

BAB 2

TINJAUAN PUSTAKA

2.1 Kebakaran

Kebakaran adalah suatu peristiwa berupa bencana, dimana timbul api yang tidak terkendali yang dapat membahayakan keselamatan jiwa maupun harta benda [6]. Kebakaran dapat diklasifikasikan berdasarkan beberapa kriteria umum yang dapat digunakan untuk menanggulangi kebakaran tersebut dengan pendekatan yang spesifik.

1.1.1 Kebakaran Struktural

Kebakaran struktural atau kebakaran bangunan adalah kebakaran yang terjadi pada objek struktural berupa bangunan, baik kebakaran yang terjadi didalam suatu struktur, atau kebakaran struktur itu sendiri [7].

1.1.2 Kebakaran Liar

Kebakaran liar adalah segala bentuk kebakaran yang terjadi pada daerah pertumbuhan tanaman-tanaman liar secara natural. Secara spesifiknya, kebakaran liar dapat dibagi menjadi beberapa macam:

1. Kebakaran Hutan
2. Kebakaran Lahan
3. Kebakaran Semak

1.2 Klasifikasi Kebakaran

Menurut Peraturan Menteri Tenaga Kerja Dan Transmigrasi No : PER.04/MEN/1980, kebakaran dapat diklasifikasikan menjadi empat golongan:

- a. Golongan A: Kebakaran bahan padat kecuali logam;
- b. Golongan B: Kebakaran bahan cair atau gas yang mudah terbakar;
- c. Golongan C: Kebakaran instalasi listrik bertegangan;
- d. Golongan D: Kebakaran logam.

1.3 Penanggulangan Kebakaran

Menurut SOP Pencegahan dan Penanggulangan Kebakaran serta Penyelamatan Diri, Penanggulangan Kebakaran adalah segala bentuk usaha untuk mencegah dan mengatasi peristiwa kebakaran yang dapat mengancam dan mengganggu kehidupan dan penghidupan masyarakat yang disebabkan, baik oleh kelalaian manusia maupun faktor lain, sehingga mengakibatkan timbulnya korban jiwa manusia, kerusakan lingkungan, kerugian harta benda serta dampak psikologis.

1.3.1 Upaya Pencegahan Kebakaran

Pencegahan kebakaran dapat dilakukan dengan upaya-upaya berikut:

1. Mengamankan instalasi listrik.
2. Penggunaan pemutus arus listrik dalam keadaan baik.
3. Membenahi kabel yang terbuka untuk menghindari hubungan pendek arus listrik
4. Menghindari pencantolan listrik.

1.3.2 Upaya Penanggulangan Kebakaran

Penanggulangan kebakaran dapat dilakukan dengan upaya-upaya berikut:

1. Menyediakan alat pemadam kebakaran seperti Alat Pemadam Api Ringan (APAR) atau selimut pemadam.
2. Menyediakan alternatif dari alat pemadam kebakaran, seperti karung goni basah sebagai pengganti selimut pemadam.
3. Menghubungi Pemadam Kebakaran.

1.4 Alat Pemadam Api Ringan

Menurut Peraturan Menteri Tenaga Kerja Dan Transmigrasi No : PER.04/MEN/1980, Alat Pemadam Api Ringan atau APAR adalah: “alat yang ringan serta mudah dilayani oleh satu orang untuk memadamkan api pada mula terjadi kebakaran”. APAR dapat dibagi menjadi empat jenis:

- a. Jenis Cairan (Air).
- b. Jenis Busa.

- c. Jenis Tepung Kering.
- d. Jenis Gas (*hydrocarbon* berhalogen dan sebagainya)

1.5 **Game**

Dalam bahasa Inggris, *game* memiliki arti dasar permainan. Secara definitif, permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan. Pemain berinteraksi dengan sistem dalam batas yang sudah ditentukan dengan istilah aturan permainan. Tantangan dan kesenangan dari bermain game didapat dari usaha pemain menyelesaikan permasalahan melalui interaksi dan sikap yang dibatasi [8].

