

BAB 2

TINJAUAN PUSTAKA

2.1 Profil Instansi

Menurut Teguh Kusuma USBP adalah :

United state of Bandung Percussion (USBP) adalah komunitas dari kota Bandung yang menaungi para pencinta musik perkusi, khususnya di kota bandung. USBP didirikan oleh 9 orang pendiri, dan diresmikan pada tanggal 22 maret 2012. Selain sebagai komunitas dan mengadakan latihan rutin kegiatan lain dari USBP juga sebagai pelatih siswa/mahasiswa di sekolah / kampus di bandung.

Kosep yang biasa di ajarkan dalam Perkusi Latin adalah conga menggunakan pola tumbao, bongo menggunakan Martilo dan timbales menggunakan Cascara (Teguh Kusuma,2018)

2.2.1 Logo Instansi

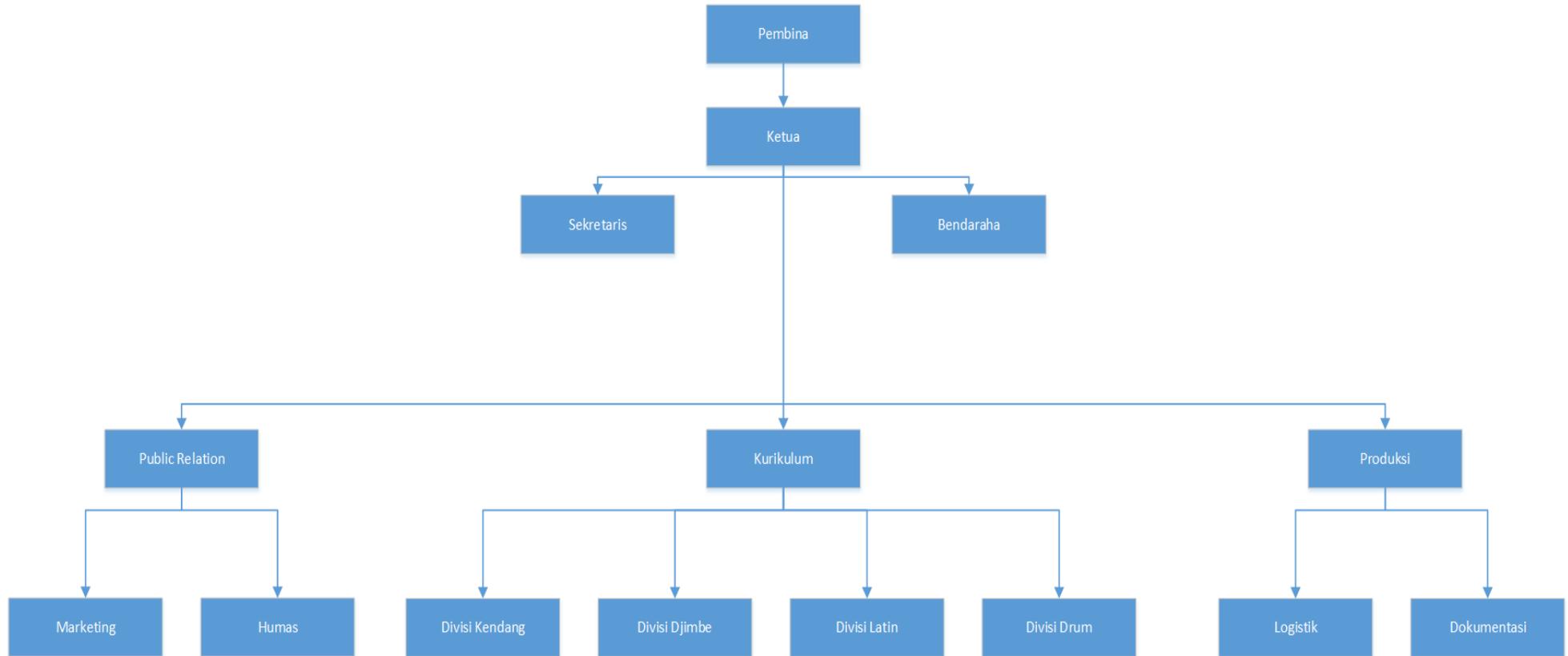
Logo dari United States of Bandung Percussion (USBP) ditunjukkan pada Gambar 2.1



Gambar 2.1 logo United States of Bandung Percussion (USBP)

2.2.2 Struktur Organisasi

Struktur organisasi dari United States of Bandung Percussion (USBP) ditunjukkan Gambar 2.2



Gambar 2.2 Struktur Organisasi United States of Bandung Percussion (USBP)

2.2 Landasan teori

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

2.2.1 Alat Musik Perkusi

Menurut Nurgiyanti tentang alat musik perkusi merupakan:

Alat musik perkusi sering juga disebut alat musik pukul atau tabuh adalah alat musik yang menghasilkan suara dengan cara di pukul, ditabuh, digoyang, digosok-gosokkan, atau dengan cara lain yang membuat objek bergetar maupun mngeluarkan suara lain baik dengan dipukul menggunakan suatu alat, dengan tongkat (stik), maupun dengan tangan kosong, di kocok, ataupun di adukan dengan benda lain yang dapat menghasilkan bunyi, instrumen perkusi biasa digunakan pada benda yang digunakan sebagai pengiring dalam suatu permainan musik yang dimainkan bersama sama atau sering disebut dengan permainan alat musik perkusi ansambel. Alat musik pukul juga disebut dengan alat musik perkusi. Perkusi kata ini berasal dari kata percussion (yang berarti memukul) dan percussus (kata benda yang berarti “pukulan”). Jenis musik yang termasuk kategori alat musik perkusi ini antara lain adalah : drum set, marimba, tamborin, dll. (Nurgiyanti, 2013).

