dapat memodelkan multiplisitas, ketampakan, penanda, polymorphism, dan karakteristik lain.
2. Antarmuka (*interface*)
Antarmuka merupakan koleksi operasi yang menspesifikasikan layanan dari suatu kelas atau komponen. Antarmuka mendeskripsikan perilaku tampak secara eksternal dari elemen.
3. Kolaborasi
Kolaborasi merupakan pendefinisian suatu interaksi dan sekelompok peran dan elemen – elemen lain yang bekerja sama untuk menyediakan suatu perilaku kooperatif yang lebih besar dari penjumlahan seluruh elemen. Kolaborasi memiliki struktur, perilaku dan dimensi. Suatu kelas dapat berpartisipasi di beberapa kolaborasi. Kolaborasi ini memperpresentasikan implementasi pola tertentu yang membentuk sistem.
4. Hubungan (*relationship*)
Hubungan antar kelas diagram beraneka ragam, yaitu :
 - a. Asosiasi
Asosiasi mempresentasikan hubungan antara insan – insan kelas. Interpretasi dari asosiasi bervariasi sesuai prespektifnya. Secara konseptual, asosiasi mempresentasikan hubungan antar kelas – kelas yang terlibat. Pada prespektif spesifikasi, terdapat tanggung jawab untuk mengetahui, dan akan dibuat eksplisit dengan operasi pengaksesan dan pembaruan. Sehingga terdapat pointer antara satu kelas dan kelas lainnya, tapi ini tersembunyi oleh pengkapsulan. Prespektif implementasi mengimplikasikan keberadaan pointer. Dengan demikian hal esensi untuk mengetahui prespektif yang digunakan saat membangun model agar menginterpretasikan secara benar.

b. Generalisasi

Subtype merupakan penambahan penting terhadap ERD (*Entity Relationship Diagram*) di orientasi objek. Fasilitas ini memiliki korespondensi dengan pewarisan di pemrograman berorientasi objek.

c. *Dependency*

Terdapat delapan *stereotype* terhadap *dependency* diantara kelas dan objek, yaitu :

- 1) *Bind*, menspesifikasikan sumber menginstansi template target menggunakan parameter actual.
 - 2) *Derive*, menspesifikasikan sumber dapat dikomputasi dari target.
 - 3) *Firend*, menspesifikasikan sumber diberi ketampakan spesial dari target.
 - 4) *instanceOf*, menspesifikasikan objek sumber adalah instan dari keals.
 - 5) *Instantiate*, menspesifikasikan sumber menciptakan instan dari target.
 - 6) *Powertype*, menspesifikasikan target adalah powertype dari sumber, powertype adalah classifier yang memiliki objek – objek adalah anak –anak dari induk tertentu.
 - 7) *Refine*, menspesifikasikan sumber adalah derajat abstraksi yang lebih baik dibanding target.
 - 8) *Use*, menspesifikasikan semantiks dari elemen sumber bergantung pada semantiks bagian publik dari target.
- (B. Hariyanto, 2004).

2.5.6 *Black Box Testing*

Menurut Khan tentang *Black Box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black Box Testing* cenderung untuk menemukan hal – hal seperti fungsi yang tidak benar atau tidak ada, kesalahan antarmuka, kesalahan pada struktur data, dan akses basis data, kesalahan performansi, dan kesalahan inisialisasi dan terminasi. (khan, 2011).

2.2.7 *Fruity Loops*

Menurut slameto, Fuity Loops merupakan sebuah *software* musik digital dimana *software* ini bisa digunakan sebagai bahan acuan dalam pembelajaran tata suara baik yang mengutamakan bagaimana cara menulis notasi dengan sesuai kompentensinya masing –masing sehingga dapat menciptakan sebuah karakter pada komposisinya. Selain itu Fuity Loops

juga berfungsi untuk mengembangkan suara dalam format *recording*, *mixing*, atau *dubbing*. (Slameto 2003).

2.2.8 Adobe Photoshop

Menurut Ansoryllah, Adobe Photoshop adalah perangkat lunak yang diproduksi oleh Adobe Systems serta berguna untuk mengolah/mengedit sebuah gambar ataupun foto. Perangkat lunak ini dapat dijalankan untuk sistem operasi Windows, MacOS X dan MacOS.

Pada awalnya Adobe Photoshop hanya diperuntukan khusus Macintosh yang merupakan awal atau versi pertama dari Adobe Photoshop yang dirilis pada tahun 1990. Namun, sebelum versi pertama dirilis Photoshop pernah berganti – ganti nama dari Display, ImagePro hingga kini yang dikenal sebagai Adobe Photoshop. (Ansoryllah, 2018).

2.2.9 Unity

Menurut Ardianto, *Unity* adalah *game engine cross-platform* yang dikembangkan oleh *Unity Technologies* (John Riccitiello, 2014). *Unity* dapat digunakan untuk mengembangkan video *game* dua dimensi maupun tiga dimensi. *Unity* memiliki banyak fitur yang mudah digunakan, sehingga bahkan pemula dapat memakainya. Salah satu keunggulan *unity* adalah *Unity* merupakan *game engine cross-platform*. *Unity* dapat membuat game untuk berbagai platform seperti Windows, Linux, MacOS, Playstation, Xbox, dan bahkan mobile platform seperti Android dan Ios.

Dalam pembuatan *game* dengan *Unity*, ada dua Bahasa pemrograman yang digunakan oleh *Unity*, Bahasa pemrograman C# atau *JavaScript*. User dapat memilih antara dua Bahasa pemrograman tersebut. Untuk mempermudah *Developer* menambah fitur dalam game, *Unity* juga menyediakan berbagai fitur tambahan, seperti *Unity Ads*, *Unity Analytics*, *Unity Multiplayer*, *Unity Collaborite*. (Ardianto, 2018).

2.2.10 Blender

Menurut Eka Ardhianto, blender adalah salah satu software open source yang digunakan untuk membuat konten multimedia khususnya 3Dimensi. Ada beberapa kelebihan yang dimiliki Blender yaitu Open Source dimana kita bebas memodifikasi source codenya, kemudian multi platform sehingga Blender tersedia untuk berbagai macam sistem operasi seperti Windows, Linux, dan MacOS. (Eka Ardhianto, 2012).