

BAB 2

TINJAUAN PUSTAKA

2.1 Profil SD YAS 1 Bandung

Sekolah Dasar Yayasan Atikan Sunda (YAS) Bandung merupakan salah satu Sekolah Dasar Swasta yang ada di Provinsi Jawa Barat. Berlokasi di Jl. R. Edang Soewanda, Padasuka, Kec. Cimenyan, Bandung, Jawa Barat. Sekolah ini berada di bawah naungan Yayasan Atikan Sunda (YAS) didirikan pada tahun 1980.

Perjalanan sejarah SD YAS 1 Bandung tidak terlepas dari perjalanan sejarah Yayasan Atikan Sunda (YAS). Yayasan Atikan Sunda merupakan salah satu Yayasan yang berkontribusi dalam bidang pendidikan terkemuka di Jawa Barat. Keberadaan Yayasan Atikan Sunda tidak dapat dilepaskan dari organisasi Paguyuban Daya Sunda. Salah satu tujuan dari Paguyuban Daya Sunda adalah untuk meningkatkan kesejahteraan masyarakat dan mencerdaskan kehidupan bangsa Indonesia serta berkontribusi dalam bidang pendidikan dan seni budaya. Maka pada tanggal 19 November 1955 didirikan Yayasan Atikan Sunda. Yayasan tersebut merintis SD YAS pada tahun 1979. Beberapa tokoh sunda yang terlibat didalamnya adalah R. Ema Bratakusumah, Endang Soewanda dan R. Danumiharja

SD YAS 1 didirikan pada tanah seluas 1,036 M² yang merupakan hak guna milik Pemerintah Kota Bandung. Alamatnya berada di Jl. R. Edang Soewanda, Padasuka, Kec. Cimenyan, Bandung. SD YAS 1 secara resmi berdiri pada tanggal 27 Juni 1980, sesuai dengan SK dari Kanwil Depdikbud Jawa Barat. Saat ini kepemimpinan SD YAS 1 Bandung dipegang oleh Novi Hendayani. Fasilitas yang dimiliki SD YAS 1 berupa 10 Ruang Kelas, 1 Perpustakaan dan 2 Sanitasi Siswa.

2.1.1 Visi dan Misi

2.1.1.1 Visi

Visi dari SD YAS 1 Bandung yaitu untuk mewujudkan pendidikan yang berkualitas dengan mengakar pada budaya sunda di landasi dengan keimanan dan ketaqwaan terhadap Tuhan Yang Maha Esa

2.1.1.2 Misi

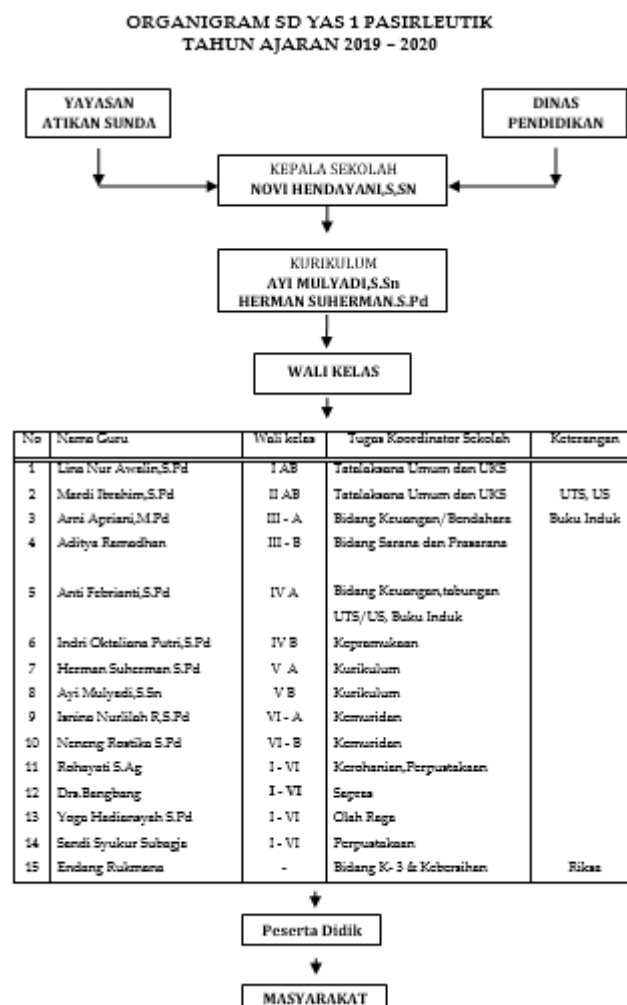
Misi dari SD YAS 1 Bandung adalah sebagai berikut :

1. Meningkatkan iman dan taqwa.

2. Memelihara dan melestarikan kesenian daerah.
3. Mengembangkan kurikulum lokal
4. Mengembangkan kegiatan ekstrakurikuler sebagai wadah pengembangan bakat, minat untuk mencapai prestasi
5. Meningkatkan profesional guru dalam pelayanan KBM
6. Pencapaian daya serap dan target kurikulum yang optimal
7. Menggunakan anggaran secara efektif dan efisien
8. Pemanfaatan media IT

2.1.2 Struktur Organisasi

Berikut adalah Struktur Organisasi yang berada di SD YAS 1 Bandung :



Gambar 2.1 Struktur Organisasi SD YAS 1 Bandung

2.1.3 Logo SD YAS 1 Bandung

Berikut adalah logo dari SD YAS 1 Bandung dapat dilihat pada gambar 2.2.



