BAB 2
TINJAUAN PUSTAKA

2.1. Landasan Teori

Sebagai landasan dalam pembuatan laporan, maka dibutuhkan penjelasan tentang teori-teori yang berkaitan dengan permasalahan dan ruang lingkup pembahasan.

2.1.1. Makanan

Makanan adalah kebutuhan pokok manusia yang dibutuhkan setiap saat dan memerlukan pengelolaan yang baik dan benar agar bermanfaat bagi tubuh. Menurut WHO, yang dimaksud makanan adalah: "*Food include all substances, whether in a natural state or in a manufactured or preparedform, wich are part of human diet.*"

2.1.1.1. Manfaat Makanan Bagi Tubuh

Secara umum, terdapat 3 manfaat makanan bagi tubuh manusia yang biasa dikenal dengan tri guna makanan. Manfaat atau kegunaan itu antara lain makanan sebagai sumber tenaga (karbohidrat, lemak, protein), makanan sebagai sumber zat pembangun (protein, air), dan makanan sebagai sumber zat pengatur (vitamin, mineral).[1]

2.1.1.2. Kriteria Makanan Sehat

Makanan sehat adalah makanan yang mengandung gizi seimbang, kaya akan serat dan zat yang dibutuhkan untuk perkembangan tubuh. Kriteria makanan sehat adalah sebagai berikut.

1. Kuantitas cukup

   Banyaknya makanan bergantung kepada kebutuhan setiap orang sesuai dengan jenis dan lama kerja, berat badan, jenis kelamin, serta usia
2. Proposional

Jumlah makanan yang dikonsumsi harus sesuai dengan proporsi makan sehat seimbang, yakni karbohidrat 60%, lemak 25%, protein 15%, serta cukup vitamin, mineral, dan air.

3. Kualitas cukup

   Makanan yang mempunyai kandungan gizi yang terdiri dari karbohidrat, lemak, protein, vitamin, mineral, dan air.

4. Sehat dan higenis

   Makanan harus steril dan bebas dari kuman yag dapat memicu penyakit.

5. Makanan segar alami

   Sayur dan buah-buahan segar lebih menyehatkan jika dibandingkan dengan makanan pabrik.

6. Makanan golongan nabati

   Makanan golongan nabati lebih menyehatkan dibandingkan makanan hewani.

7. Dimasak tidak berlebihan

   Makanan yang dimasak secara berlebihan dapat menyebabkan hilangnya sejumlah vitamin dan mineral.

2.1.1.3. Makanan Tidak Sehat

Makanan tidak sehat (*Junk food)* adalah makanan cepat saji yang tidak sehat dan memiliki sedikit kandungan gizi dan berbahaya bagi kesehatan. Contoh dari *junk food* adalah ice cream, mie instan, permen, gorengan, hamburger, pizza, hot dog, sosis, fried chicken, hingga cokelat dan minuman soda. Adapun penyakit yang disebabkan oleh *junk food* ini adalah seperti gizi buruk, obesitas, diabetes, dan masalah pada sistem pencernaan.

2.1.2. *Game*

Game berasal dari kata bahasa inggris yang memiliki arti dasar "Permainan". Permainan dalam hal ini merujuk pada pengenian "kelincahan intelektual" (*intellectual playability*). *Game* juga bisa diartikan sebagai arena keputusan dan aksi pemainnya. Ada target-target dan misi untuk dapat dicapai pemainnya. Kelincahan intelektual, pada tingkat tertentu merupakan ukuran sejauh mana *game* itu menarik untuk dimainkan secara maksimal.[2]

Dari beberapa sumber buku dan penelitian, pengertian *game* adalah sebagai berikut:

1. Menurut IGDA (*International Garne Developers Association*) *Education SID*, *Game* adalah suatu kegiatan dengan beberapa *rules* atau aturan. *Game* merupakan bentuk permainan dimana sering kali (tetapi tidak selalu) melibatkan konflik, baik dengan pemain lain, dengan sistem permainan itu sendiri, atau dengan acak/ nasib / keberuntungan. Kebanyakan *game* memiliki tujuan, tapi tidak semua (Contoh: The Sints, SimCity). Kebanyakan *game* juga telah menentukan titik mulai dan titik akhir, tetapi tidak semua (Contoh: World of Warcraft).
2. Pengertian *game* menurut J. Huizinga dalam karya klasiknya Homo Ludens (1938) memberikan definisi pennainan adalah aktivitas yang berlangsung dalam beberapa batasan ruang dan waktu, dalam urutan yang terlihat, berdasarkan kepada aturan yang secara bebas diterima dan diluar lingkup kebutuhan.

2.1.2.1. Jenis *Game*

Jenis *game* biasa juga disebut *genre game*. Selain berarti jenis, *genre* juga berarti gaya atau format dari sebuah *game*. Jenis-jenis *game* adalah sebagai berikut.

1. Maze *Game*

   Jenis *game* ini biasanya menggunakan maze sebagai *setting* atau latar *game*. Jenis *game* Maze ini termasuk jenis *game* yang paling awal muncul. Contoh *game* ini adalah *game* Pacman dan Digger.



Gambar 2.1 *Game* Pacman

2. Board *Game*

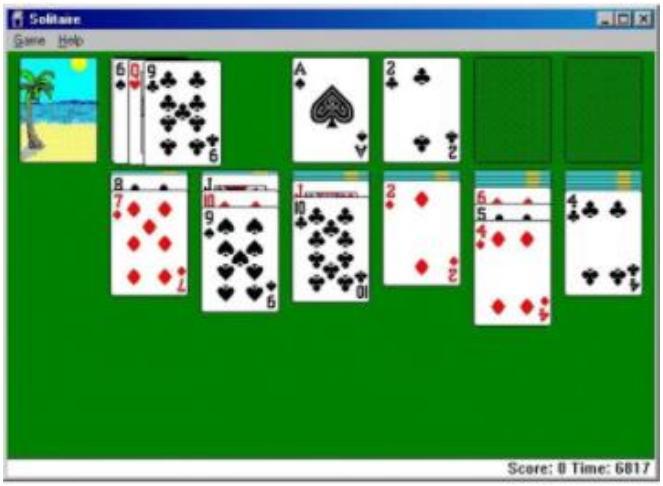
   Game jenis ini sama dengan *game board* tradisional seperti monopoli. Hanya saja permainan tradisional ini dimainkan melalui komputer.



Gambar 2.2 *Game* of Thrones

3. *Card Game*

   Jenis *game* kartu juga tidak jauh berbeda dari *game* tradisional aslinya. Namun, tampilannya lebih bervariasi dari versi tradisional. *Game* ini juga termasuk *game* yang awal muncul. Contoh *game* ini adalah Solitaire dan Hearts.



Gambar 2.3 Solitaire

4. *Battle Card Game*

   Contoh *game* ini yang populer yaitu Battle Card Pokemon. *Game* ini jarang ditemukan di Indonesia. Film kartun yang bercerita tentang permainan battle card ini pernah ditayangkan di stasiun televisi Indonesia.



Gambar 2.4 *Game Card* Pokemon

5. *Quiz Game*

   Game jenis ini merupakan *game* dengan bentuk kuis. Contoh *Quiz Game* yang pernah beredar yaitu *game* kuis Who Wants to Be Millionaire.



Gambar 2.5 *Game* Who Want To Be A Millionaire

6. *Puzzle Game*

   Jenis *game* ini memberi tantangan dengan cara menjatuhkan atau melenyapkan sesuatu dari sisi atas ke bawah atau dari kiri ke kanan. Contoh *game* ini adalah Tetris.



Gambar 2.6 *Game* Tetris

7. *Fighting Game*

   Jenis *game* ini sesuai dengan namanya berisi tentang pertarungan. Contoh *game* ini yaitu Street Fighter, Samurai Showdown, Virtual Fighter dan Kungfu.