2.2.1.1 Conga

Menurut Tri Atmaja tentang alat musik conga :

Conga adalah sebuah sebutan untuk drum tradisional dari Cuba yang berasal dari Afrika, mempunyai head tunggal berbentuk bulat tinggi dan langsing mengerucut ke bawah. Alat musik ini berasal dari daerah congolese Afrika. Seseorang yang memainkan conga disebut “conguero”. Pada mulanya conga berasal dari afrika dan terbuat dari kayu yang dilubangi, tetapi di Cuba sendiri conga terbuat dari beberapa papan yang di ikat menjadi satu seperti tong. (Tri Atmaja,2016).

Gambar alat musik conga seperti yang ditunjukkan Gambar 2.3.



Gambar 2.3 Alat Musik Conga

(Sumber : <https://www.topsimages.com>)

Menurut Evans bahwa suara conga :

1. *Open Tone*, Nada pertama dan terpenting yang harus dicapai adalah nada drum atau nada terbuka yang dihasilkan dengan memukul tepi drum dengan telapak tangan, memungkinkan jari-jari terpental dari kepala sehingga menghasilkan nada terbuka.
2. *The closed tone*, nada kedua, atau ketiadaan dihasilkan dengan memukul bagian tengah drum dengan tangan penuh atau telapak tangan. pada stroke ini telapak tangan atau tangan tetap di kepala menghasilkan suara gedebuk masih memiliki resonansi karena kedalaman dan bentuk drum conga besar.
3. *A sharp percussive stroke*, pukulan ditempatkan di kepala di tepi tempat *open tone* dilakukan. suara dapat dihasilkan hampir di mana saja di kepala tetapi lebih mudah untuk mendapatkan suara lebih dekat ke tepi, perbedaan antara suara *open tone* dan suara *slap* adalah bahwa dalam nada terbuka jari-jari terpental dari kepala saat dalam suara tamparan mereka menggeret kepala. jika telapak tangan menyentuh sebagian kecil kepala, jari-jari itu memungkinkan jari-

jari yang sekejap menampar keras kepala. tangan memulai pukulan dengan tangan lurus dan jepitan jari di kepala setelah telapak tangan menyentuh kepala menghasilkan *snap*. (Evans, 1996)

2.2.1.2 Bongo

Menurut Salloum tentang alat musik bongo :

Bongos, seperti yang kita kenal sekarang, pertama kali digunakan dalam musik Kuba yang disebut anak (diucapkan *sone*). *Sone* berevolusi sekitar tahun 1900 dari provinsi Oriente di Kuba timur. Gaya pedesaan awal termasuk vokal, gitar (*tres*), bass (*botija* atau *marimbula*), dan perkusi. Kemudian putra itu bermigrasi ke pusat-pusat perkotaan dan terompet atau *cornet* ditambahkan. Pada tahun 1920-an putranya telah menjadi kegemaran tarian paling populer di Havana dan segera menyebar ke Amerika Serikat.

Ketika musik berevolusi begitu juga drum itu sendiri. Awalnya bongo kepala ditempelkan pada kulit kayu dan diperketat oleh panas matahari atau di atas api. Namun, kepala kulit alami menyajikan masalah ketegangan dengan perubahan kelembapan sehingga drummer terus-menerus perlu menyetel drum. Selama batang tombak logam tahun 1930-an digunakan untuk menghindari ketidaknyamanan memanaskan kepala. Perkembangan bongos asli hingga hari ini telah dilestarikan dengan koleksi bongo yang luas di National Museum of Music di Havana, Kuba. (Salloum, 1997).

Gambar alat musik bongo seperti yang ditunjukkan Gambar 2.4.



Gambar 2. 4 Alat Musik Bongo

(Sumber : <https://www.topsimages.com>)

Menurut Salloum bahwa suara bongo :

1. *The open tone*, Biasanya dimainkan dengan jari telunjuk, tengah atau pertama tiga digit secara bersamaan. Jari memukul tepi drum dan melambung dari kepala.
2. *Muff*, Dimainkan dengan memukul kepala dengan tiga jari pertama dan menjebak udara di antara jari-jari dan kepala tanpa memantul.
3. *Slap – Closed*, Dimainkan dengan memukul tepi drum dengan tangan yang ditangkupkan tanpa rebound (seperti menyambar kepala drum).
4. *Muted stroke*, dimainkan dengan menekan sebentar ibu jari kiri dan eminence ke kepala saat menyerang dengan jari telunjuk kanan. (Salloum, 1997).

2.2.1.3 Timbales

Menurut Banoe tentang alat musik timbales :

Timbales adalah Perangkat alat musik perkusi latin, berupa drum satu head yang berbadan logam, dipergunakan secara berpasangan dan disangga tiang berkaki, dilengkapi dengan pasangan cowbell dan woodblock. (Banoe, 2003).