Gambar 2.2 Logo SD YAS 1 Bandung

2.2 Landasan Teori

2.2.1 Sekolah Dasar

Sekolah Dasar merupakan jenjang paling dasar pada Pendidikan formal di Indonesia. Sekolah Dasar ditempuh dalam waktu 6 tahun, yang diawali dari kelas 1 hingga kelas 6, hingga saat ini murid kelas 6 diwajibkan mengikuti Ujian Nasional yang merupakan salah satu persyaratan untuk melanjutkan ke jenjang selanjutnya.

Menurut Suharjo (2006: 1) sekolah dasar pada dasarnya merupakan Lembaga Pendidikan yang menyelenggarakan program Pendidikan enam tahun bagi anak-anak dalam usia 6-12 tahun” [4], berdasarkan Undang-Undang Nomor 20 Tahun 2003 tentang sistem Pendidikan Nasional bahwa “jenjang pendidikan dasar dan menengah adalah jenis pendidikan formal untuk peserta didik usia 7 sampai 18 tahun dan merupakan persyaratan dasar bagi pendidikan yang lebih tinggi” [5]. Jika usia anak pada saat masuk sekolah dasar merujuk pada definisi pendidikan dasar dalam undang-undang tersebut, maka pengertian sekolah dasar dapat dikatakan sebagai institusi pendidikan yang menyelenggarakan pendidikan dasar selama masa 6 tahun yang ditujukan bagi anak usia 7-12 tahun. Batasan usia 12 tahun inilah yang digunakan peneliti dalam melakukan penelitian.

2.2.2 Matematika

Matematika merupakan ilmu universal yang mendasari perkembangan teknologi modern, mempunyai peran penting dalam berbagai disiplin dan memajukan daya pikir manusia. Perkembangan pesat di bidang teknologi informasi dan komunikasi ini dilandasi oleh perkembangan matematika di bidang teori bilangan, aljabar, analisis, teori peluang dan matematika diskrit. Untuk menguasai dan mencipta teknologi di masa depan diperlukan penguasaan matematika yang kuat sejak dini [6].

2.2.3 Multimedia

Multimedia adalah pemanfaatan komputer untuk membuat dan menggabungkan teks, grafik, suara, gambar gerak (video dan animasi) dengan menggabungkan *link* dan *tool* yang memungkinkan pemakai melakukan navigasi, berinteraksi, berkreasi dan berkomunikasi [7].

Sedangkan menurut Hofstetter mendefinisikan multimedia sebagai penggunaan komputer untuk menampilkan dan menggabungkan teks, grafik, suara dan video dengan menggunakan *link* dan alat-alat lainnya yang memungkinkan pemakai untuk mengatur dan berinteraksi, membuat dan melakukan komunikasi. Definisi ini berisikan empat komponen yang penting untuk multimedia. Pertama, harus ada komputer untuk melakukan koordinasi apa yang dilihat dan didengar, dan yang berinteraksi dengan pengguna. Kedua, harus ada hubungan yang menghubungkan informasi. Ketiga, harus ada perangkat navigasi yang memungkinkan pengguna melintasi informasi yang terkoneksi melalui *web*. Yang terakhir, harus ada cara bagi pengguna untuk mengumpulkan, memproses dan mengkomunikasikan informasi dengan ide-ide pengguna. Jika satu dari komponen ini tidak ada atau hilang, maka tidak dapat disebut sebagai multimedia [8].

2.2.3.1 Elemen Multimedia

Sebagaimana telah dijelaskan pada definisi multimedia bahwa multimedia memiliki beberapa elemen. Berikut ini adalah elemen multimedia meliputi teks, suara, gambar, animasi, video [8].

1. Teks

Teks merupakan bentuk media yang paling umum digunakan dalam menyajikan informasi, baik yang menggunakan model baris perintah ataupun GUI. Teks dapat disajikan dengan berbagai bentuk font maupun ukuran.

2. Suara

Suara digunakan untuk menyajikan informasi tertentu, misalnya digunakan untuk memperdengarkan cara melafalkan sebuah kata dalam bahasa Inggris. Dengan bantuan suara, pengguna dapat mendengarkan bunyi suatu kata dengan tepat. Suara pada komputer disimpan dengan berbagai format.

3. Gambar

Umumnya, gambar disimpan dengan cara dimampatkan. Tujuannya adalah untuk menghemat ruang dalam penyimpanan eksternal. Untuk melakukan pemampatan ini, ada dua teknik yang digunakan. Yaitu teknik *lossless* dan teknik *lossy*. Teknik pemampatan *lossy* adalah suatu teknik yang memampatkan data sehingga gambar rekonstruksi hasil pemampatan mempunyai perbedaan dengan gambar asli, tetapi bagi mata manusia kelihatan sama. Sedangkan teknik pemampatan *lossless* adalah suatu teknik yang menghasilkan gambar rekonstruksi yang sama dengan gambar aslinya.