Gambar 2.7 *Game* Street Fighter

8. *Racing Game*

   Racing game adalah *game* tentang mengendarai kendaraan dengan cepat. Contoh *game* ini yaitu Need for Speed Underground dan Toca Race Driver



Gambar 2.8 *Game* Toca Race Driver

9. *Simulation Video Game*.

   Genre ini merupakan bentuk permainan simulasi. Di sini pemain membangun sebuah area, kota, negara atau koloni. Contoh *game* ini yaitu Ship Simulator, Train Simulator, dan Crane Simulator.



Gambar 2.9 *Game* Ship Simulator

10. *Role Playing Game*

    Jenis *game* ini pemainnya memainkan sebuah tokoh atau karakter. Biasanya ada alur cerita yang harus dijalankan. Contoh *game* ini adalah Legacy of Kain, Blade of Sword, dan Beyond Divinity.



Gambar 2.10 *Game* Legacy of Kain

11. *Adventure Game*

    Adventure game merupakan genre *game* petualangan. Di sepanjang perjalanan pemain akan menemukan peralatan yang akan disimpan dan berguna sebagai petunjuk perjalanan. Contoh: *Game* Sam and Max atau Beyond and Evil.



Gambar 2.11 *Game* Sam and Max

12. *Game Sports*

Jenis *game* ini memiliki tema olahraga. *Game* yang mengetengahkan *genre* olahraga disebut *Sport Game*. [2]



Gambar 2.12 *Game* Soccer

2.1.2.2. *Game* Edukasi

Game edukasi adalah *game* yang khusus dirancang untuk mengajarkan pengguna terhadap suatu pembelajaran tertentu, pengembangan konsep dan pemahaman dan membimbing mereka dalam melatih kemampuan mereka, serta memotivasi mereka untuk memainkannya. [3]

2.1.2.2.1. Kriteria *Game* Edukasi

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

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. Karena aplikasi ini ditujukan untuk anak-anak maka desain antarmuka harus sesuai dengan nuansa anak-anak, yaitu menampilkan warna-warna yang ceria.

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 [3].

2.1.3. Android

Android merupakan sistem operasi perangkat *mobile* berbasis linux yang mencakup sistem operasi, middleware, dan aplikasi. Beberapa pengertian lain dari android, yaitu:

1. Merupakan *platform* terbuka (*Open Source*) bagi para pengembang (Programer) untuk membuat aplikasi.
2. Merupakan sistem operasi yang dibeli Google Inc, dari Android Inc.
3. Bukan Bahasa pemograman, akan tetapi hanya menyediakan lingkungan hidup atau *runtime environment* yang disebut DVM (Dalvik Virtual Machine) yang telah dioptimasi untuk *device*/alat dengan system memoti yang kecil.

2.1.3.1. Sejarah

Pada tanggal 5 November 2007, Android dirilis pertama kali. Android bersama OHA menyatakan mendukung pengernbangan *open source* pada perangkat mobile. Sekitar bulan September 2007, Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai sistem operasinya. Telepon selular ini diproduksi oleh HTC Corp. dan tersedia di pasaran pada tangga 5 Januari 2008.

Pada tanggal 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc., Garmin Ltd, Softbank, Sony Ericsson,

Toshiba Corp, dan Vodafone Broup Plc. Seiring dengan pembentukan OHA, OHA mengumumkan produk perdana Android mereka, perangkat *mobile* yang merupakan modifikast kernel Linux 2.6.

Pada masa sekarang ini banyak vendor-vendor yang menggunakan sistem operasi Android untuk Smartphone mereka, seperti HTC, Motoroi Samsung, LG, HKC, Huawei, Archos, Webstation Camangi, Dell, Nexu SciPhone, WayteQ, Sony Ericsson, Acer, Philip. s, Nexian, Asus, dan masih banyak lagi vendor-vendor lain.

