

BAB II

TINJAUAN PUSTAKA

II.1 Rancang Bangun

Rancang Bangun (desain) adalah tahap dari setelah analisis dari siklus pengembangan sistem yang merupakan pendefinisian dari kebutuhankebutuhan fungsional, serta menggambarkan bagaimana suatu sistem dibentuk yang dapat berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi, termasuk menyangkut mengkonfigurasi dari komponen-komponen perangkat keras dan perangkat lunak dari suatu sistem. (Ria Apriyani Devina, 2010)

II.2 Aplikasi

Aplikasi adalah suatu program yang dibuat oleh pemakai yang ditujukan untuk melakukan suatu tugas khusus. Berdasarkan definisi di atas maka dapat disimpulkan bahwa aplikasi adalah program yang dibuat untuk melakukan tugas khusus dalam perusahaan. (Nety, dkk, 2010)

II.3 Game

Game atau permainan biasanya dilakukan untuk kesenangan dan kadang-kadang digunakan sebagai alat pendidikan. Untuk membuat sebuah game terlebih dahulu pembuat game harus membuat deskripsi yang menceritakan game yang akan dibuat. Selain itu dibutuhkan juga design game yang sederhana

untuk mempermudah pembuatan game. Dari design yang telah dibuat dapat diketahui semua elemen-elemen yang dibutuhkan dalam pembuatan game, misalnya karakter user, karakter musuh, animasi serangan dan sebagainya. Membuat game akan membutuhkan gambar dari tiap elemen-elemen yang ada, background image, dan lagu. Semua hal diatas dapat dikatakan sebagai resources game. (Ahmad Zuli Amrullah, dkk, 2012)

II.3.1 Jenis-jenis Game

Beberapa jenis-jenis game yaitu :

- a. Internet game adalah sebuah game yang didesain untuk bekerja melalui banyak jaringan-jaringan. Contoh game : Ragnarok Online
- b. Mini game adalah sebuah game yang sederhana, tidak membutuhkan banyak waktu, serta terkadang dapat membuat orang kecanduan. Contoh game : Dinner Dash, Sally Salon
- c. Racing game adalah game yang berjenis balapan yang biasanya teknik kita dapat memainkan dijadikan patokan keberhasilan dalam misi. Contoh game : Puppy Racing, Speed Warrior
- d. Fighting game adalah game yang biasanya ada dua karakter yang bertarung untuk memperoleh kemenangan atau tercapainya suatu misi. Contoh game : Weird Fighters
- e. Adventure game / Action Adventure adalah sebuah game dimana user dapat mengeksplorasi 1 tempat atau lebih, melawan musuh yang ditemui,

berinteraksi dengan karakter yang ditemui, dan terkadang memecahkan teka-teki. Contoh game : Summoner Saga

- f. Shooting game adalah game yang bertipe menembak musuh atau sasaran tertentu. Contoh game : Time crisis, House of The Dead
- g. Logic game adalah sebuah game yang mengharuskan user untuk berfikir menggunakan logika untuk dapat menyelesaikan game tersebut. Contoh game: Jeu Kakuro, Smart Kittio.
- h. Casual game adalah sebuah game yang dibuat untuk semua kalangan, dan dapat dimainkan setiap orang tanpa harus memiliki kemampuan khusus. Contoh game : Abduction! World. (Ahmad Zuli Amrullah, dkk, 2012)

II.4 Definisi *Puzzle Game*

Puzzle game adalah permainan yang terutama tentang pemecahan teka-teki, kadang-kadang tanpa memasukkan teka-teki menjadi alur cerita atau tujuan yang lebih besar. Itu tidak berarti bahwa mereka koleksi acak. Permainan Teka-teki biasanya variasi pada tema tunggal. Untuk menjadi sukses komersial, permainan puzzle harus menantang (tapi tidak terlalu keras), visual menarik, dan menyenangkan untuk dimainkan. Scott Kim adalah seorang desainer yang menciptakan teka-teki untuk media cetak, situs web, dan game komputer. Dia telah bekerja di lapangan selama bertahun-tahun dan telah mengidentifikasi delapan langkah dalam desain permainan puzzle. Empat langkah pertama terdiri dari proses menentukan aturan, sementara empat terakhir terdiri dari proses membangun teka-teki dan permainan itu sendiri. (Rollings And Adams, 2003)

Delapan langkah Scott Kim, yaitu :