4. Animasi

Animasi adalah pembentukan gerakan dari berbagai media atau objek yang divariasikan dengan gerakan transisi, efek-efek, juga suara yang selaras dengan gerakan animasi tersebut atau animasi yang merupakan penayangan frame-frame gambar secara cepat untuk menghasilkan kesan gerakan.

5. Video

Video merupakan elemen multimedia paling kompleks karena penyampaian informasi yang lebih komunikatif dibandingkan gambar biasa. Walaupun terdiri dari elemen-elemen yang sama seperti grafik, suara, dan teks, namun bentuk video berbeda dengan animasi. Perbedaannya adalah dalam video informasi disajikan dalam kesatuan utuh dari objek yang dimodifikasi sehingga terlihat saling mendukung penggambaran yang seakan terlihat hidup.

2.2.4 *Game*

Dalam kamus bahasa Indonesia *game* adalah permainan. Sebuah permainan adalah sebuah sistem dimana pemain terlibat dalam konflik buatan, disini pemain berinteraksi dengan sistem dan konflik dalam permainan merupakan rekayasa atau buatan, dalam permainan terdapat peraturan yang bertujuan untuk membatasi perilaku pemain dan menentukan permainan. Permainan merupakan sebuah aktivitas rekreasi dengan tujuan bersenang-senang, mengisi waktu luang, atau berolahraga santai. Permainan biasanya dilakukan sendiri atau bersama-sama [9].

Permainan video (video game) adalah permainan yang menggunakan interaksi antarmuka pengguna melalui gambar yang dihasilkan oleh piranti video. Permainan video umumnya menyediakan sistem penghargaan misalnya skor yang dihitung berdasarkan tingkat keberhasilan yang dicapai dalam menyelesaikan tugas-tugas yang ada di dalam permainan. Sistem elektronik yang digunakan untuk menjalankan permainan video disebut platform, contohnya adalah komputer pribadi dan konsol permainan [9].

2.2.4.1 *Jenis-jenis Game*

Menurut Andrew Rollings dan Ernest Adam (2003) Jenis *game* dapat diklasifikasikan menjadi beberapa tipe antara lain [10]:

1. **Action Games**, biasanya meliputi tantangan fisik, teka-teki (*puzzle*), balapan, dan beberapa konflik lainnya. Dapat juga meliputi masalah ekonomi sederhana, seperti mengumpulkan benda-benda.
2. **Real Time Strategy (RTS)**, adalah game yang melibatkan masalah strategi, taktik, dan logika. Contoh *game* jenis ini adalah *Age of Empire*, *Warcraft*, dan sebagainya..
3. **Role Playing Games (RPG)**, kebanyakan game jenis ini melibatkan masalah taktik, logika, dan eksplorasi atau penjelajahan. Dan juga kadang meliputi teka-teki dan masalah ekonomi karena pada game ini biasanya melibatkan pengumpulan barang-barang rampasan dan menjualnya untuk mendapatkan senjata yang lebih baik. Contoh dari game ini adalah *Final Fantasy*, *Ragnarok*, *Lord of The Rings*, dan sebagainya.

4. **Real World Simulation**, meliputi permainan olahraga dan simulasi masalah kendaraan termasuk kendaraan militer. *Games* ini kebanyakan melibatkan masalah fisik dan taktik, tetapi tidak masalah eksplorasi, ekonomi, dan konseptual. Contoh nya seperti *game Championship Manager*.
5. **Contrustion and Management**, seperti *game Roller Coaster Tycoon* dan *The Sims*. Pada dasarnya adalah masalah ekonomi dan konseptual. *Game* ini jarang yang melibatkan konflik dan eksplorasi, dan hampir tidak pernah meliputi tantangan fisik.
6. **Advanture games**, mengutamakan masalah eksplorasi dan pemecahan teka-teki. Namun terkadang meliputi masalah konseptual, dan tantangan fisik namun sangat jarang.
7. **Puzzle games**, ditujukan untuk memecahkan suatu masalah tertentu. Hampir semua tantangan disini menyangkut masalah logika yang biasanya dibatasi oleh waktu.
8. **Slide scrolling games**, pada jenis *game* ini karakter dapat bergerak ke samping diikuti dengan gerakan *background*. Contoh *game* tipe seperti ini adalah *Super Mario*, *Metal Slug*, dan sebagainya.
9. **Shooting**, *Game* bertipe *Shooting* sebagian besar menggunakan *mouse* sebagai alat pengendalinya. Pada *game* ini pemain seolah-olah berperan sebagai penembak *FPS (First Person Shooter)* atau pemain mengendalikan seorang penembak *TPS (Third Person Shooter)*. Contoh *game* bertipe *Shooting* antara lain, *Duck Hunt*, *Counter Strike*, *Onimusha*, dan lain sebagainya.
10. **Fighting**, *Game* bertipe *fighting* pada dasarnya sama dengan *game* bertipe *action*, hanya saja *game* bertipe *fighting* pemain mengendalikan sebuah karakter untuk berkelahi dengan karakter lain sampai salah satu karakter kalah. Contoh dari *game* bertipe *fighting* antara lain, *Street Fighter*, *Tekken*, *Duel*, dan lain sebagainya.
11. **Racing**, *Game* bertipe *racing* pada dasarnya adalah sebuah permainan menggerakan kamera. Pemain diberikan sebuah kendaraan atau sejenisnya

untuk menempuh rute tertentu. Contoh dari *game* bertipe *racing* antara lain, *Need For Speed*, *Moto GP*, *Formula 1*, dan lain sebagainya.