Menurut Silverman & Quintana tentang alat musik timbales :

Cuban percussionists mengembangkan gaya bermain yang berbeda pertama tympani maka pailitas cubanas atau timbales. Orang Kuba menggunakan tangan dan jari mereka untuk menghasilkan suara, teknik yang disebut "manoseo del

cuero." Juga, sebagai pelopor cascara, drummer Kuba akan bermain di cangkang atau cascara tympani. Mereka juga lebih sering bermain di tepi atau "perbatasan" tympani. Tangan kiri akan menyimpan waktu di hembra, semakin besar dari dua tympanis. Suara ini digunakan selama bagian tertentu dari lagu, pelopor untuk timbales modern, ketika, misalnya, cascara biasanya digunakan untuk ayat-ayat. Bunyi baqueteo juga pertama kali dikembangkan pada tympani di mana ketukan simultan diproduksi oleh tongkat di kepala atau cascara sementara jari-jari menghasilkan suara lain, mengisi di mana satu tangan tidak bermain. (Silverman & Quintana, 1998).

Gambar alat musik timbales seperti yang ditunjukkan Gambar 2.5.



Gambar 2. 5 Alat Musik Timbales

(Sumber : <https://www.topsimages.com>)

Menurut Silverman & Quintana suara timbales :

1. The sound of the shell, Atau cascara, adalah bagian yang sangat penting dari keseluruhan suara timbales. Suara kering dan renyah dengan serangan yang cerah. Logam yang digunakan untuk membuat timbales banyak berhubungan dengan suara. Baja dan kuningan adalah logam yang paling umum digunakan untuk membuat timbale dan masing-masing memberikan rasa yang khas ke cascara.

2. The macho, atau lebih kecil dari dua drum, memiliki dua suara utama yang tersedia untuk Anda. Salah satu suara yang paling penting dari macho adalah rim shots. Suara ini adalah salah satu nada timbale yang paling dikenal. Hal ini dicapai dengan memukul, dengan kekuatan yang cukup, kepala dan pelek secara bersamaan menimbulkan nada yang tajam, cerah dan agak resonansi.
3. The hembra, atau drum rendah, juga memiliki beberapa suara yang tersedia untuk Anda. Sangat jarang sekali memainkan rim, seperti di macho, di hembra. Lebih umum, nada terbuka drum digunakan. Suara ini dapat dicapai dengan bermain dengan tongkat. (Silverman & Quintana,1998).

2.2.2 Black Box Testing

Menurut Khan tentang *Back Box Testing* :

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).

2.2.3 Scalling Methods

Menurut Dunn rankin tentang skala likert :

Skala *Likert* adalah skala yang digunakan untuk mengukur persepsi, sikap atau pendapat seseorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti. Dengan Skala Likert, variabel yang akan diukur dijabarkan menjadi indikator variabel. Tabel skor skala *likert* seperti yang ditunjukkan Tabel 2.1.

Tabel 2.1 Skor Skala Likert

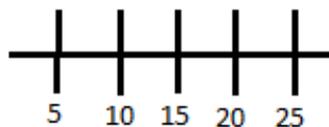
Pernyataan	Skor
Sangat Setuju (SS)	5
Setuju (S)	4
Cukup Setuju (CS)	3
Tidak Setuju (TS)	2
Sangat Tidak Setuju (STS)	1

Untuk menghitung total skor dari setiap responden, dilakukan dengan menjumlahkan setiap skor yang diperoleh responden. Langkah-langkah nya sebagai berikut :

1. Menentukan total skor maksimal.
(skor terbesar x jumlah responden)
2. Menentukan total skor minimal.
(skor terkecil x jumlah responden)
3. Menentukan nilai median.
(total skor maksimal + total skor minimal) / 2
4. Menentukan nilai kuartil 1.
(total skor minimal + nilai median) / 2
5. Menentukan nilai kuartil 3.
(total skor maksimal + nilai median) / 2

Setelah melakukan perhitungan, akan dibuat sebuah skala skor. (Dunn-Rankin, 2013).

Gambar skala skor seperti yang ditunjukkan Gambar 2.6

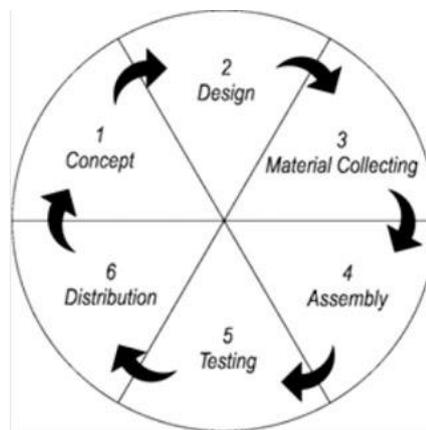
**Gambar 2.6 Skala Skor**

2.2.4 Multimedia Development Life Cycle (MDLC)

Menurut Nurrajizah tentang MDLC :

Multimedia Development Life Cycle (MDLC) merupakan tahapan pengembangan multimedia, dalam penelitian ini menggunakan MDLC Luther yang kemudian dikembangkan lagi oleh Sutopo. Pengembangan multimedia ini memiliki enam tahapan, yaitu : *concept, design, material collecting, assembly, testing, dan distribution*. (Nurrajizah, 2013).

