**BAB II**

**LANDASAN TEORI**

* 1. **Game**

Kata *game* berasal dari bahasa inggris yang berarti permainan. Dalam kamus besar bahasa indonesia, “Permainan adalah sesuatu yang digunakan untuk bermain; Barang atau sesuatu yang dipermainkan”. *Game* adalah permainan yang menggunakan media elektronik, merupakan sebuah hiburan yang berbentuk multimedia yang dibuat semenarik mungkin agar pemain bisa mendapatkan sesuatu sehingga adanya kepuasan batin (Yudhanto, 2010).

*Game* seperti halnya lagu atau film, sangat mudah dicari di *internet*. Bahkan bisa diunduh, dipakai dan disebarluaskan secara ilegal. Jenis – jenis *game* yang saat ini beredar di Indonesia, antara lain: MMO-RPG, *arcade*, *adventure*, *fighting*, *sport*, edugame dan lain sebagainya.

* 1. ***Arcade***

*Arcade game* adalah genre game yang mengandalkan ketangkasan, kedinamisan, dan penentuan pemainnya memegang kontrol. Ciri – ciri game arcade adalah mempunyai level singkat, kontrol yang mudah serta tingkat kesulitan yang bertambah dengan cepat (Sofiyah, 2016).

* 1. **Aritmatika**

Aritmatika atau aritmetika atau dulu disebut ilmu hitung merupakan cabang tertua matematika yang mempelajari operasi dasar bilangan(Laksono, AW, 2010).

* 1. **Android**

Android bukanlah robot berbasis artificial intelligence (AI), melainkan sistem operasi bergerak yang dewasa ini sangat terkenal(Kadir, 2013). Android adalah sistem operasi berbasis linux yang dirancang untuk perangkat bergerak layar sentuh (Yudhanto dan Wijayanto, 2017). Android adalah sebuah *software* untuk perangkat *mobile* yang mencakup sistem operasi, *middleware*, dan aplikasi kunci. Android *Set Development Kit* (SDK) menyediakan alat dan *Application Programming Interface* (API) diperlukan untuk memulai pengembangan aplikasi pada platform android menggunakan bahasa pemrograman java.

Android adalah sistem operasi berbasis linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh berbagai piranti bergerak. Awalnya, Google Inc membeli Android Inc., pendatang baru yang mebuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan piranti keras, piranti lunak, 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, sebuah lisensi perangkat lunak dan standar terbuka perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi android. Pertama yang mendapat dukungan penuh dari *Google Play Services* dan kedua adalah yang benar – benar bebas distribusinya tanpa dukungan langsung atau dikenal sebagai *Open Handset Distribution* (OHD).

* 1. **Android Studio**

Android studio adalah *Integrated Developing Environment* (IDE) resmi untuk pengembangan aplikasi Android dan bersifat open source serta didukung oleh Google (Andi Juansyah, 2015). Karena didukung langsung oleh Google, maka Google Play Services dapat kita tautkan dengan mudah kedalam proyek android kita. Android studio merupakan hasil pengembangan Google yang merekronstruksi IDE java yang bernama Intelij IDEA. Dikarenakan Google menghentikan dukungan pembaruan untuk IDE selain Android Studio dan Intelij IDEA. Bahkan Google menyatakan *depecrated* (tidak lagi dibutuhkan) pada IDE eclipse yang menjadi IDE pertama untuk android.

Android studio mendukung bahasa pemrograman java dan C++ untuk platform android. Android studio juga mendukung proses desain tampilan android baik itu *layout* ataupun *style*. Android studio memiliki sifat *auto save*, sehingga tidak perlu mengklik tombol *save*. Android studio menggunakan gradle pada java dan cmake pada C++ untuk mengatur kerja *compiler*.

* 1. **GNU Image Manipulation Program**

*GNU Image Manipulation Program* atau yang sering disebut GIMP adalah aplikasi grafis *open source* untuk pengolahan gambar. Tool yang dimiliki tidak kalah lengkap dengan tool grafis berbayar seperti *Adobe Photoshop*.

* 1. **Audacity**

*Audacity* adalah program pengolah suara / audio *open source* (gratis) yang disediakan oleh vendor *Audacity*, bagi para pengguna atau penggemar mixing lagu. Dibandingkan dengan pengolah audio berbayar lainnya, *Audacity* bisa dibilang cukup untuk mengolah audio, terutama bagi pemula. *Audacity* mengolah audio dengan cara memotong, memperbanyak, menyatukan *track* satu dengan yang lain, merekam suara atau memberikan efek khusus pada suara.

