

BAB II

LANDASAN TEORI

2.1 Makanan Sehat

Makanan yang sehat yaitu makanan yang higienis dan bergizi. Makanan yang higienis adalah makanan yang tidak mengandung kuman penyakit dan tidak mengandung racun yang dapat membahayakan kesehatan. Makanan yang bergizi yaitu makanan yang nilai gizinya sesuai dengan kebutuhan berdasarkan usia dan jenis bahan makanan. Makanan yang memiliki asupan gizi seimbang sangat penting dalam proses tumbuh kembang dan kecerdasan anak. Bahan makanan yang dikonsumsi harus mengandung gizi yang lengkap terdiri dari karbohidrat, lemak, protein, vitamin, mineral, dan air menurut Nurchayati dan Pusari (2014).

2.2 Game

Menurut Chowanda dan Dewi (2011), *game* didefinisikan sebagai salah satu aktivitas bermain, dimana terdapat pemain dan pemain berusaha untuk memenuhi tujuan sesuai dengan peraturan yang telah dirancang. *Game* atau permainan biasanya dilakukan untuk kesengangan atau kadang-kadang digunakan sebagai alat pendidikan, Amrullah (2012). Dan menurut Hanoraga (2013), *game* berasal dari kata bahasa Inggris yang memiliki arti dasar permainan. Permainan dalam hal ini merujuk pada pengertian kelincahan intelektual. Ada beberapa unsur yang harus diketahui tentang *game* dan akan dijelaskan selanjutnya.

2.2.1 Game Genre

Game genre atau jenis *game* dikategorikan berdasarkan perbedaan cara bermain daripada perbedaan secara visual dan naratif.

Beberapa *game* genre antara lain adalah *Action*, *Adventure*, *Role Playing*, *Simulation*, *Strategy*, *Educational* dan *Casual*.

Casual game merupakan *game* sederhana dengan peraturan yang *simple* atau bermain secara teknik. Tujuan *game* ini lebih pada hiburan, namun dengan komitmen terhadap *game* yang sangat rendah.

Pada perancangan Tugas Akhir ini *game* yang akan dibuat adalah *game* yang bergenre *casual*. Genre *casual* yang biasanya ditargetkan dengan pemain yang tidak ditentukan usianya. *Game casual* dapat digabungkan dari semua jenis permainan dan cocok di genre manapun.

Game bergenre *casual* dapat digunakan sebagai media pengajaran atau pengetahuan yang menarik bagi pengguna. Karena tertuju pada anak-anak maka bukan tingkat kesulitan yang dipentingkan akan tetapi waktu untuk menyelesaikan tantangan dari *game* tersebut.

Biasanya *game casual* dibedakan dengan aturan sederhana dan tidak memerlukan kompleksitas. Karena tidak memerlukan waktu panjang atau keahlian dalam bermain. Dan relatif rendah produksi dan distribusi bagi pengembang.

2.2.2 Game Design

Untuk merancang sebuah *game*, diperlukan empat elemen dasar yang membentuknya yaitu:

- 1) Mekanisme adalah sekumpulan prosedur atau aturan-aturan yang dimiliki oleh *game*, terdiri dari dunia, objek beserta atribut, aksi, peraturan, kemampuan dan peluang.

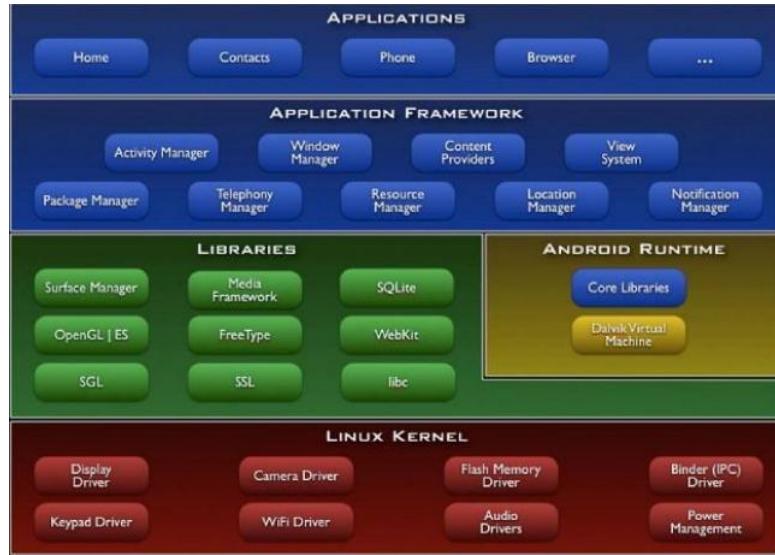
- 2) Storyboard adalah cerita yang menggunakan urutan kejadian-kejadian keseluruhan *game*.
- 3) Estetika adalah *game* tersebut memuat ide, cara berpikir serta semangat yang menjadi satu *game* yang berbeda atau mirip dengan yang lainnya.
- 4) Teknologi adalah untuk membuat *game* yang dapat melakukan hal-hal tertentu, membentuk esensi dari *game* tersebut, serta peletakan *game* itu sendiri.