Gambar MDLC seperti yang ditunjukkan Gambar 2.7.



Gambar 2.7 Multimedia Development Life Cycle

(Sumber : Nurrajizah, 2013)

Menurut Nurrajizah tahapan-tahapan dalam MDLC yang tersusun secara sistematis adalah sebagai berikut:

1. Konsep (*Concept*)

Tahap konsep merupakan tahap awal dalam siklus MDLC. Pada tahap konsep, dimulai dengan menentukan tujuan pembuatan aplikasi serta menentukan pengguna aplikasi tersebut.

2. Perancangan (*Design*)

Konsep yang sudah matang akan memudahkan dalam menggambarkan apa yang harus dilakukan. Tujuan dari tahap perancangan adalah membuat spesifikasi secara terperinci mengenai arsitektur proyek, tampilan dan kebutuhan material proyek, serta gaya. Tahap ini menggunakan storyboard untuk menggambarkan rangkaian cerita atau deskripsi tiap scene sehingga

dapat dimengerti oleh pengguna, dengan mencantumkan semua objek multimedia dan tautan ke scene lain.

3. Pengumpulan Bahan(*Material Collecting*)

Material Collecting adalah tahap pengumpulan bahan yang sesuai dengan kebutuhan. Bahan-bahan tersebut antara lain gambar, foto, animasi, video, audio, serta teks baik yang sudah jadi ataupun yang masih perlu dimodifikasi sesuai dengan kebutuhan yang ada. Bahan-bahan tersebut dapat diperoleh secara gratis atau dengan pemesanan kepada pihak lain sesuai dengan rancangan yang telah dibuat pada tahap sebelumnya.

4. Pembuatan(*Assembly*)

Tahap assembly adalah tahap pembuatan keseluruhan bahan multimedia. Aplikasi yang akan dibuat didasarkan pada tahap design, seperti storyboard. Tahap ini biasanya menggunakan perangkat lunak authoring, seperti Macromedia Director ataupun Adobe Flash.

5. Pengujian(*Testing*)

Pengujian dilakukan untuk memastikan bahwa hasil pembuatan aplikasi multimedia sesuai dengan rencana. Ada dua jenis pengujian yang digunakan, yaitu pengujian alpha dan pengujian beta. Pengujian alpha seperti menampilkan tiap halaman, fungsi tombol serta suara yang dihasilkan. Jika ada malfunction maka aplikasi akan segera diperbaiki. Jika telah lolos dalam pengujian alpha maka akan dilanjutkan dengan pengujian beta. Pengujian beta adalah pengujian yang dilakukan oleh pengguna, dengan membuat kuisisioner tentang aplikasi yang dibuat.

6. Distribusi(*Distribution*)

Tahap ini adalah tahap terakhir dalam siklus pengembangan multimedia. Pendistribusian dapat dilakukan setelah aplikasi dinyatakan layak pakai. Pada tahap ini, aplikasi akan disimpan dalam suatu media penyimpanan seperti CD, perangkat mobile atau situs web. Jika media penyimpanan tidak cukup untuk menampung aplikasinya, kompresi terhadap aplikasi tersebut akan dilakukan. Tahap evaluasi termasuk ke dalam tahap ini. Adanya evaluasi sangat

dibutuhkan untuk pengembangan produk yang sudah dibuat sebelumnya agar menjadi lebih baik. (Nurajizah, 2013).

2.2.5 UML (*Unified Modelling Language*)

Menurut R. A. S. dan M. Shalahuddin UML adalah :

UML (*Unified Modeling Language*) adalah salah standar mendefinisikan *requirement*, membuat analisis & desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek. (R. A. S. And M. Shalahuddin, 2018) .

Menurut B. Hariyanto UML adalah :

UML adalah bahasa grafis untuk mendokumentasi, menspesifikasikan, dan membangun sistem perangkat lunak. UML berorientasi objek, menerapkan banyak level abstraksi, tidak bergantung proses pengembangan, tidak bergantung bahasa dan teknologi, pemanduan beberapa notasi di beragam metodologi, usaha bersama dari banyak pihak, didukung oleh kakas-kakas yang diintegrasikan lewat XML (XMI). Standar 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 pertukran 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 kakas berorientasi objek
6. Mendukung konsep-konsep pengembangan level lebih tinggi seperti komponen, kolaborasi, framework, dan pattern