1.5.1 **Jenis-Jenis Game**

Game dapat diklasifikasikan dalam beberapa jenis melalui pada *platform* apa game tersebut dijalankan, yaitu:

1. **Arcade games**, yaitu *game* yang dimainkan menggunakan mesin khusus yang disebut dengan mesin *arcade* atau *arcade cabinet*. Mesin *arcade* atau *arcade cabinet* biasanya hanya bisa dimainkan ketika pemain sudah membayar tarif permainan terlebih dahulu menggunakan uang koin, *token*, atau saldo elektrik melalui sebuah kartu khusus.
2. **PC Games**, yaitu *video game* yang dimainkan menggunakan *Personal Computer*.
3. **Console games**, yaitu *video game* yang dimainkan menggunakan perangkat keras khusus untuk bermain *game* yang disebut *console* seperti Playstation, XBOX, dan lain-lain.
4. **Handheld games**, yaitu *video game* yang dimainkan menggunakan *console portable*, yaitu *console* yang bisa dibawa kemana-mana.
5. **Mobile games**, yaitu *video game* yang dapat dimainkan pada *smartphone* atau tablet.

1.5.2 Genre Game

Melalui ragam perancangan dan konsepnya, *game* dapat dibagi atas beberapa genre, diantaranya yaitu:

1. **Action (Aksi)**

Genre ini memfokuskan pada keterlibatan sang pemain dalam aksi yang terjadi di lingkungan game dalam bentuk pertarungan, penjelajahan *level*, serta interaksi dengan dunia di dalam game.

2. **Fighting (pertarungan)**

Genre pertarungan terfokus pada pertarungan antara pemain dengan pemain lainnya atau representasi dari pemain lain dengan cara perkelahian antara dua karakter. *Genre* ini memiliki ciri khas banyaknya karakter yang dapat dipilih sebagai perwakilan pemain, dengan tiap karakter memiliki gerak-gerik atau teknik perkelahian yang unik antara satu dengan lain karakter.

3. **Adventure (Petualangan)**

Genre petualangan memiliki fokus lebih terhadap penjelajahan dunia di dalam game. Tidak jarang dunia di dalam game terbuka secara luas dengan akses yang tidak terstruktur (*open world*). Narasi atau cerita dalam *game* umumnya didapat seiring dengan penjelajahan dunia, dimana sang pemain yang diwakili oleh satu atau sekelompok karakter dapat berinteraksi dengan berbagai macam aspek dalam *game* seperti bertarung dengan musuh atau berbicara dengan karakter lain.

4. **Strategy (strategi)**

Identik dengan latar peperangan, *genre* ini umumnya menempatkan pemain dalam sudut pandang pemimpin sebuah pasukan dimana sang pemain diharuskan untuk mengumpulkan sumber daya, membangun markas dan pertahanan, melatih prajurit dan mengalahkan lawan yang memiliki peran mirip dengan sang pemain.

5. **Simulation (Simulasi)**

Game dengan *genre* simulasi fokus berupaya merepresentasikan kenyataan sebagai lingkungan *game*. Topik yang direpresentasikan pun

beragam dan secara teknis tidak memiliki batasan. Contoh *game* simulasi yang paling umum adalah simulasi penerbangan.

6. *Puzzle* (teka-teki)

Video game jenis ini sesuai namanya berintikan mengenai pemecahan teka-teki, baik itu menyusun balok, menyamakan warna bola, memecahkan perhitungan matematika, melewati labirin, sampai mendorong-dorong kota masuk ke tempat yang seharusnya, itu semua termasuk dalam jenis ini. Sering pula game jenis ini adalah juga unsur *game* dalam *video game* petualangan maupun *game* edukasi.

7. *Sport game* (Olahraga)

Game ini merupakan adaptasi dari kenyataan, membutuhkan kelincahan dan juga strategi dalam memainkannya. *Game* berupa kompetisi antara dua pemain atau lebih, di mana pemain dapat berupa individual atau tim. Contoh *game* tipe ini antara lain sepakbola, bola basket, tenis, dan bilyard.

8. *RPG* (*Role Playing Game*)

Video game jenis ini sesuai dengan terjemahannya, bermain peran, memiliki penekanan pada tokoh/peran perwakilan pemain di dalam *game*, yang biasanya adalah tokoh utamanya, dimana seiring kita memainkannya, karakter tersebut dapat berubah dan berkembang ke arah yang diinginkan pemain dalam berbagai parameter yang biasanya ditentukan dengan naiknya *level*, baik dari status kepintaran, kecepatan dan kekuatan karakter, senjata yang semakin sakti, ataupun jumlah teman maupun mahluk peliharaan.