* 1. **SQLite**

SQLite adalah pustaka proses yang mengimplementasikan mesin *database* SQL mandiri, tanpa server, tanpa konfigurasi dan transaksional. Kode untuk SQLite ada di wilayah publik sehingga dapat digunakan untuk tujuan apapun entah komersial maupun pribadi. SQLite adalah *database* yang paling banya digunakan di dunia dengan lebih banyak aplikasi yang dapat kita hitung, termasuk beberapa proyek berprofil tinggi.

* 1. **Java**

Teknologi java adalah bahasa pemrograman dan platform. Bahasa pemrograman java adalah bahasa tingkat tinggi yang dapat dicirikan oleh semua kata kunci berikut: *Simple, Object oriented, Distributed, Multithreaded, Dynamic, Architecture neutral, Protable, High preformance, Robust, Secure*.

Dalam bahasa pemrograman java, semua kode sumber ditulis pertama kali dalam *file* teks biasa yang diakhiri dengan ekstensi .java. File-file sumber tersebut kemudian dikompilasi menjadi file .class dengan *compiler* javac. File .class tidak mengandung kode yang asli untuk prosesor anda. Melainkan berisi *bytecode* bahasa mesin virtual java.

Karena java VM tersedia di banyak sistem operasi yang berbeda, file .class yang sama dapat berjalan di Microsoft Windows, Sistem operasi Solaris, Linux atau MacOS. Beberapa mesin virtual, seperti Java SE HotSpot secara sekilas, melakukan langkah – langkah tambahan saat runtime untuk memberi aplikasi Anda dorongan kinerja. Ini termasuk berbagai tugas seperti menemukan *bottleneck* kinerja dan mengkompilasi ulang (ke kode asli) bagian kode yang sering digunakan.

* 1. **Flowchart**

Bagan alir (*flowchart*) adalah bagan (*chart*) yang menunjukan hasil (*flow*) didalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama untuk alat bantu komunikasi dan untuk dokumentasi (Jogiyanto, 2005)

*Flowcart* adalah bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelsaian suatu masalah. *Flowcart*merupakan cara penyajian dari suatu algoritma (Ladjamuddin, 2006).

Pedoman dalam menggambar suatu bagan alir, analis sistem atau pemrograman sebagai berikut;

* + 1. Bagan alir sebaiknya digambar dari atas ke bawah dan mulai dari bagian kiri dari suatu halaman
    2. Kegiatan didalam bagan alir harus ditunjukan dengan jelas
    3. Harus ditunjukan darimana kegiatan akan dimulai dan dimana akan berakhirnya
    4. Masing-masing kegiatan didalam bagan alir sebaiknya digunakan suatu kata yang mewakili  suatu pekerjaan, misalnya;“persiapkan” dokumen “hitung” gaji
    5. Masing-masing kegiatan didalam bagan alir harus didalm urutan yang semestinya
    6. Kegiatan yang terpotong dan akan disambung ketempat lain harus ditunjukan dengan jelas menggunakan symbol penghubung
    7. Gunakanlah symbol-simbol bagan alir yang standar.

Ada 5 macam menurut jogiyanto bagan alir diantaranya;

* + 1. Bagan alir sistem (*sistems flowchart*)

       Bagan alir sistem (*system flowchart*) merupakan bagan yang menunjukkan arus pekerjaan secara keseluruan dari sistem. Bagan menjelaskan urutan-urutan dari prosedure-prosedure yang ada dalam sistem. Bagan alir sistem menunjukan apa yang dikerjakan sistem.

* + 1. Bagan alir dokumen (*document flowchart*)

      Bagan alir dokumen (*document flowchart*) atau disebut bagan alir formulir (*form flowchart*) atau *paperwork flowchart* merupakan bagan alir yang menunjukan arus dari laporan dan formulir termasuk tembusan-tembusannya. Bagan alir dokumen ini menggunakan simbol-simbol yang sama dengan yang digunakan di dalam bagan alir sistem.

* + 1. Bagan alir skematik (*schematic flowchart*)

      Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem. Perbedaannya adalah bagan alir skematik menggunakan simbol-simbol bagan alir sistem , juga menggunakan gambar - gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir.

* + 1. Bagan alir program (*program flowchart*)

       Bagan alir program (*program flowchart*) merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

* + 1. Bagan alir proses (*process flowchart*)

Bagan alir proses (*process flowchart*) merupakan bagan alir yang banyak digunakan teknik industri. Bagan alir juga berguna bagi anilis sistem untuk menggambarkan proses dalam suatu prosedure.

Simbol dari bagan alir ( *flowchart* ) adalah seperti yang ditunjukkan pada tabel 2.1.