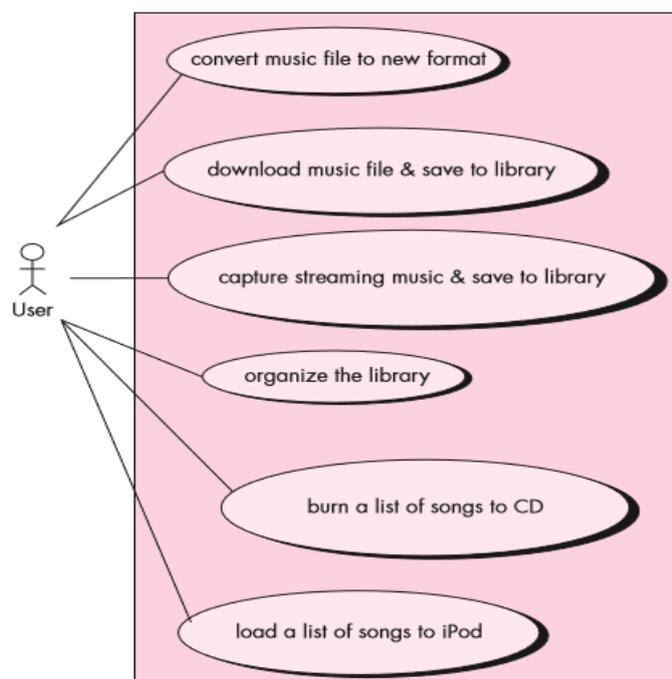
Secara fundamental, UML berkaitan dengan penangkapan dan komunikasi pengetahuan.

Konsep-konsep yang diterapkan di UML adalah satu model berisi informasi mengenai sistem (atau domain), 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* adalah :

Diagram *Usecase (usecase diagram)* merupakan salah satu diagram untuk memodelkan aspek perilaku sistem masing-masing digram *usecase* menunjukkan sekumpulan *usecase*, aktor, dan hubungannya. Diagram *usecase* adalah 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 2.8.



Gambar 2.8 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 antara 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 aktual 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 adalah 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 mempertukarkan informasi dengan sistem. Dalam perspektif aktor, *usecase* melakukan sesuatu yang berharga bagi aktor.

Aktor adalah tipe (kelas), bukan instan. Aktor merepresntasikan peran bukan pemakai individu dari sistem. Aktor mempunyai nama. Nama yang dipilih harus menyatakan peran aktor.

Aktor berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan. *Usecase* selalu diawali oleh aktor yang mengirim pesan, disebut dengan 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* adalah :

- 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 antar *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

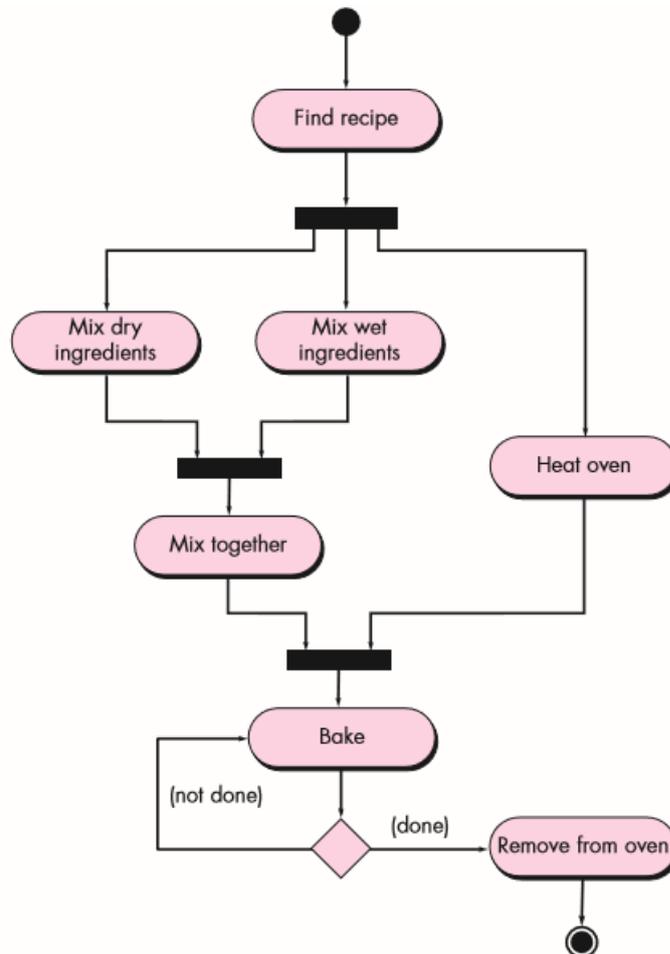
(B. Hariyanto, 2004).

2.2.5.2 *Activity Diagram*

Menurut B. Hariyanto *Activity Diagram*

Diagram aktivitas 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 nonatomik 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 2.9.



Gambar 2. 9 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. Logik dari proses bisnis

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

Diagram aktivitas berfokus pada aktivitas-aktivitas, potongan-potongan dari proses yang boleh 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 dari flowchart terutama karena diagram aktivitas secara eksplisit mendukung aktivitas-aktivitas paralel dan sinkronasi aktivitas-aktivitas itu.

Penggunaan utama model aktivitas adalah :

1. Memodelkan *workflow*

Pembuatan *workflow* adalah sebagai berikut :

- a. Tetapkan fokus dari *workflow*. Untuk sistem yang tidak sepele, tidak mungkin menunjukkan semua *workflow* yang penting di satu diagram.
- b. Pilih objek bisnis yang mempunyai tanggungjawab level tinggi untuk bagian dari *workflow*. Ini dapat berupa sesuatu yang nyata di sistem, atau lebih abstrak. Dalam kasus tertentu dapat digunakan *swimlane* untuk masing-masing bisnis penting.
- c. Identifikasikan precondition dari state awal workflow dan postcondition dari state akhir workflow. Ini penting untuk memodelkan batas dari workflow.
- d. Mulai dari state awal workflow, spesifikasikan aktivitas dan aksi yang ada dan tempatkan sebagai aktivitas di diagram.
- e. Untuk aksi yang rumit, atau sekelompok aksi yang muncul berulang kali, jadikan menjadi satu state aktivitas dan kemudian buat diagram aktivitas tersendiri untuk menggambarkannya.
- f. Buat trasisi yang menghubungkan antar aktivitas.
- g. Jika terdapat objek penting yang terlibat di workflow, maka gambarkan di diagram aktivitas.