1. Cari inspirasi : Hal ini dapat berasal dari berbagai sumber, termasuk permainan lainnya. Tetris, terinspirasi oleh permainan noncomputer disebut Pentominoes. Inspirasi bisa datang dari karya seni, cerita, atau subyek tertentu. Lain sumber inspirasi adalah dinamis bermain dari beberapa jenis: *flipping switch*, memutar tombol, geser benda di sekitar, atau mengambil mereka dan menempatkan mereka. Atau, ada dinamika yang lebih kompleks antara objek-objek: keseimbangan, refleksi, koneksi, dan transmisi.
2. Sederhana: Menyederhanakan tugas-tugas yang terlihat sulit agar dapat dengan mudah dimengerti dan dipahami oleh orang yang memainkannya.
3. Buat tetapan konstruksi : Satu-satunya cara untuk memastikan bahwa konsep bekerja teka-teki adalah untuk memainkannya. Dengan membangun sebuah prototipe kertas atau versi yang sederhana dalam sesuatu seperti Macromedia Flash untuk melihat apakah ia bekerja.
4. Tentukan aturan : Ini adalah bagian kunci dari desain teka-teki. teka-teki Kebanyakan ditandai dalam empat hal: papan, atribut, bergerak, dan tujuan atau kondisi kemenangan.
5. Membangun teka-teki : Sebuah teka-teki menantang pemain untuk mendapatkan solusi dari masalah yang ada, tentu saja, jalan tidak sederhana. Setiap teka-teki mengharuskan pemain membuat pilihan, beberapa di antaranya menyebabkan buntu. Teka-teki yang baik memerlukan wawasan dari pemain, bahwa "Aha!" saat itu terjadi ketika

pemain menyadari betapa puzzle bekerja dan bagaimana mengatasinya.

6. Test : Test digunakan untuk mengetahui apakah *puzzle* yang dibuat terlalu mudah atau sulit. Berusaha mencari kelemahan dari *gameplay* atau *user interface* agar terlihat baik bagi pemain.
7. Urutan rancangan : merancang *puzzle* secara berurutan secara linear. Memikirkan perpindahan *game* dari satu tempat ke tempat lainnya secara baik.
8. Presentasi : mempresentasikan *puzzle game* yang telah dibuat, mulai dari *story line*, *game play*, aturan main, dan ketentuan-ketentuan yang berlaku dalam *game*. (Rollings And Adams, 2003)

II.5 Konsep Aturan Permainan Game *Puzzle*

Puzzle adalah permainan yang terdiri dari 3 baris dan 3 kolom (9 kotak) dengan memiliki identitas berupa angka 1, 2, 3, 4, 5, 6, 7, 8 dan satu kotak kosong. Pergeseran dilakukan ke arah kotak kosong secara *vertical* atau *horizontal* tidak bias sembarangan, hanya satu kotak yang dapat digeser setiap kali perpindahan angka sampai susunan puzzle menjadi sempurna. Tujuan dari permainan puzzle adalah mengatur kondisi awal yang diperoleh secara acak menjadi seperti bentuk semula berupa urutan angka 1 sampai 8.

II.6 Android

Android adalah kumpulan perangkat lunak yang ditujukan bagi perangkat bergerak mencakup sistem operasi, middleware, dan aplikasi kunci. Android Standart Development Kit (SDK) menyediakan perlengkapan dan

Application Programming Interface (API) yang diperlukan untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Android dikembangkan oleh Google bersama Open Handset Alliance (OHA) yaitu aliansi perangkat selular terbuka yang terdiri dari 47 perusahaan Hardware, Software dan perusahaan telekomunikasi ditujukan untuk mengembangkan standar terbuka bagi perangkat selular. (Ahmad Zuli Amrullah, dkk, 2012)

II.6.1 Sejarah dan Perkembangan Android

Pada awalnya terdapat berbagai macam sistem operasi pada perangkat selular, diantaranya sistem operasi Symbian, Microsoft Windows Mobile, Mobile Linux, iPhone, dan sistem operasi lainnya. Namun diantara sistem operasi yang ada belum mendukung standar dan penerbitan API yang dapat dimanfaatkan secara keseluruhan dan dengan biaya yang murah. Kemudian Google muncul dengan platform Android, yang menjanjikan keterbukaan, keterjangkauan, open source, dan framework berkualitas. Pada tahun 2005, Google mengakuisisi perusahaan Android Inc. untuk memulai pengembangan platform Android. Dimana terlibat dalam pengembangan ini Andy Rubin, Rich Miner, Nick Sears, dan Chris White. Pada pertengahan 2007 sekelompok pemimpin industri bersama-sama membentuk aliansi perangkat selular terbuka, Open Handset Alliance (OHA). Bagian dari tujuan aliansi ini adalah berinovasi dengan cepat dan menanggapi kebutuhan konsumen dengan lebih baik, dengan produk awalnya adalah platform Android. Dimana Android dirancang untuk melayani kebutuhan operator telekomunikasi, manufaktur

handset, dan pengembang aplikasi. OHA berkomitmen untuk membuat Android open source dengan lisensi Apache versi 2.0.