12. **MMO (Massive Multiplayer Online)**, merupakan *game* yang dapat dimainkan secara Bersama-sama menggunakan koneksi *internet*. Hal yang diutamakan dalam sebuah *game* bertipe *MMO* adalah kebersamaan dengan pemain lain. Contoh *game* bertipe *MMO* antara lain, *Ragnarok*, *Gunbound*, dan *Seal*, dan lain sebagainya.

2.2.4.2 Game Edukasi

Game Edukasi adalah *game* yang khusus dirancang untuk mengajarkan pemain suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya [11].

Menurut Hurd dan Jenuings, perancangan *game* edukasi yang baik haruslah memenuhi kriteria dari *game* edukasi itu sendiri. Berikut ini adalah beberapa kriteria dari sebuah *game* edukasi, yaitu [11] :

1. Nilai Keseluruhan (*Overall Value*)

Nilai keseluruhan dari suatu *game* terpusat pada desain dan panjang durasi *game*. Aplikasi ini dibangun dengan desain yang menarik dan interaktif. Untuk penentuan panjang durasi, aplikasi ini menggunakan fitur *timer*.

2. Dapat Digunakan (*Usability*)

Mudah digunakan dan diakses adalah poin penting bagi pembuat *game*. Aplikasi ini merancang sistem dengan interface yang *user friendly* sehingga *user* dengan mudah dapat mengakses aplikasi.

3. Keakuratan (*Accuracy*)

Keakuratan diartikan sebagai bagaimana kesuksesan model/gambaran sebuah *game* dapat dituangkan ke dalam percobaan atau perancangannya. Perancangan aplikasi ini harus sesuai dengan model *game* pada tahap perencanaan.

4. Kesesuaian (*Appropriateness*)

Kesesuaian dapat diartikan bagaimana isi dan desain *game* dapat diadaptasikan terhadap keperluan *user* dengan baik. Aplikasi ini

menyediakan menu dan fitur yang diperlukan *user* untuk membantu pemahaman *user* dalam menggunakan aplikasi.

5. Relevan (*Relevance*)

Relevan artinya dapat mengaplikasikan isi *game* ke target *user*. Agar dapat relevan terhadap *user*, sistem harus membimbing mereka dalam pencapaian tujuan pembelajaran.

6. Objektifitas (*Objectives*)

Objektifitas menentukan tujuan *user* dan kriteria dari kesuksesan atau kegagalan. Dalam aplikasi ini objektivitas adalah usaha untuk mempelajari hasil dari permainan.

7. Umpan Balik (*Feedback*)

Untuk membantu pemahaman *user* bahwa permainan (*performance*) mereka sesuai dengan objek *game* atau tidak, feedback harus disediakan. Aplikasi ini menyajikan animasi dan efek suara yang mengindikasikan kesuksesan atau kegagalan permainan.

2.2.4.3 *Game Aksi-Petualangan (Action-Adventure)*

Genre *game action-adventure* adalah genre *game* yang menggabungkan elemen inti dari genre *game action* dan *game adventure*, terutama memiliki elemen-elemen penting seperti teka-teki. Genre *game action-adventure* membutuhkan banyak keterampilan fisik yang sama dengan *game action*, tetapi juga menawarkan alur cerita, banyaknya karakter, sistem inventaris, dialog, dan fitur-fitur lain dari *game adventure*. *Game* dengan genre *action-adventure* lebih cepat daripada *game action* asli, karena memiliki tantangan fisik dan konseptual. *Gameplay* masih mengikuti genre *game adventure* seperti mengumpulkan *item*, eksplorasi dan interaksi dengan seseorang, sering kali menghubungkan area-area penting dan pemecahan teka-teki. Sementara kontrolnya bergaya *arcade* seperti pergerakan karakter dan beberapa perintah aksi dan ada tujuan akhir di luar skor yang didapatkan. Dalam sebagian *game* dengan genre *action-adventure*, pemain mengontrol satu avatar sebagai tokoh protagonis. *Game action-adventure* biasanya menyertakan kombinasi elemen cerita yang rumit, yang ditampilkan untuk pemain menggunakan audio dan video. Cerita ini sangat bergantung pada pergerakan

karakter pemain, yang memicu peristiwa cerita dan mempengaruhi *gameplay* [12]. *Action-advanture* memiliki beberapa subgenre yaitu:

1. *First-Person Action-Advanture*

Genre yang memanfaatkan *gameplay first-person shooter*, melepaskan aksi secara konstan demi elemen-elemen penting pada *game* petualangan seperti pemecahan masalah dalam alur cerita dan plot yang kompleks.

2. *Third-Person Action-Advanture*

Genre yang memanfaatkan *gameplay Third-person shooter*, melepaskan aksi secara konstan demi elemen-elemen penting pada *game* petualangan seperti pemecahan masalah dalam alur cerita dan plot yang kompleks.

3. *Immersive Sim*

Biasanya permainan yang dimainkan dari sudut pandang *First-Person* yang mencakup unsur permainan peran dan permainan platform yang menciptakan banyak sistem permainan yang dapat digunakan pemain untuk menyelesaikan tujuan dalam berbagai cara, menciptakan rasa agensi pemain dan permainan yang muncul.