9. *Education* (edukasi)

Game edukasi merupakan paket *software* yang menciptakan kemampuan pada lingkungan game yang diberikan sebagai alat bantu untuk memotivasi atau membantu siswa untuk melalui prosedur *game* secara teliti untuk mengembangkan kemampuannya. Developer yang membuatnya, harus memperhitungkan berbagai hal agar *game* ini benar-benar dapat mendidik, menambah pengetahuan dan meningkatkan keterampilan yang memainkannya. Target segmentasi pemain harus pula

disesuaikan dengan tingkat kesulitan dan desain visual ataupun animasinya.

2.6 *Gamification*

Gamification adalah bentuk pendekatan pembelajaran yang menerapkan elemen-elemen yang biasanya ditemukan dalam *game* atau *videogame* kepada aktivitas *non-gaming* [9]. Elemen-elemen yang diangkat bisa merupakan hal-hal seperti adanya *score* setiap tercapainya parameter pembelajaran yang ideal, memperkenalkan pemecahan masalah dalam bentuk *challenge*, atau hal-hal lebih abstrak yang fundamental seperti menstrukturkan kurikulum pembelajaran seperti layaknya sebuah *level* dalam *game*, dan seterusnya.

2.7 **HTML 5**

HTML (*HyperText Markup Language*) merupakan bahasa pengkodean yang menjadi pondasi dasar pembentuk aplikasi web. Tidak mungkin kita bisa membangun aplikasi web tanpa memahami HTML dengan baik. Di sisi lain, perkembangan HTML telah mencapai versi 5 (atau dikenal HTML5) yang merupakan standar web masa depan. Sangat rasional jika HTML5 dijadikan sebagai standar web karena menawarkan kemampuan-kemampuan powerful dan impresif. Tidak hanya itu saja, dukungan untuk pengembangan aplikasi web mobile juga menjadi bagian dari revolusi HTML menjadi lebih baik.

HTML 5 merupakan revisi baru yang merujuk pada spesifikasi HTML 4.01. proyek ini digagas oleh *Web HyperText Application Technology Working Group* (WHATWG) karena melihat HTML 4.01 sudah cukup lama tidak diperbaharui oleh konsorsium world wide web (*World Wide Web Consortium / W3C*). Saat ini W3C justru sedang fokus mengembangkan XHTML 2.0. Pada akhirnya W3C dan WHATWG berinisiatif untuk bekerja sama mengembangkan HTML5.

Berbeda sekali dengan versi-versi sebelumnya, HTML5 sudah menyediakan fitur *built-in* (tanpa tambahan plugin atau API) untuk mengimplementasikan animasi, *canvas*, *event-handler*, validasi form, *drag & drop*, audio & video, geolocation, dan komunikasi socket. Fitur-fitur canggih ini memungkinkan kita untuk menghasilkan aplikasi yang responsif dan menawan.

Tujuan umum dari HTML5 adalah untuk memperbaiki kemampuan HTML sehingga dapat mendukung multimedia sambil tetap memastikannya terbaca dengan baik oleh manusia dan dipahami oleh komputer serta peralatan-peralatan mobile. Tujuan ini tercermin jelas melalui fitur-fitur dan kemampuan-kemampuan yang disediakan. Salah satu contoh perbaikan kemampuan adalah dengan mengurangi kebergantungan plugin eksternal (seperti Flash) yang cenderung tidak hemat sumber daya.

Sebagai simpulan, HTML5 dibangun dengan merujuk pada spesifikasi HTML 4.01. Ini juga menegaskan bahwa HTML5 menjamin adanya kompatibilitas yang baik terhadap versi di bawahnya. Meskipun demikian, ada penambahan-penambahan yang jelas tidak didapatkan di versi sebelumnya, yaitu: *error handling*, API JavaScript, detail implementasi, dan rendering engine HTML5 [10].

2.8 Android

Android adalah sistem operasi untuk telepon seluler yang berbasis Linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri dan digunakan oleh berbagai macam perangkat *mobile*. Awalnya, Google Inc. membeli Android Inc., pendatang baru yang membuat peranti lunak untuk telepon seluler. Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan hardware, software, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Pada saat perilisan perdana Android, 5 November 2007, Android bersama Open Handset Alliance menyatakan mendukung pengembangan standar terbuka pada perangkat seluler. Di lain pihak, Google merilis kode-kode Android di bawah lisensi Apache. Di dunia ini terdapat dua jenis distributor sistem operasi Android. Pertama yang mendapat dukungan penuh dari Google atau Google Mail Services (GMS) dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung Google atau dikenal sebagai *Open Handset Distribution* (OHD) [11].