Pada saat ini juga Android bersaing dengan Apple dalam sistem operasi untuk PC Tablet. Pada saat 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). [4]

2.1.3.2. Versi

Adapun versi-versi Android yang pernah dirilis adalah sebagai berikut:

1. Android versi 1.0
2. Android versi 1.5 *Cupcake*
3. Android versi 1.6 *Donut*
4. Android versi 2.0 *Eclair*
5. Android versi 2.2 *Froyo*
6. Android versi 2.3 *Gingerbread*
7. Android versi 3.0 *Honeycomb*
8. Android versi 4.0 *Ice Cream Sandwich*
9. Android versi 4.1 *Jelly Bean*
10. Android versi 4.4 *KitKat*
11. Android versi 5.0 *Lollipop*
12. Android versi 6.0 *Marshmallow*
13. Android versi 7.0 *Nougat*
14. Android versi 8.0 *Oreo*. [5]

2.1.4. *Object Oriented Programming*

Object Oriented Programming (OOP) merupakan salah satu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Keuntungan dalam menggunakan OOP ialah sebagai berikut:

1. Produktivitas meningkat

   Menggunakan OOP dapat meningkatkan produktivitas karena kelas dan objek dapat dipakai ulang untuk masalah-masalah yang melibatkan objek tersebut.
2. Cepat dalam pengembangan

   Menggunakan OOP dapat mempercepat pengembangan suatu perangkat lunak karena kesalahan dalam pengkodean akan berkurang pada saat analisis dan perancangan karena sistem dibangun dengan baik dan benar.
3. Mudah dalam pemeliharaan

   Menggunakan OOP dapat memudahkan dalam pemeliharaan karena dengan model objek, pola-pola yang cenderung tetap dan stabil dapat dipisahkan dan pola-pola yang mungkin sering berubah-ubah.
4. Terdapat konsistensi

   Dalam menggunakan OOP terdapat konsistensi karena sifat pewarisan dan penggunaan notasi yang sama pada saat analisis, perancangan maupun pengkodean.
5. Kualitas perangkat lunak meningkat

   Menggunakan OOP dapat meningkatkan kualitas perangkat lunak sebab pendekatan pengembangan lebih dekat dengan dunia nyata dan adanya konsistensi pada saat pengembangannya, perangkat lunak yang dihasilkan akan mampu memenuhi kebutuhan pemakai serta mempunyai sedikit kesalahan.

2.1.4.1. Konsep Dasar *Object Oriented Programming*

Dalam rekayasa perangka lunak, konsep OOP dapat diterapkan dalam tahap analisis, perancangan, pemograman, dan pengujian. Sistem OOP adalah sebuah sistem yang komponennya dienkapsulasi menjadi kelompok data dan fungsi.

Setiap komponen dalam sistem tersebut dapat mewarisi atribut dan sifat dan komponen lainnya dan dapat berinteraksi satu sama lain.

Konsep dasar yang harus dipahami dalam metodologi berorientasi objek adalah sebagi berikut:

1. Kelas (*class*)

   Kelas adalah kumpulan objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (*relationship*) dan arti.

2. Objek (*object*)

   Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek juga bisa merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

3. Metode (*method*)

   Metode (*method*) disebut juga service atau operator adalah prosedur atau fungsi seperti yang terdapat dalam bahasa pascal, tetapi cara kerjanya agak berlainan. Metode adalah subprogram yang tergabung dalam objek bersama-sama dengan atribut.

4. Atribut (*attribute*)

   Atribut adalah variable global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. [18] Atribut berarti properti, kualitas dan karakteristik yang dapat ditentukan pada orang atau barang dan data dimana objek atau kelas memiliki sebuah nilai.

5. Abstraksi (*abstraction*)

   Abtraksi atau *abstraction* secara sederhana dikatakan sebagai proses memilah beberapa atribut dan beberapa operasi suatu obyek hanya sampai pada yang benar-benar diperlukan saja, dan membuang atribut dan operasi yang tidak

diperlukan untuk persoalan yang dihadapi. Abstraksi bisa juga prinsip untuk merepresentasikan dunia nyata yang kompleks menjadi satu bentuk model yang sederhana dengan mengabaikan aspek-aspek lain yang tidak sesuai dengan permasalahan.