2. Memodelkan operasi

Diagram aktivitas untuk memodelkan operasi adalah sebagai berikut :

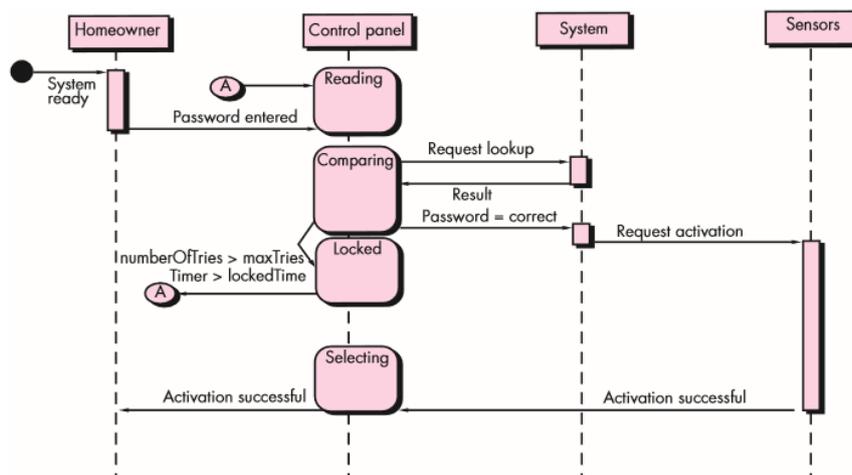
- a. Kumpulkan abstraksi yang terlibat di operasi. Ini meliputi parameter operasi, atribut di kelas dan kelas-kelas tetangga yang terlibat.
- b. Identifikasi preconditions dari state awal operasi dan postcondition dari state akhir dari operasi. Juga identifikasi invarian di kelas yang harus selalu dipenuhi selama eksekusi operasi.

- c. Dimulai dari state awal operasi, spesifikasikan aktivitas-aktivitas dan aksi-aksi yang terjadi dan buat di diagram aktivitas sebagai state aktivitas atau state aksi.
- d. Gunakan pencabangan bila diperlukan untuk menspesifikasikan jalur kondisi dan iterasi.
- e. Hanya jika operasi itu dimiliki oleh kelas aktif, gunakan forking dan joining bila perlu untuk menspesifikasikan alur paralel dari kendali.

2.2.5.3 Sequence Diagram

Menurut B. Hariyanto *Sequence Diagram* adalah :

Diagram sekuen mendeskripsikan komunikasi di antara objek-objek, meliputi pesan-pesan yang ada dan urutan pesan tersebut muncul. Contoh *Sequence diagram* dapat dilihat pada Gambar 2.10.



Gambar 2.10 Contoh *Sequence Diagram*

(sumber : R. S. Pressman, 2010)

Diagram sekuen digunakan untuk :

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

Diagram sekuen digunakan untuk memodelkan skenario 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.

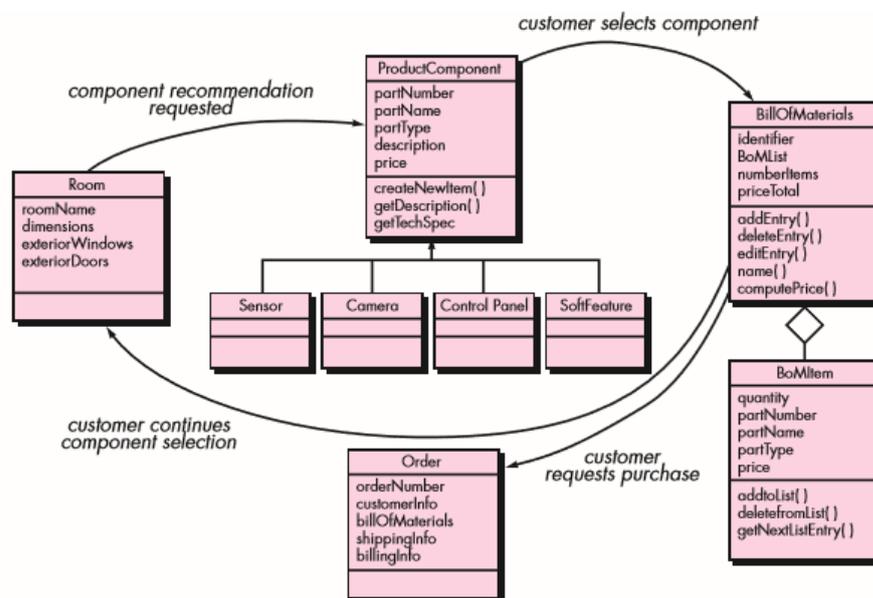
Diagram sekuen menunjukkan objek sebagai garis vertikal dan tiap kejadian sebagai panah horisontal dari objek pengirim ke objek penerima. Waktu berlalu dari atas ke bawah dengan lama waktu tidak relevan. Diagram ini hanya menunjukkan barisan kejadian, bukan pewaktuan nyata. Kecuali untuk sistem waktu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

2.2.5.4 Class Diagram

Menurut B. Hariyanto *Class Diagram* adalah :

Diagram kelas 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 hubungan antar kelas-kelas itu, atribut-atribut dan operasi-operasi di kelas-kelas. (B. Hariyanto, 2004).