2.8.1 Versi Android

Berikut adalah berbagai versi Android sebagai berikut :

Tabel 2.1 Versi Android Yang Telah Rilis

Nama Versi	Versi	Tanggal Rilis
Android (Alpha)	1.0	23 September 2008
Beta	1.1	9 Februari 2009
Cupcake	1.5	27 April 2009
Donut	1.6	15 September 2009
Eclair	2.0 – 2.1	26 Oktober 2009
Froyo	2.2 – 2.2.3	20 Mei 2010
Gingerbread	2.3 – 2.3.7	6 Desember 2010
Honeycomb	3.0 – 3.2.6	22 Februari 2011
Ice Cream Sandwich	4.0 – 4.0.4	18 Oktober 2011
Jelly Bean	4.1 – 4.3.1	9 Juli 2012
Kitkat	4.4 – 4.4.4	31 Oktober 2013
Lollipop	5.0 – 5.1.1	12 November 2014
Marshmallow	6.0 – 6.0.1	5 Oktober 2015
Nougat	7.0 – 7.1.2	22 Agustus 2016

2.9 Multimedia

Multimedia adalah penggunaan komputer untuk menyajikan dan menggabungkan teks, suara, gambar, animasi dan video dengan alat bantu dan koneksi (*link*) sehingga pengguna dapat bernavigasi, berinteraksi, berkarya dan berkomunikasi. Multimedia sering digunakan dalam dunia hiburan. Selain dari dunia hiburan, Multimedia juga diadopsi oleh dunia *game*. Multimedia juga dapat diartikan sebagai penggunaan beberapa media yang berbeda dalam menyampaikan informasi berbentuk teks, grafik, *audio*, animasi, dan video.

2.9.1 Elemen Multimedia

Komponen multimedia terbagi atas lima jenis yaitu [12]:

1. Teks

Teks merupakan elemen multimedia yang menjadi dasar untuk menyampaikan informasi, karena teks adalah jenis data yang paling sederhana dan membutuhkan tempat penyimpanan yang paling kecil. Teks merupakan cara yang paling efektif dalam mengemukakan ide-ide kepada pengguna, sehingga penyampaian informasi akan lebih mudah dimengerti oleh masyarakat. Jenis-jenis teks seperti *Printed Text*, yaitu teks yang dihasilkan oleh *word processor* atau *word editor* dengan cara diketik yang nantinya dapat dicetak. *Scanned Text* yaitu teks yang dihasilkan melalui

proses *scanning* tanpa pengetikan dan *Hypertext* yaitu jenis teks yang memberikan link ke suatu tempat atau meloncat ke topik tertentu.

2. **Grafik**

Sangat bermanfaat untuk mengilustrasi informasi yang akan disampaikan terutama informasi yang tidak dapat dijelaskan dengan kata-kata. Jenis-jenis grafik seperti *bitmap* yaitu gambar yang disimpan dalam bentuk kumpulan pixel, yang berkaitan dengan titik-titik pada layar monitor. *Digitized picture* adalah gambar hasil rekaman video atau kamera yang dipindahkan ke komputer dan diubah ke dalam bentuk *bitmap*. *Hyperpictures*, sama seperti *hypertext* hanya saja dalam bentuk gambar.

3. **Audio**

Multimedia tidak akan lengkap jika tanpa audio (suara). Audio bisa berupa percakapan, musik atau efek suara. Format dasar audio terdiri dari beberapa jenis :

1. *Wave*

Merupakan format file digital audio yang disimpan dalam bentuk digital dengan eksistensi WAV.

2. *MIDI*

MIDI (*Musical Instrument Digital Interface*) memberikan cara yang lebih efisien dalam merekam musik dibandingkan *wave*, kapasitas data yang dihasilkan juga jauh lebih kecil. MIDI disimpan dalam bentuk MID.

4. **Video**

Video menyediakan sumber yang kaya dan hidup untuk aplikasi multimedia. Dengan video dapat menerangkan hal-hal yang sulit digambarkan lewat kata-kata atau gambar diam dan dapat menggambarkan emosi dan psikologi manusia secara lebih jelas.