6. Enkapsulasi (*encapsulaion*)

   Pembungkusan atribut data dan layanan (operasi-operasi) yang dipunyai objek untuk menyembunyikan implementasi dan objek sehingga objek lain tidak mengetahui cara kerjanya.

7. Pewarisan (*inheritance*)

   Pewarisan merupakan mekanisme yang memungkinkan satu objek mewarisi sebagaian atau seluruh definisi dan objek lain sebagai bagian dan dirinya.

8. Polimorfisme (*polymorphism*)

   Polimorfisme yaitu konsep yang menyatakan bahwa sesuatu yang sama dapat mempunyi bentuk dan perilaku berbeda. Polimorfisme mempunyai arti bahwa operasi yang sama mungkin mempunyai perbedaan dalam kelas yang berbeda. Polimorfisme berarti suatu objek untuk digunakan di banyak tujuan yang berbeda dengan nama yang sama sehingga menghemat baris program [6].

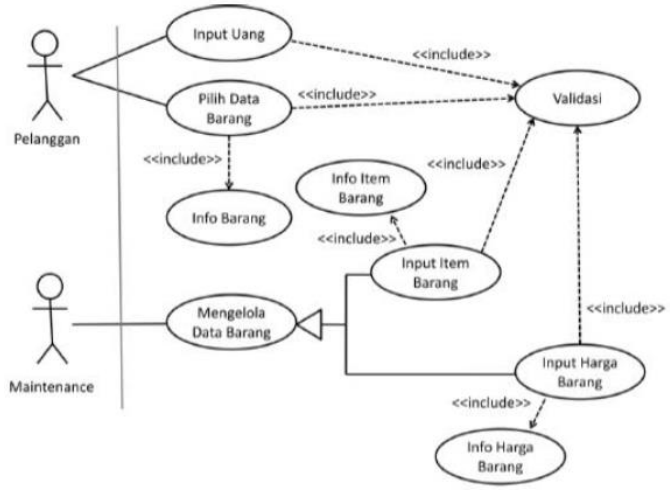
2.1.5. *Unified Modeling Language*

Unified Modeling Language (UML) adalah sekumpulan alat yang digunakan untuk melakukan abstraksi terhadap sebuah sistem atau perangkat lunak berbasis objek. UML merupakan singkatan dari *Unified Modeling Language*. UML juga menjadi salah satu cara untuk mempermudah pengembangan aplikasi yang berkelanjutan. Aplikasi atau sistem yang tidak terdokumentasi biasanya dapat menghambat pengembangan karena *developer* harus melakukan penelusuran dan mempelajari kode program. UML juga dapat menjadi alat bantu untuk *transfer* ilmu tentang sistem atau aplikasi yang akan dikembangkan dari satu *developer* ke *developer* lainya. Tidak hanya antar *developer* terhadap orang bisnis dan siapapun dapat memahami sebuah sistem dengan adanya UML. [6]

Dalam UML terdapat banyak diagram yang digunakan untuk melakukan pemodelan, disini penulis hanya menjelaskan beberapa diagram sesuai yang digunakan pada tahap perancangan yaitu sebagai berikut:

1. ***Use case Diagram***

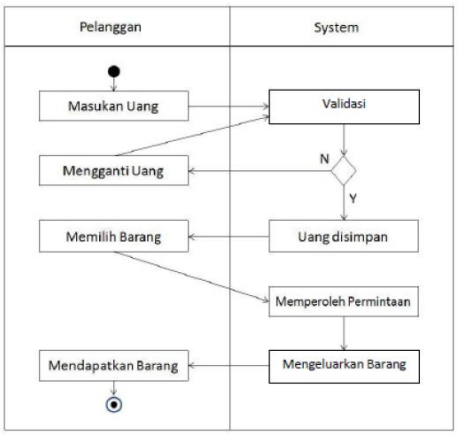
   Diagram ini menggambarkan kumpulan use case, aktor, dan hubungan mereka. *Use case* adalah hubungan antara fungsionalitas sistem dengan aktor internal/eksternal dari sistem.[8]