Contoh *Sequence diagram* dapat dilihat pada Gambar 2.11.



Gambar 2.11 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 di diagram kelas 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 di sistem berorientasi objek. Kelas mendeskripsikan satu blok pembangun sistem. Kelas memiliki sejumlah fitur, kita dapat memodelkan multiplisitas, ketampakan, penanda, polymorphism, dan karakteristik-karakteristik lain.

2. Antarmuka (*interfaces*)

Antarmuka (*interfaces*) 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 merepresentasikan implementasi pola tertentu yang membentuk sistem.

4. Hubungan (*relationship*)

Hubungan antar kelas di diagram kelas beraneka ragam, yaitu :

a. Asosiasi

Asosiasi merepresentasikan hubungan antara instan-instan kelas. Interpretasi dari asosiasi bervariasi sesuai perspektifnya. Secara konseptual, asosiasi merepresentasi hubungan antara kelas-kelas yang terlibat. Pada perspektif spesifikasi, terdapat tanggungjawab untuk mengetahui, dan akan dibuat eksplisit dengan operasi pengaksesan dan pembaruan. Ini berarti terdapat pointer antara satu kelas dan kelas lainnya, tapi ini tersembunyi oleh pengkapsulan. Perspektif implementasi mengimplikasikan keberadaan pointer. Dengan demikian hal esensi untuk mengetahui perspektif 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 koerspondensi dengan pewarisan di pemrograman berorientasi objek.

c. *Dependency*

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

- 1) *Bind*, menspesifikasikan sumber menginstansiasi template target menggunakan parameter aktual.
- 2) *Derive*, menspesifikasikan sumber dapat dikomputasi dari target.
- 3) *Friend*, menspesifikasikan sumber diberi ketampakan spesial oleh target.
- 4) *instanceOf*, menspesifikasikan objek sumber adalah instan dari kelas.
- 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.

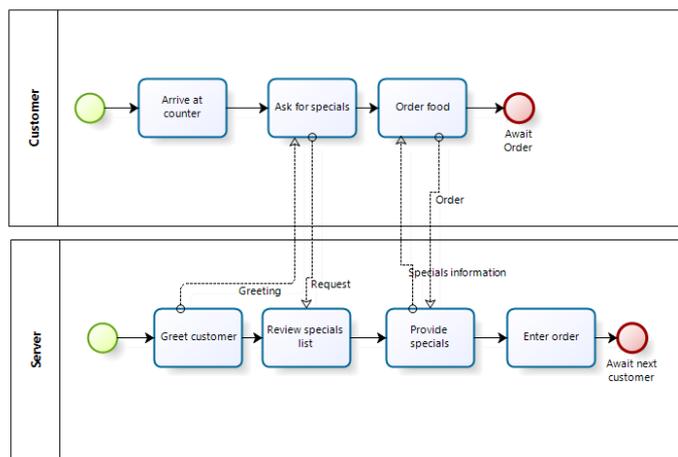
(B. Hariyanto, 2004).

2.2.6 BPMN

Menurut Rahmawati, BPMN adalah:

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 2.12.



Gambar 2.12 Contoh BPMN

(Sumber : Rahmawati, 2017)

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

1. *Flow Objects* 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.

2.2.7 Multimedia

Menurut Vaughan, Multimedia adalah:

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

1. Teks

Setiap kata dapat memiliki beberapa makna, oleh sebab itu saat akan menggunakan teks, penting untuk memperhatikan akurasi dan keringkasan pada kata yang akan dipilih. Kata dan simbol dalam bentuk apapun, baik tertulis maupun lisan adalah bentuk 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 juga disebut raster, digunakan untuk photo 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, koordinat 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)

2.2.8 Android

Menurut Abdul Azij, Android adalah:

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 seluler. Berikut versi yang telah dikeluarkan oleh android:

3.1.2.1 Alpha

Dalam hal ini (Android 1.0) adalah versi pertama dari Sistem Operasi Android oleh Google. Ini memiliki fungsi dasar dengan browser sederhana dan aplikasi Google lainnya seperti Gmail, Maps, dan YouTube.

3.1.2.2 Beta

Kemudian dengan Android 1.1 beberapa fungsi tambahan ditambahkan, API berubah dari Level 1 di Android 1.0 ke 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.

3.1.2.3 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 perubahan UI.

3.1.2.4 Donut

Versi Android 1,6 dijuluki sebagai DONUT. Dalam android versi ini Google menambahkan dukungan teknologi baru CDMA/EVDO, dukungan layar WVGA dan mesin sintesis pengucapan yang memungkinkan aplikasi android tertentu mampu mengucapkan suatu teks.