5. **Animasi**

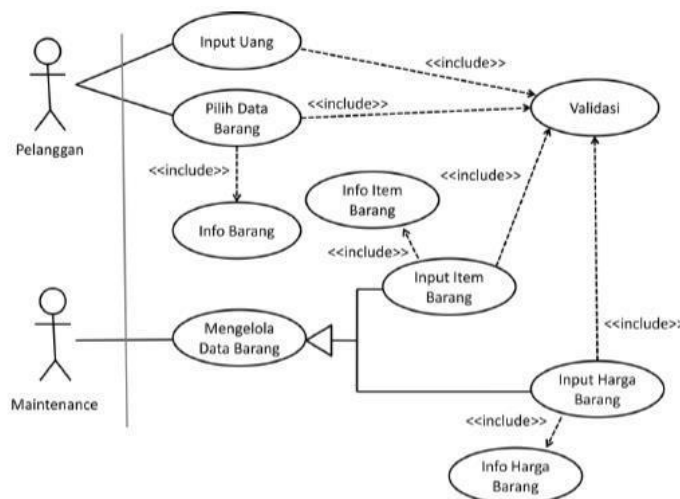
Animasi adalah simulasi gerakan yang dihasilkan dengan menayangkan rentetan *frame* ke *layer*. *Frame* adalah satu gambar tunggal pada rentetan gambar yang membentuk animasi.

2.10 Unified Modelling Language (UML)

UML menggunakan berbagai macam diagram dengan fungsi masing-masing untuk menggambarkan setiap proses dari sistem berorientasi objek. Berikut merupakan beberapa diagram UML diantaranya [13]:

2.10.1 Use Case Diagram

Use Case Diagram merupakan pemodelan yang digunakan untuk menggambarkan kelakuan (behavior) dari sistem yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Secara kasar, *Use Case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Contoh dari *Use Case Diagram* dapat dilihat pada gambar 2.1.



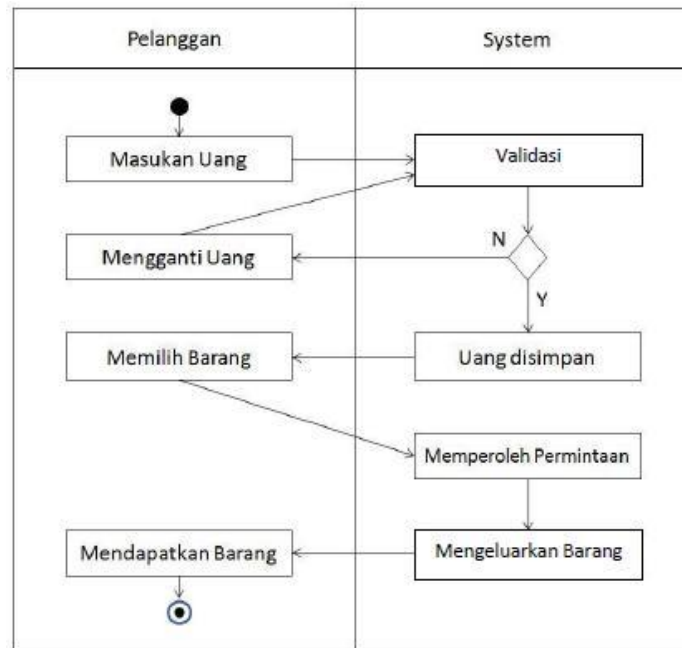
Gambar 2.1 Contoh Use Case Diagram

2.10.2 Activity Diagram

Activity Diagram adalah sebuah diagram yang menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Dalam diagram aktivitas yang perlu diperhatikan adalah bahwa diagram aktivitas menggambarkan aktivitas sistem, bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. *Activity diagram* juga banyak digunakan untuk mendefinisikan hal-hal berikut :

1. Rancangan proses bisnis di mana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.

2. Urutan atau pengelompokan tampilan dari sistem/*user interface* di mana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
3. Rancangan pengujian di mana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya. Contoh dari *activity diagram* dapat dilihat pada gambar 2.2.