Gambar 2.13 Contoh *Use Case Diagram*

2. ***Activity Diagram***

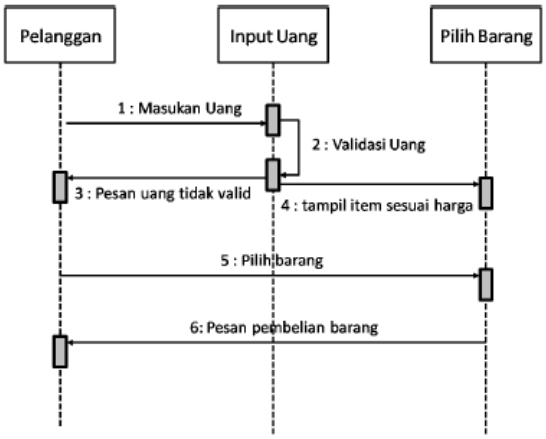
   Activity diagram menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.[8]



Gambar 2.14 Contoh *Activity Diagram*

3. Sequence Diagram

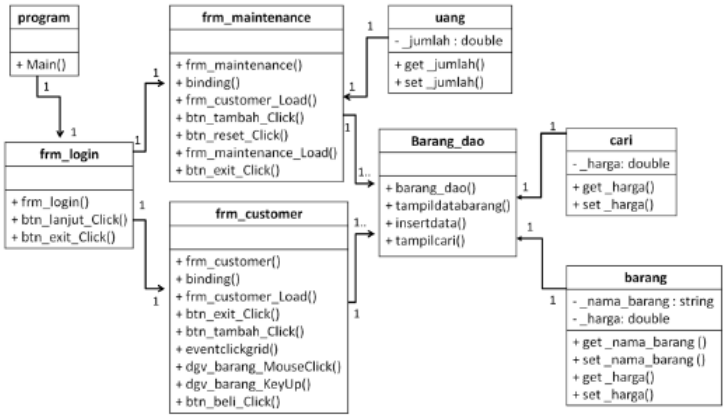
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atar dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait).[8]



Gambar 2.15 Contoh *Sequence Diagram*

4. Class Diagram

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).[8]



Gambar 2.16 Contoh *Class Diagram*

2.1.6. Perangkat Lunak yang Digunakan

Perangkat Lunak yang digunakan dalam pembangunan *game* edukasi pengenalan makanan sehat dan tidak sehat berbasis android ini diantaranya adalah Construct 2 sebagai *game engine* dan Adobe Photoshop sebagai *tool* pendukung.

2.1.6.1. Construct 2

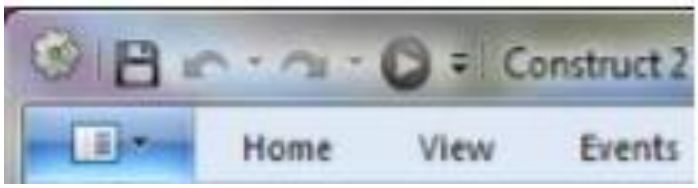
Construct 2 merupakan sebuah *game engine* berbasis HTML5 buatan Scirra yang berasal dari London, Inggris. Dengan menggunakan *game* engine ini, programmer pemula maupun expert dapat dengan mudah membuat sebuah *game*. Construct 2 dikhususkan pada *game* berbasis 2(dua) dimensi yang menyediakan banyak fitur untuk mempercantik *game* yang ingin dirancang. Dalam Construct 2 tersedia 70 visual effect yang menggunakan engine webGL, serta dilengkapi dengan 20 built-in plugin dan behaviours. Melalui Construct 2, programmer dapat mem-publish *game* nya melalui beberapa platform seperti HTML5 Website, Google Chrome Webstore, Facebook, Phonegap, Windows Phone 8.

Berikut ini beberapa komponen penting pada User Interface Construct 2.