2.2.3 Game Engine

Game engine adalah sistem perangkat lunak yang dirancang untuk menciptakan dan pengembangan video *game*. Ada banyak mesin permainan yang dirancang untuk berkerja pada konsol permainan video dan sistem operasi desktop seperti Microsoft Windows, Linux dan Mac Os X. fungsionalitas inti biasanya disediakan oleh mesin permainan mencakup mesin render untuk merender 2D atau 3D grafis, mesin tabrakan, suara, *script*, animasi, kecerdasan buatan, jaringan, *streaming*, manajemen memori, *threading*, dukungan lokalisasi dan adegan grafik. Proses pengembangan permainan sering dihemat oleh sebagian besar menggunakan *game engine* untuk menciptakan permainan sama atau berbeda (Ekasari, 2012).

2.3 Android

Android adalah sistem operasi untuk smartphone yang dibuat oleh perusahaan Android, Inc. sistem ini dikembangkan dengan memanfaatkan linux kernel siklus hidupnya merupakan logika dasar aliran dari sebuah aplikasi yang dibangun atau disebut platform (Amrullah,2012). Android juga menyediakan tool bagi pengembang yaitu SDK dan API. Android mempunyai arsitektur pada sistem operasinya. Berikut gambar arsitektur android.



Gambar 2.1. Arsitektur Android

2.4 Unified Modelling Language (UML)

UML adalah salah satu tool untuk mengembangkan sistem berorientasi objek. UML disebut sebagai bahasa pemodelan bukan metode. Kebanyakan metode terdiri paling sedikit prinsip, bahasa pemodelan dan proses. Bahasa pemodelan (sebagian besar grafik) merupakan notasi dari metode yang digunakan untuk mendesain secara cepat (Nurita, 2013).

Diagram dalam UML adalah diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah model sistem biasanya mempunyai beberapa diagram untuk setiap jenisnya.

Adapun jenis-jenis diagram antara lain:

2.4.1 UseCase Diagram

Menggambarkan sejumlah eksternal aktor dan hubungannya ke *usecase* yang diberikan oleh sistem. *UseCase* digambarkan hanya yang dilihat dari

luar oleh aktor (keadaan lingkungan sistem yang dilihat user) dan bagaimana fungsi yang ada didalam sistem.

2.4.2 Class Diagram

Menggambarkan struktur statis class dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Class dapat dihubungkan dengan lainnya melalui sejumlah cara: *assosiated* (terhubung satu dengan yang lain), *dependent* (satu class tergantung / menggunakan class yang lainnya), *specialized* (satu class merupakan spesialisasi dari class lainnya), atau *packaged* (grup bersama sebagai suatu unit).

2.4.3 Sequence Diagram

Menggambarkan kolaborasi dinamis antar sejumlah objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek.

2.4.4 Collaboration Diagram

Menggambarkan kolaborasi dinamis seperti sequence diagram. Dalam menunjukkan pertukaran pesan, *collaboration* diagram menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekanannya pada waktu atau urutan gunakan *sequence diagram*, tetapi jika penekannya pada konteks gunakan *collaboration diagram*.

2.4.5 Statechart Diagram

Menggambarkan suatu state (kondisi) yang dimiliki oleh suatu objek dari suatu kelas dan kejadian yang menyebabkan state berubah secara dinamis.

2.4.6 Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktivitas lainnya seperti usecase atau interaksi.

2.4.7 Component Diagram

Menggambarkan struktur fisik kode dari komponen. Komponen dapat berupa *source code*, komponen biner atau *executable component*. Sebuah komponen berisi tentang *logic class* atau class yang diimplementasikan sehingga membuat pemetaan dari *logical view* ke *component view*.