4. *Platform-Advanture Games*

Genre yang menekankan eksplorasi dan pemecahan teka-teki, tetapi juga menampilkan konvensi permainan platform tradisional.

5. *Isometric Platform-Advanture*

Genre yang menampilkan lingkungan yang dapat dijelajahi dengan bebas dengan *gameplay* tiga dimensi dan grafis dua dimensi menggunakan tampilan isometrik.

6. *Stealth Games*

Genre yang menekankan untuk menghindari deteksi oleh musuh daripada terlibat dalam pertempuran langsung, yang mengarah kepada penekanan terhadap eksplorasi dan memecahkan teka-teki dari jenis lain *game action*.

7. *Survival Horror*

Genre yang menekankan *Inventory Management* (Manajemen Persediaan) dan memastikan pemain memiliki cukup amunisi dan *item* pemulihan untuk bertahan hidup. Namun, *survival-horror* adalah genre tematik dengan

gameplay yang beragam, jadi tidak semua *game survival horror* memiliki fitur ini.

8. *Survival Games*

Genre yang memiliki kelangsungan hidup terbuka tanpa unsur supranatural yang dapat ditemukan di *game survival horror*.

2.2.5 *Unified Modeling Language (UML)*

Menurut Rosa dan Shalahuddin mendefinisikan UML (*Unified Modeling Language*) adalah “Salah standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisa dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek” [15]. Sedangkan Mulyani mengatakan UML (*Unified Modeling Language*) adalah “Sebuah teknik pengembangan sistem yang menggunakan bahasa grafis sebagai alat untuk pendokumentasian dan melakukan spesifikasi pada sistem”.

Dari beberapa penjelasan teori tersebut dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah bahasa yang sering digunakan untuk membangun sebuah sistem perangkat lunak dengan melakukan penganalisaan desain dan spesifikasi dalam pemrograman berorientasi objek. Ada hal-hal yang sangat mendasar di dalam model UML. Beberapa hal tersebut adalah sebagai berikut:

1. *Classes*, diuraikan sebagai sekelompok dari objek yang mempunyai atribut, operasi, dan hubungan semantik. Sebuah kelas mengimplementasikan 1 atau lebih *interfaces*. Sebuah kelas dapat digambarkan sebagai sebuah persegi panjang, mempunyai sebuah nama, atribut dan metode pengoperasiannya.
2. *Interfaces*, merupakan sebuah antarmuka yang menghubungkan dan melayani antar kelas dan atau elemen. *Interface* atau antarmuka mendefinisikan sebuah set atau kelompok dari spesifikasi pengoperasian, umumnya digambarkan dengan sebuah lingkaran yang disertai dengan namanya. Sebuah atarmuka berdiri sendiri dan umunya merupakan pelengkap dari kelas atau komponen.

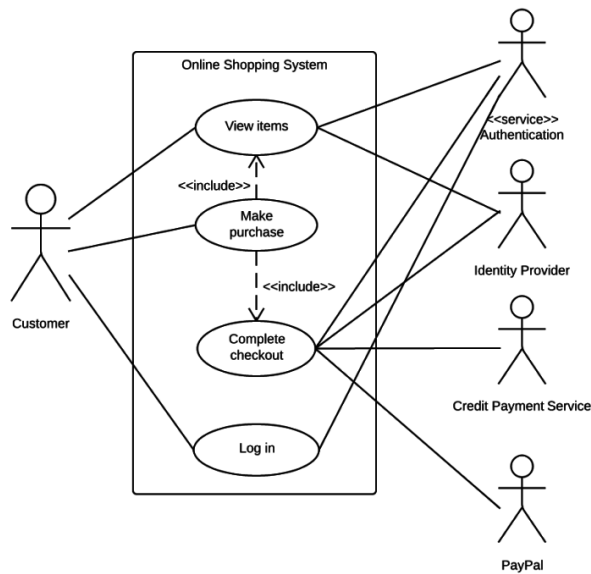
3. *Collaboration*, yang didefinisikan dengan interaksi dan sebuah kumpulan / kelompok dari kelas-kelas atau elemen-elemen yang bekerja secara bersama-sama.
4. *Use cases*, adalah uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. *Use case* digunakan untuk membentuk tingkah laku benda dalam sebuah model serta di realisasikan oleh sebuah *collaboration*.
5. *Nodes*, merupakan fisik dari elemen-elemen yang ada pada saat dijalankannya sebuah sistem, contohnya adalah sebuah komputer, umumnya mempunyai sedikitnya memori dan prosesor. Sekelompok komponen mungkin terletak pada sebuah node dan juga mungkin akan berpindah dari node satu ke node lainnya. Umumnya node ini digambarkan seperti kubus serta hanya mengandung namanya.

UML (*Unified Modeling Language*) memiliki diagram-diagram yang digunakan dalam pembuatan aplikasi berorientasi objek, diantaranya.

2.2.5.1 Use Case Diagram

Use Case Diagram menggambarkan apa saja aktivitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Yang menjadi persoalannya adalah apa yang dilakukan bukan bagaimana melakukannya. Diagram *use case* dekat kaitannya dengan kejadian-kejadian. Kejadian atau skenario merupakan contoh apa yang terjadi ketika seseorang berinteraksi dengan sistem. Diagram *use case* ini menjelaskan dan menerangkan kebutuhan atau *requirement* yang diinginkan *user*, serta sangat berguna dalam menentukan struktur organisasi dan model dari sebuah sistem.