1. *File menu* dan *ribbon tabs*

Pada bagian atas terdapat quick access toolbar yang terdiri dari fungsi-fungsi yang paling sering digunakan, terdiri dari save, undo, redo, dan preview. Toolbar ini dapat ditambahkan atau dikurangi sesuka hati sesuai kebutuhan dengan memilih tombol panah di sebelah preview. Tombol biru yang memiliki panah ke bawah

adalah file menu. Tombol-tombol di sebelahnya merupakan ribbon tabs yang memiliki fungsi masing-masing, tombol tersebut merupakan tombol drop-down untuk melakukan berbagai pengaturan pada objek.



Gambar 2.17 *File menu* dan *ribbon tabs*

2. *Layout view*

   Layout view merupakan *workspace* untuk programmer mulai merancang atau membuat *game*, mendesain, membuat menu dan lain sebagainya.

3. *View tab*

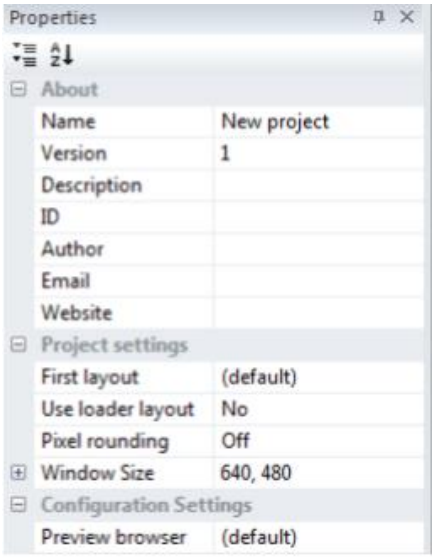
   View tab merupakan tab yang terdiri dari *layout* tempat programmer merancang desain *game* dan *event sheet* tempat dimana programmer melakukan validasi atau pengaturan *game*.



Gambar 2.18 *View Tab*

4. *Properties bar*

   Properties bar merupakan tempat mengatur *property* dari objek-objek yang terdapat dalam *game*. Terdiri dari *About, Project settings, dan Configuration settings.*



Gambar 2.19 *Properties bar*

5. *Projects bar* dan *Layers bar*

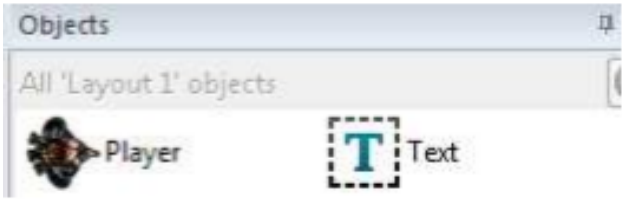
Projects bar terdiri dari segala sesuatu yang terdapat dalam projects, termasuk layout dan event sheet. Sedangkan layers bar merupakan tempat untuk mengatur banyaknya layer pada layout game serta menentukan layer yang aktif.



Gambar 2.20 *Projects bar* dan *layers bar*

6. *Objects bar*

Objects bar terdiri dari daftar objek-objek yang terdapat dalam layout game. Pada objects bar, programmer dapat melakukan pengaturan terhadap objek-objek tersebut. [9]



Gambar 2.21 Objects bar

2.1.6.2. Adobe Photoshop

Adobe Photoshop adalah sebuah *image editor* atau program penyunting gambar yang berfungsi untuk membuat, menyunting, dan memodifikasi gambar-gambar digital yang terdapat di dalam komputer. Photoshop merupakan program penyunting gambar yang paling hebat dan paling popular hingga saat ini. Kemampuan serta fasilitasnya yang lengkap membuatnya diminati oleh para seniman, profesional maupun pemula yang membutuhkan sebuah program gambar yang lengkap namun mudah dalam penggunaannya.

Adobe Photoshop saat ini digunakan untuk membuat gambar-gambar untuk keperluan seperti iklan, brosur, poster, seta berbagai macam *output* cetak lainnya. Bagi para penggemar photografi, photoshop juga dapat digunakan untuk me-*retouch* foto yang sudah rusak hingga memodifikasi foto menjadi lebih baik.[10]