3.1.2.5 Eclair

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

3.1.2.6 Froyo

Android versi 2.2-2.2.3 diperkenalkan dengan fungsi dan aplikasi USB tethering & WiFi hotspot, dukungan Adobe flash, peningkatan kecepatan dan kinerja aplikasi dengan fitur-fitur baru.

3.1.2.7 Gingerbread

Gingerbread (Android 2.3-2.3.7) diperkenalkan dengan Antarmuka Pengguna yang diperbarui yang memberikan lebih banyak kemudahan untuk digunakan. Fiturnya seperti sensor, beberapa kamera (Depan & belakang), keyboard virtual, saran teks yang lebih baik, kemampuan input suara dan tekan terus kemampuan copy paste.

3.1.2.8 Honeycomb

Platform 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 mode penyamaran tambahan dan obrolan video & Gtalk didukung.

3.1.2.9 Ice Cream Sandwich

Ice Cream sandwich hadir di tahun 2011 membawa semua tampilan baru. Versi android yang satu ini memberikan kemudahan lebih kepada pengguna seperti, pengguna dapat dengan cepat menggesek untuk menutup aplikasi, tata letak galeri baru dan editor foto built in.

3.1.2.10 Jelly Bean

Google membuat Sistem Operasi lebih responsif dengan Jelly Bean dan memperkenalkan berbagi file dengan Android Beam. Keunggulan utama dari versi ini adalah antarmuka pengguna yang lebih halus.

3.1.2.1 Kitkat

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

3.1.2.2 Lollipop

Lollipop – Android versi 5.0-5.1.1 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 direset ulang tanpa memasukkan id google dan passwordnya.

3.1.2.3 Marshmallow

Marshmallow dirilis pada tahun 2015, muncul dengan baterai yang lebih cerdas dan mode tidur (itu mencegah tugas tertentu dari berjalan jika ponsel sedang mengatur siaga). Now On Tap, pengaturan privasi yang lebih baik, lebih mudah untuk meng-upgrade ponsel, sensor sidik jari dan dibangun di pesan suara visual.

3.1.2.4 Nougat

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

3.1.2.5 Oreo

Android Oreo pertama kali dirilis sebagai preview pengembang pada tanggal 21 Maret 2017. Android Oreo mengusung fitur baru yang membuat smartphone lebih cepat, pintar dan powerful dibandingkan dengan versi android yang sebelumnya. Beberapa fitur unggulan dari versi ini adalah background limit, Google Play Protect, emoji baru dan picture in picture. (Abdul Azij , 2018)

2.2.9 Multiplayer

Menurut Lukman yusuf bahwa :

Multiplayer adalah sebutan dalam game yang dapat dimainkan secara bersama-sama melalui jaringan internet. Dalam perangkat smartphone, konektivitas yang dapat digunakan untuk menghubungkan perangkat smartphone ke internet dapat melalui mobile internet yaitu 3G dan 4G dan WLAN. (Lukman yusuf, 2017)

2.2.10 Fruity Loops

Menurut Fahmi, Fruity Loops adalah:

Fruity loops atau biasa disebut FL studio adalah “program untuk memproduksi audio dan musik yang mendukung keperluan rekaman audio multi-track, sequencing dan mixing.” Audio yang dibuat dengan program ini dapat diekspor ke format .WAV, .MP3, .OGG, dan juga .MID. (Fahmi,2014).

Gambar Logo Fruity Loops seperti yang ditunjukkan Gambar 2.13.



Gambar 2. 13 Fruity Loops

(Sumber : <https://www.image-line.com/>)

2.2.11 Unity

Menurut Ardianto, Unity adalah :

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 bahwa 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 Collaborate*. (Ardianto,2018)

Gambar Logo Unity seperti yang ditunjukkan Gambar 2.14.



Gambar 2. 14 Logo Unity

(Sumber : <https://unity3d.com/>)

2.2.12 Adobe Photoshop

Menurut Ansoryllah, Adobe Photoshop adalah:

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

Pada awalnya Adobe Photoshop hanya diperuntukkan 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)

Gambar Logo Adobe Photoshop seperti yang ditunjukkan Gambar 2.15.



Gambar 2. 15 Adobe Photoshop

(Sumber : <https://www.adobe.com/>)

2.2.13 Software Development Kit (SDK)

Software Development Kit (SDK) adalah suatu *kit* atau *library* dari bahasa pemrograman untuk pengembangan atau pembangunan suatu perangkat lunak dan biasanya SDK terdiri dari kumpulan *tools* yang dibutuhkan. Misalnya bahasa pemrograman java, mempunyai SDK yang berisi suatu *library* yang dapat digunakan untuk membuat suatu aplikasi berbasis java.

2.2.14 Goggle Play Games Service

Menurut Erdiawan, Google Play Games Service adalah:

Google Play Games adalah fitur sosial untuk game mobile yang dikembangkan Google untuk pengguna Android. Pengguna bisa melihat achievement atau pencapaian dari game yang mereka mainkan, bermain *multiplayer* secara *realtime*, progress game akan tersimpan secara *cloud* dan melihat *score* tertinggi yang pengguna dapat dari game. (Erdiawan, 2016)

Gambar Logo Google play games seperti yang ditunjukkan Gambar 2.16.



Gambar 2. 16 Logo Google Play Games

(Sumber : <https://play.google.com/>)