Selama tahap desain, diagram *use case* berperan untuk menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model terdapat satu atau beberapa diagram *use case* yang menggambarkan fungsi sistem yang diharapkan. Contoh *use case* dapat dilihat pada gambar 2.3.



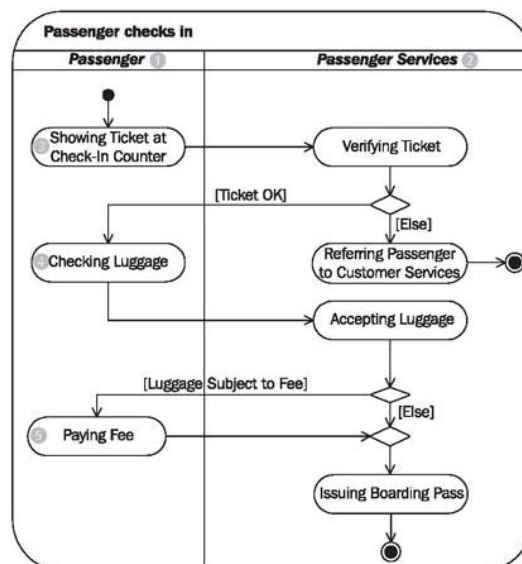
Gambar 2.3 Use Case Diagram

2.2.5.2 Activity Diagram

Activity Diagram sering digunakan oleh *flowchart*. Diagram ini berhubungan dengan diagram *statechart*. Diagram *statechart* berfokus pada objek yang dalam suatu proses atau proses menjadi suatu objek, *diagram activity* berfokus pada aktivitas-aktivitas yang terjadi yang terkait dalam suatu proses tunggal. Sebuah aktivitas dapat direalisasikan oleh salah satu *use case* atau lebih, aktivitas menggambarkan proses yang berjalan, sementara *use case* menggambarkan bagaimana aktor menggunakan sistem untuk melakukan aktivitas. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktivitas-aktivitas tersebut bergantung satu sama lain. Komponen-komponen yang ada pada *Diagram Activity* adalah sebagai berikut:

1. *Start State*, komponen yang berbentuk simbol lingkaran berwarna hitam yang solid. Komponen ini menunjukkan dimulainya *workflow* pada sebuah *diagram activity* dan hanya boleh ada satu *start point*.
2. *End State*, komponen yang berbentuk simbol lingkaran hitam yang seperti mata sapi atau *bull's eye*. Komponen ini menggambarkan akhir dari sebuah *diagram activity*. Berbeda dengan *start state*, komponen ini diperbolehkan lebih dari satu *end state*.

3. *Activity*, komponen yang berbentuk simbol belah ketupat atau biasa disebut *lozenge*. Komponen ini menggambarkan sebuah pekerjaan atau tugas dalam *workflow*.
4. *State Transition*, komponen yang berbentuk simbol garis lurus dengan panah di bagian ujungnya. Komponen ini menggambarkan kegiatan apa berikutnya setelah suatu kegiatan sebelumnya.
5. *Decisions*, komponen yang berbentuk *diamond* yang menggambarkan suatu titik pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi. Contoh *activity diagram* dapat dilihat pada gambar 2.4.

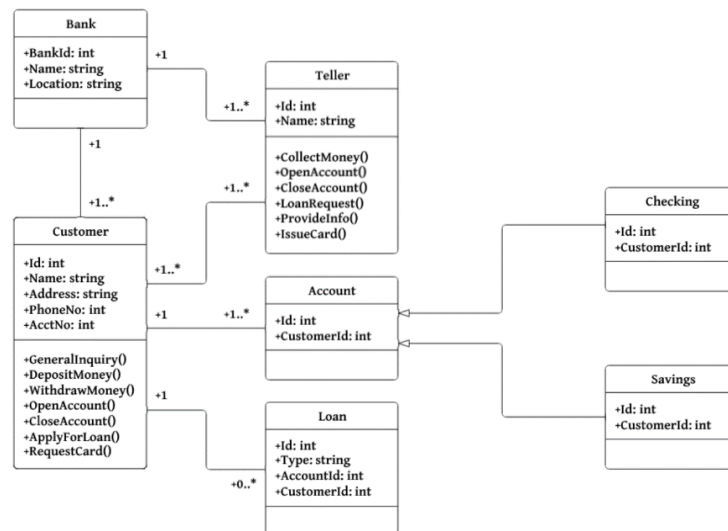


Gambar 2.4 Activity Diagram

2.2.5.3 Class Diagram

Class Diagram memberikan pandangan secara luas dari suatu sistem dengan menunjukkan kelas-kelasnya dan hubungan mereka. Diagram *class* bersifat statis, menggambarkan hubungan apa yang terjadi bukan apa yang terjadi jika mereka berhubungan. Diagram *class* ini mirip dengan ER-Diagram pada perancangan database, bedanya pada ER-Diagram tidak terdapat operasi atau metode tetapi hanya atribut. Diagram Class memiliki susunan struktur *class* yaitu sebagai berikut:

1. *Main Class*, kelas yang memiliki fungsi awal yang dieksekusi ketika sistem mulai berjalan.
 2. *Interface Class*, kelas yang mendefinisikan dan mengatur tampilan ke pemakai. Disebut juga dengan *Boundary Class*.
 3. *Defining Class*, kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian *use case*.
 4. *Entity Class*, kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.
- Contoh *class diagram* dapat dilihat pada gambar 2.5.