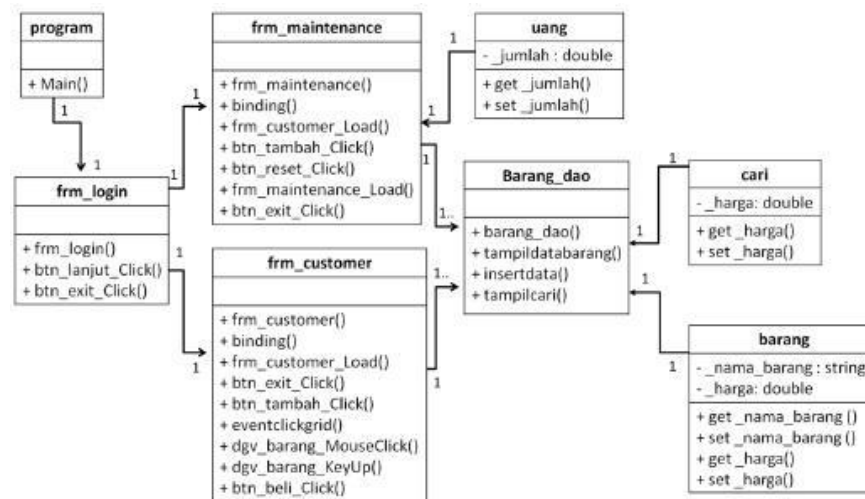


Gambar 2.2 Contoh Activity Diagram

2.10.3 Class Diagram

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi.

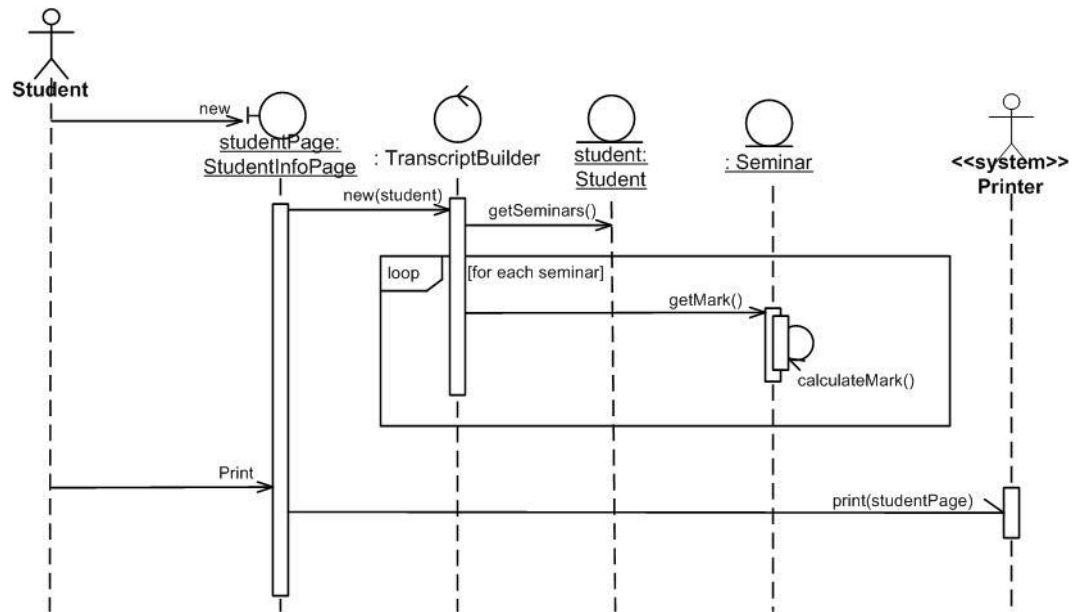
1. Atribut merupakan variabel-variabel yang dimiliki oleh suatu kelas
2. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Contoh dari *Class Diagram* dapat dilihat pada gambar 2.3.



Gambar 2.3 Contoh Class Diagram

2.10.4 Sequence Diagram

Sequence Diagram adalah diagram yang menggambarkan kelakuan objek pada *Use Case* dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antarobjek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *Use Case* beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Banyaknya diagram sekuen yang harus digambarkan adalah sebanyak pendefinisian *Use Case* yang memiliki proses itu sendiri atau yang penting semua *Use Case* yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada *Sequence Diagram* sehingga semakin banyak *Use Case* yang didefinisikan maka diagram sekuen yang harus dibuat juga semakin banyak. Contoh dari *Sequence Diagram* dapat dilihat pada gambar 2.4.



Gambar 2.4 Contoh Sequence Diagram

2.11 Game Engine

Game Engine adalah sebuah perangkat lunak yang dirancang untuk membuat sebuah game. Sebuah game engine biasanya dibangun dengan mengenkapsulasi beberapa fungsi standar yang umum digunakan dalam pembuatan sebuah game. Misalnya, fungsi *rendering*, pemanggilan suara, *network*, atau pembuatan partikel untuk *special effect*. Sebagian besar *game engine* umumnya berupa *library* atau sekumpulan fungsi-fungsi yang penggunaannya dipadukan dengan bahasa pemrograman.