Android pertama kali diluncurkan pada 5 November 2007, dan smartphone pertama yang menggunakan sistem operasi Android dikeluarkan oleh T-Mobile dengan sebutan G1 pada bulan September 2008. Hingga saat ini Android telah merilis beberapa versi Android untuk menyempurnakan versi sebelumnya. Selain berdasarkan penomoran, pada setiap versi Android terdapat kode nama berdasarkan nama-nama kue.(Ahmad Zuli Amrullah, dkk, 2012)

II.6.2 Versi Android

1. Android versi 1.1
2. Android versi 1.5 (Cupcake)
3. Android versi 1.6 (Donut)
4. Android versi 2.0/2.1 (Eclair)
5. Android versi 2.2 (Froyo: Frozen Yoghurt)
6. Android versi 2.3 (Gingerbread)
7. Android versi 3.0/3.1 (Honeycomb)
8. Android versi 4.0 (ICS :Ice Cream Sandwich)

II.7 Eclipse

Eclipse adalah IDE (integrated development Environment) merupakan aplikasi pengembangan java/android, eclipse memiliki plugin yang dapat membuat projec yang berbasis android. ADT (Android Development Tools) Eclipse sangat

diminati oleh para pengembang perangkat lunak, ini disebabkan karena eclipse merupakan software free dan open source. Eclipse juga menyediakan plugins untuk mengembangkan perangkat lunak yang digunakan untuk memodifikasi dan membuat aplikasi-aplikasi yang inovatif. Pada saat ini eclipse merupakan salah satu IDE favorite karena gratis dan open source berarti setiap orang boleh melihat kode program perangkat lunak ini. Selain itu kelebihan dari eclipse yang membuatnya populer adalah kemampuannya untuk dikembangkan oleh pengguna dengan membuat komponen yang disebut plugin. (Yazid Achyarudin, 2011)

II.8 UML (*Unified Modelling Language*)

Unified Modelling Language (UML) adalah definisi resmi dari bahasa pictorial, dimana terdapat simbol umum dan hubungan yang memiliki satu makna umum. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, system operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. UML lebih cocok untuk penulisan piranti lunak dalam bahasa-bahasa berorientasi objek karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya. (Kimmel, 2005)

II.8.1 Use Case Diagram

Use case diagram, menggambarkan tujuan penting fungsionalitas sistem yang akan ditampilkan pada user. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* merepresentasikan sebuah

interaksi antara aktor dengan sistem. *Usecase* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, *meng-create* sebuah daftar belanja, dan sebagainya. Seorang / sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. (Kimmel, 2005)

Use case diagram dapat membantu dalam penyusunan *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang *testcase* untuk semua *feature* yang ada pada sistem.

II.8.2 Class Diagram

Class diagram digunakan untuk memperlihatkan *class* pada sistem dan hubungan antara *class*. *Class diagram* tidak menjelaskan fungsionalitas atau bagaimana *instances class* berinteraksi. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi).

Class diagram menggambarkan struktur dan deskripsi *class*, *package*, dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. (Kimmel, 2005)

II.8.3 Activity Diagram

Activity diagram digunakan untuk menggambarkan alur aktivitas dari proses bisnis atau dari sebuah *use case* secara berurutan. Diagram ini juga digunakan untuk menggambarkan *action* (tindakan) yang akan dieksekusi ketika suatu proses sedang berjalan dan beserta hasil dari proses eksekusi tersebut.

Digunakan untuk menggambarkan aliran aktivitas baik dalam proses bisnis maupun *use case*. Diagram ini juga dapat digunakan untuk memodelkan aksi yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari aksi tersebut. (Kimmel, 2005)

II.8.4 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 atas dimensi vertikal (waktu) dan dimensi *horizontal* (objek-objek yang terkait).

Sequence diagram biasanya digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang men-*trigger* aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan. Masing-masing objek, termasuk aktor, memiliki *lifeline* vertikal. *Message* digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, *message* akan dipetakan menjadi operasi / metoda dari *class*. *Activation bar* menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah *message*. (Kimmel, 2005)