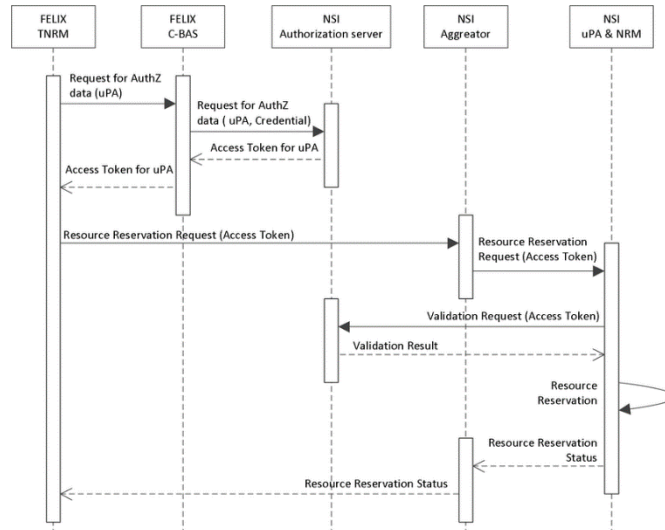
Gambar 2.5 Class Diagram

2.2.5.4 Sequence Diagram

Sequence Diagram merupakan salah satu diagram interaksi yang menjelaskan bagaimana suatu proses itu dilakukan, pesan apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Objek-objek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut. Biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diagram ini secara khusus berasosiasi dengan *Diagram Use Case* dan memperlihatkan tahap demi tahap

apa yang seharusnya terjadi. Komponen-komponen yang ada pada *Diagram Sequence* adalah sebagai berikut:

1. *Object*, komponen berbentuk kotak yang mewakili sebuah *class* atau objek. Komponen ini mendemonstrasikan bagaimana sebuah objek berperilaku pada sebuah sistem.
2. *Activation Boxes*, komponen yang berbentuk persegi panjang yang menggambarkan waktu yang diperlukan sebuah objek untuk menyelesaikan tugas. Lebih lama waktu yang diperlukan, maka *activation boxes* akan lebih panjang.
3. *Actors*, komponen yang bentuknya seperti *stick figure*. Komponen ini yang mewakili seorang pengguna yang berinteraksi dengan sistem.
4. *Lifeline*, komponen yang berbentuk garis putus-putus. *Lifeline* biasanya memuat kotak yang berisi nama dari sebuah objek berfungsi menggambarkan aktifitas dari suatu objek. Contoh *sequence diagram* dapat dilihat pada gambar 2.6.



Gambar 2.6 Sequence Diagram

2.2.6 Adobe Illustrator

Adobe *Illustrator* CS6 merupakan salah satu *software* untuk membuat desain grafis. *Software* ini sangat populer dan sudah diakui kecanggihannya. Kelengkapan fasilitas dan kemampuannya yang luar biasa dalam mendesain grafis, menjadikan *software* ini mulai banyak dipakai oleh para desainer komputer, karena keberadaannya benar-benar mampu membantu dan memudahkan pemakai dalam menyelesaikan pekerjaan desain grafis [13]



Gambar 2.7 Logo Adobe Illustrator CS6

2.2.7 Unity

Unity merupakan sebuah *game engine*, yaitu *software* pengolah gambar, grafik, suara, input, dan lain-lain yang ditujukan untuk membuat *game*. *Unity* merupakan *game engine* multiplatform yang mampu dipublikasi secara *standalone* (.exe), berbasis *web*, *android*, *IoT*, *XBOX*, maupun *PS3*, dengan catatan mendapatkan lisensi. *Unity* adalah *game engine* yang dibuat oleh *Unity Technology*. Kelebihan *unity* dibandingkan *game engine* lainnya adalah kemampuan membuat *game cross platform*. *Unity* dapat membuat berbagai macam *game*, seperti *Role Playing Game* (RPG), *shooter*, *racing*, dan lain sebagainya [12].



GAMBAR 2.8 Logo Unity

2.2.7.1 Sejarah dan Perkembangan *Unity*

Dalam beberapa tahun perkembangannya, sebelum dirilis, *Unity* telah diluncurkan pertama kali sebagai versi pra – rilis dengan GooBall sebuah video game yang didesain khusus untuk *Apple Macintosh*. GooBall, dengan *unity* pra – rilis, telah diluncurkan atau diumumkan pada bulan Maret tahun 2005, sementara itu *Unity* diluncurkan secara resmi sebagai aplikasi yang bersifat komersial pada dua bulan setelahnya yaitu bulan Juni tahun 2005. Satu tahun kemudian yaitu tahun 2006, aplikasi pengembang *game* ini telah menjadi nominasi untuk *Apple design awards* dalam kategori “*Best OS X Graphics*” [12]. *Unity* juga disebut sebagai aplikasi pengembang multiplatform, yang mana artinya *unity* mendukung untuk mengembangkan aplikasi *game* dan aplikasi yang lain untuk beberapa platforms seperti *game console*, *Mobile Phone* platforms, *Windows* dan *OS X* [12].