Pada beberapa kasus *game engine* kadang pula disebut sebagai *middleware*. Hal ini dikarenakan karena *game engine* digunakan sebagai perantara antara bahasa pemrograman dengan format data dari berbagai perangkat lunak penghasil asset. Hal ini tentu saja memudahkan developer karena tidak perlu repot lagi memikirkan cara mengolah data yang dihasilkan dari berbagai perangkat lunak yang berbeda.

Sebuah *game engine* juga biasanya dilengkapi dengan fungsi-fungsi grafis atau komputasi dasar yang jarang diketahui oleh developer. *Game engine* membungkus itu semua untuk menjadikan proses pembuatan menjadi lebih cepat dan mudah. Beberapa *game engine* jenis ini diantaranya adalah TrueVision, Irrlicht dan Ogre3D [14].

2.11.1 Perkembangan *Game Engine*

Di awal tahun 2000, *game engine* mengalami perkembangan yang cukup signifikan. Beberapa *game engine* mulai dilengkapi dengan *World Editor*. Sehingga alih-alih menggunakan perangkat lunak 3D seperti 3Dmax atau Blender, pembuatan level atau “dunia” *game* dapat dibuat melalui sebuah perangkat lunak tersendiri yang telah dirancang khusus untuk *game engine* tersebut.

Perkembangan ini bukannya tanpa masalah. Di satu sisi, *software* ini memudahkan para developer pemula yang membutuhkan kecepatan untuk melihat produk dari rancangan mereka. Tapi di sisi lain perangkat lunak ini bisa “membatasi” kemampuan developer untuk berkreasi lebih jauh. Sehingga bagi sebagian *developer*, membangun “dunia” *game* lewat *script* tetap puluhan kali lebih *powerfull*.

Terlepas dari kelebihan dan kekurangannya, *World Editor* sebagai bagian dari *game engine* tetap memiliki peran penting sebagai alat untuk mempercepat pembuatan sebuah level atau dunia. Selain itu, beberapa *game engine* bahkan telah membuat Integrated Development Environment (IDE) tersendiri untuk proses pengembangannya.

Dengan kata lain, *game engine* tersebut tidak lagi harus tergantung dengan *software development* seperti Delphi atau Ms. Visual C++ misalnya, karena *game engine* tersebut telah dilengkapi dengan editor dan *compiler* sendiri.

2.12 *Tools yang digunakan*

Tools yang digunakan dalam pembangunan *game* ini diantaranya adalah Construct 2, Adobe Photoshop CS6 dan C2Buildozer.

2.12.1 Construct 2

Construct 2 merupakan sebuah *game engine 2D* untuk *HTML 5* yang dikembangkan oleh Scirra Ltd, sebuah perusahaan *start up* yang bermarkas di London. *Game engine* ini dikembangkan dengan konsep *behavior & event attachment* sehingga logika dalam game dapat dibangun tanpa harus mengetikkan satu baris *coding* pun (cukup *drag-and-drop* saja).

Construct pertama kali dirilis dengan nama Construct Classic pada tahun 2007. Pada tahun 2011, Scirra merilis versi terbaru Construct yaitu Construct 2

(atau biasa juga dikenal dengan nama C2). Pada awal 2015, tim Scirra mengumumkan akan meluncurkan Construct 3 yang diklaim akan merevolusi editor yang digunakan serta mendukung lebih banyak sistem operasi seperti Mac dan Linux. Meskipun demikian, Construct 3 akan dengan tetap memiliki dukungan penuh terhadap seluruh *project* yang dibangun oleh Construct 2.

Sejak awal Construct 2 memang dirancang untuk memaksimalkan proses pengembangan *game* tanpa perlu melakukan *scripting/coding*. Tujuannya adalah untuk mempercepat proses pembuatan *game* dan membantu para *non-programmers* untuk menciptakan *game* impian mereka. Pengaturan logika pada Construct 2 disajikan dalam *Event Sheet* yang berisi berbagai metode dan fungsi dasar khas pembangunan sebuah *game*. Hasil akhir dari *game* yang dibangun oleh Construct 2 adalah *game* berbasis *HTML 5* yang dapat dijalankan di berbagai browser seperti Chrome, Firefox, Internet Explorer, Safari dan Opera [15].

2.12.2 Adobe Photoshop

Photoshop adalah sebuah *image editor* atau program manipulasi gambar yang berfungsi untuk membuat, menyunting, dan memodifikasi gambar-gambar digital yang terdapat di dalam komputer.