2.4.8 Deployment Diagram

Menggambarkan arsitektur fisik dari perangkat keras dan perangkat lunak sistem, menunjukkan hubungan komputer dengan perangkat(nodes) satu sama lain dan jenis hubungannya. Di dalam *nodes*, *executable*, *component* dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

2.5 Unity

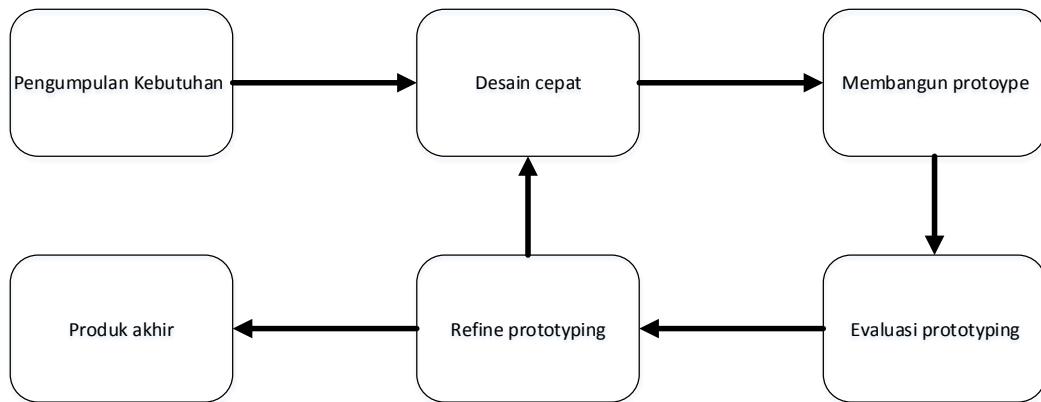
Unity *game engine* menurut Ekasari(2012) adalah *software* yang digunakan untuk membuat video *game* berbasis dua atau tiga dimensi dan dapat digunakan secara gratis, selain untuk membuat *game*, Unity juga dapat digunakan untuk membuat konten yang interaktif lainnya seperti, visual arsitektur dan real-time 3D animasi.

Selain sebagai *game engine* Unity juga dapat digunakan sebagai sebuah editor bagi *Game* yang sudah ada. Unity dibuat dengan menggunakan bahasa perogram C++, tapi pengguna tidak perlu menggunakan bahasa C++ yang sulit, karena Unity mendukung bahasa program lain seperti JavaScript, C#, dan Boo, Unity memiliki kemiripan dengan *Game engine* lainnya seperti, Blender *Game engine*, Virtools, Gamestudio, adapun kelebihan dari Unity, Unity dapat dioperasikan pada platform Windows dan Mac Os dan dapat menghasilkan *Game* untuk Windows, Mac, Linux, Wii, iPad, iPhone, google Android dan juga browser. Untuk browser, kita memerlukan sebuah *plugin*, yaitu Unity Web Player, sama halnya dengan Flash Player pada Browser. *Game* Unity juga mendukung dalam pembuatan *Game* untuk console *Game* Xbox 360 dan PlayStation 3.

2.6 System Development Life Cycle (SDLC)

SDLC mengacu pada model dan proses yang digunakan untuk mengembangkan sistem perangkat lunak dan menguraikan proses, yaitu pengembang menerima perpindahan dari permasalahan ke solusi.

Prototype merupakan metode pengembangan perangkat lunak yang dibangun sesuai dengan kebutuhan klien dengan tujuan untuk memungkinkan pengguna perangkat lunak untuk mengevaluasi hasil akhir produk yang telah dikomunikasikan.



Gambar 2.2 Model Pengembangan *SDLC Prototype*

Tahapan-tahapan prototyping

Tahapan dalam pengembangan prototyping adalah sebagai berikut:

1. Pengumpulan Kebutuhan

Pelanggan dan Pengembang bersama mendefinisikan format seluruh perangkat lunak, mengidentifikasi semua kebutuhan, dan garis besar sistem yang akan dibuat.

2. Membangun prototyping

Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan

3. Evaluasi prototyping

Evaluasi prototyping dilakukan oleh pelanggan apakah sudah dibangun sesuai kebutuhan.

4. Mengkodekan sistem

Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.

5. Menguji sistem

Setelah sistem menjadi perangkat lunak yang siap pakai, harus di tes dengan pengujian sistem dengan arsitektur pengujian yang ada.

6. Evaluasi sistem

Pelanggan mengevaluasi apakah sistem sudah sesuai yang diharapkan

7. Menggunakan sistem

Perangkat lunak yang telah diuji dan diterima siap untuk digunakan (Tuteja and Dubey, 2012).