Sejak *unity* secara resmi dirilis sebagai *Unity* versi 1.0.1, banyak pembaharuan (*update*), *upgrades* dan fitur yang telah ditambahkan selama tahun perilisannya tersebut dan *Unity* terus berkembang secara terus – menerus. *Unity* hadir dengan beberapa pilihan perizinan (*license*), dari yang versi gratis untuk *Unity* dasar dan *Unity* pro yang dijual dengan harga \$1200. Baik *Unity* yang versi gratis dan *Unity* yang versi pro menawarkan banyak fitur yang dapat digunakan [12].

2.2.7.2 Fitur – Fitur Unity

Banyak kelebihan *game engine unity*, dapat dilihat dari beberapa fitur berikut ini [12] :

1. *Rendering*

Graphics engine yang digunakan adalah *Direct3D* (Windows, Xbox 360), *OpenGL* (Mac, Windows, Linux, PS3), *OpenGL ES* (Android, iOS), dan *proprietary APIs* (Wii). Ada pula kemampuan untuk *bump mapping*, *reflection mapping*, *parallax mapping*, *screen space ambient occlusion* (SSAO), *dynamic shadows* menggunakan *shadow maps*, *render-to-texture* dan *full-screen post-processing effects*.

2. *Scripting*

Script game engine dibuat dengan *Mono 2.6*, sebuah implementasi *opensource* dari *.NET Framework*. Programmer dapat menggunakan *UnityScript* (bahasa terkustomisasi yang terinspirasi dari *syntax ECMAScript*, dalam bentuk *JavaScript*), *C#*, atau *Boo* (terinspirasi dari *syntax* bahasa pemrograman *python*). Dimulai dengan dirilisnya versi 3.0, *Unity* menyertakan versi *MonoDevelop* yang terkustomisasi untuk *debug script*.

3. *Asset Tracking*

Unity juga menyertakan *Server Unity Asset* sebuah solusi terkontrol untuk *developer game asset* dan *script*. Server tersebut menggunakan *PostgreSQL* sebagai *backend*, sistem audio dibuat menggunakan *FMOD library* (dengan kemampuan untuk memutar *Ogg Vorbis compressed audio*), *video playback* menggunakan *Theora codec*, *engine* daratan dan vegetasi (dimana mendukung *tree billboarding*, *Occlusion Culling* dengan *Umbra*), *built-in lightmapping* dan *global illumination* dengan *Beast*, *multiplayer networking* menggunakan *RakNet*, dan navigasi *mesh* pencari jalur *built-in*.

4. *Platforms*

Unity support pengembangan ke berbagai platform. Didalam *project*, *developer* memiliki kontrol untuk mengirim perangkat *mobile*, *web*

browser, desktop, and console. Unity juga memungkinkan spesifikasi kompresi tekstur dan pengaturan resolusi di setiap platform yang didukung. Saat ini platform yang didukung adalah BlackBerry 10, Windows 8, Windows Phone 8, Windows, Mac, Linux, Android, iOS, Unity Web Player, Adobe Flash, PlayStation 3, Xbox 360, Wii U dan Wii. Meskipun tidak semua terkonfirmasi secara resmi, Unity juga mendukung PlayStation Vita yang dapat dilihat pada game Escape Plan dan Oddworld: New 'n' Tasty.

5. *Asset Store*

Diluncurkan November 2010, Unity Asset Store adalah sebuah resource yang hadir di Unity editor. Asset store terdiri dari koleksi lebih dari 4,400 asset packages, beserta 3D models, textures dan materials, sistem particle, musik dan efek suara, tutorial dan project, scripting package, editor extensions dan online service.

6. *Physics*

Unity juga memiliki support built-in untuk PhysX physics engine (sejak Unity 3.0) dari Nvidia dengan penambahan kemampuan untuk simulasi real-time cloth pada arbitrary and skinned meshes, thick ray cast, dan collision layers.

2.2.8 Microsoft Visual Studio

Microsoft Visual Studio merupakan sebuah perangkat lunak lengkap (*suite*) yang dapat digunakan untuk melakukan pengembangan aplikasi. Baik itu aplikasi bisnis, aplikasi personal, ataupun komponen aplikasinya dalam bentuk aplikasi *console*, aplikasi *Windows*, ataupun aplikasi *Web*. Visual Studio mencakup kompiler, SDK, *Integrated Development Environment* (IDE), dan dokumentasi (umumnya berupa *MSDN Library*). Kompiler yang dimasukkan ke dalam paket Visual Studio antara lain Visual C++, Visual C#, Visual Basic, Visual Basic .NET, Visual InterDev, Visual J++, Visual J#, Visual FoxPro, dan Visual SourceSafe.

Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam *native code* (dalam bentuk bahasa mesin yang berjalan di *Windows*) ataupun *managed code* (dalam bentuk Microsoft *Intermediate Language* di atas .NET

Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi *Silverlight*, aplikasi *Windows Mobile* (yang berjalan di atas *.NET Compact Framework*) [14].



GAMBAR 2.9 Logo Microsoft Visual Studio