Photoshop merupakan program penyunting gambar yang paling umum digunakan dalam bidang manipulasi gambar. Photoshop memiliki keseimbangan antara kesederhanaan dalam penggunaan dengan keluwesan dan kemampuan *software* untuk melakukan proses-proses manipulasi gambar yang rumit.

Photoshop lazimnya digunakan untuk membuat aspek grafis dari sebuah poster, brosur, majalah, halaman web, dan lain-lain. Dalam dunia fotografi, photoshop juga dapat digunakan untuk meningkatkan kualitas foto atau memperbaiki bagian-bagian yang rusak [16].

2.12.3 Apache Cordova

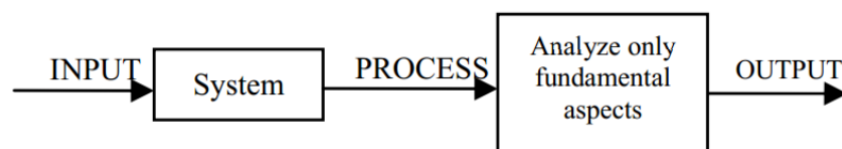
Apache Cordova adalah sebuah aplikasi yang dapat digunakan untuk membangun aplikasi *mobile* berbasis CSS3, HTML5, dan Javascript [17]. Aplikasi yang dibangun menggunakan Apache Cordova merupakan *package* aplikasi yang memiliki akses terhadap API *native*, dan dilihat menggunakan *web view*.

2.13 Teknik Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk menilai kelengkapan fungsionalitas program, serta mengevaluasi ketahanan sistem terhadap skenario yang dapat berujung terhadap kesalahan sistem. Melalui pengujian perangkat lunak, *programmer* dapat mengetahui apa yang harus diperbaiki pada sistem supaya sistem dapat berjalan secara utuh dan konsisten. Teknik pengujian yang digunakan adalah teknik *Black Box*.

Teknik *Black Box* adalah teknik pengujian untuk mengevaluasi fungsional program menggunakan kebutuhan fungsional program yang sudah didefinisikan di awal. Metode *Black Box* memungkinkan pengembang perangkat lunak untuk menilai dan mengamati secara rinci apakah *input* yang diberikan kepada program sudah sesuai dengan kebutuhan fungsionalnya. *Black Box* dapat menemukan kesalahan dalam kategori berikut :

- a. Fungsi-fungsi yang tidak berjalan atau hilang
- b. Kesalahan pada antarmuka
- c. Kesalahan dalam struktur data atau akses basis data eksternal
- d. Inisialisasi dan kesalahan terminasi
- e. Validitas fungsional
- f. Sensitifitas terhadap nilai input tertentu
- g. Batasan dari suatu data



Gambar 2.5 Sistem Kerja dan Teknik Pengujian *Black Box*

Berikut ini adalah ciri-ciri *black box testing*

1. *Black box testing* berfokus pada kebutuhan fungsional pada *software*, berdasarkan pada spesifikasi kebutuhan dari *software*.
2. *Black box testing* bukan teknik alternatif daripada *white box testing*. Lebih daripada itu, ia merupakan pendekatan pelengkap dalam mencakup *error* dengan kelas yang berbeda dari *metode white box testing*.
3. *Black box testing* melakukan pengujian tanpa pengetahuan detail struktur *internal* dari sistem atau komponen yang dites. juga disebut sebagai *behavioral testing*, *specification-based testing*, *input/output testing* atau *functional testing*

2.14 Usability Testing

Usability testing adalah sebuah metode pengujian yang bertujuan untuk mengevaluasi kegunaan suatu aplikasi berdasarkan kinerja penggunaan *user* [18]. Metode-metode *usability testing* yang digunakan dikembangkan oleh Jakob Nielsen.

Jakob Nielsen merumuskan sepuluh metode yang dapat digunakan untuk mengukur kegunaan atau *usability* suatu aplikasi:

1. Evaluasi Heuristik
2. Pengukuran Performa
3. *Thinking Aloud* (Bertukar Pikiran)
4. Observasi
5. Kuesioner
6. Wawancara
7. *Focus Group*
8. Pencatatan Penggunaan yang Sesungguhnya
9. *Feedback* pengguna

Metode yang digunakan dalam penelitian ini adalah wawancara, dimana lima orang pengguna diwawancara mengenai pengalaman mereka dalam menggunakan aplikasi secara mendalam.