Tabel 2.1. simbol flowchart (Jogiyanto, 2009).

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Pengertian | Keterangan |
| 1. |  | Mulai / berakhir ( *Terminal* ) | Digunakan untuk memulai, mengakhiri, atau titik henti dalam sebuah proses atau program; juga digunakan untuk menunjukkan pihak eksternal. |
| 2. |  | Dokumen | Sebuah dokumen atau laporan; dokumen dapat dibuat dengan tangan atau dicetak oleh komputer. |
| 4. |  | Input / Output; | Digunakan untuk menggambarkan berbagai media input dan output dalam sebuah bagan alir program. |
| 5. |  | Disk Bermagnit | Data disimpan secara permanen pada disk bermagnit. |
| 6. |  | Penghubung Pada Halaman Berbeda | Menghubungkan bagan alir yang berada dihalaman yang berbeda. |
| 7. |  | Pemasukan Data On Line | Entri data alat oleh on line seperti terminal CRT dan komputer pribadi. |
| 8. |  | Pemrosesan Komputer | Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi |
| 9. |  | Arus Dokumen atau Pemrosesan | Arus dokumen atau pemrosesan; arus normal adalah ke kanan atau ke bawah. |
| No | Simbol | Pengertian | Keterangan |
| 10. |  | Keputusan | Sebuah tahap pembuatan keputusan |
| 11. |  | Penghubung Dalam Sebuah Halaman | Menghubungkan bagan alir yang berada pada halaman yang sama. |
| 12. |  | Dokumen Rangkap | Digambarkan dengan menupuk simbol dokumen dan pencetakan nomor dokumen dibagian depan dokumen pada bagian kiri atas. |

* 1. **Use Case Diagram**

Use case adalah rangkaian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor. Use case digunakan untuk membetuk tingkah laku benda/thing dalam sebuah mode serta direalisasikan oleh sebuah kolaborator (Yasin, 2012), yang di tunjukkan dengan tabel 2.4.

Tabel 2.2 Simbol- simbol use case (Yasin, 2012)

|  |  |
| --- | --- |
| SIMBOL | NAMA |
|  | Use Case |
|  | Actor |
|  | System Boundary |
| <<Include>> | Relasi antar Usecase |
| <<extends>> | Relasi antar Usecase |

**2.11.1 Bagian – bagian Use case**

Secara umum *Usecase* mempunyai bagian-bagian penting sebagai berikut:

1. Use case

merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unti atau actor, biasanya dinyatakan dengan kata kerja. Dinotasikan dengan gambar horizontal elipse dan tidak boleh ada 2 atau lebih nama usecase yang sama. Hubungan anatar usecase dihubungkan dengan garis putus-putus dan disebut *inheritance* atau *association*.

1. Actors

Actor dalam usecase memiliki beberapa peran sebagai berikut ;

1. Menggambarkan orang, sistem atau external entitas yang menyediakan atau menerima informasi dari sistem.
2. Aktor menggambarkan sebuah tugas atau peran dan bukan jabatan.
3. Aktor memberikan input atau menerima informasi.
4. Aktor biasanya menggunakan kata benda.
5. Tidak boleh ada komunikasi langsung antar actor.
6. Meletakkan actor utama di pojok kiri atas.
7. System boundary

System boundary digambarkan dengan kotak disekitar usecase, untuk menggambarkan jangkauan sistem. *System boundary* digunakan apabila memberikan beberapa alternative sistem yang dapat dijadikan pilihan. Sistem boundary dalam penggunaaannya bersifat optional.

1. Relationship

Terdapat 4 jenis relasi yang timbul pada usecase diagram yaitu:

1. Asosiasi antara *actor* dan *usecase*

Ujung panah pada asosiasi mengindikasikan siapa atau apa yang meminta interaksi. Menggunakan garis tanpa panah. Jika asosiasi menggunakan panah terbuka untuk mengindikasi bila actor berinteraksi secara pasif dengan sistem.

1. Asosiasi antar usecase

Hubungan antar usecase dihubungkan dengan garis putus-putus. Terdapat dua macam, yaitu “<<include>>” dan “<<extends>>”.

1. << include >>

Digunakan untuk usecase yang harus selalu menyertakan usecase lain. Tanda panah terbuka harus terarah ke sub usecase. Cara penggambaran association include biasanya secara horizontal.

1. <<extends>>

Merupakan alternative/ perluasan dari usecase lain jika kondisi atau syarat terpenuhi. Tanda panah terbuka harus terarah ke parent / base usecase. Gambaran association extend biasanya secara vertical.

1. Generalisasi/ inheritance antara usecase

Digambarkan dengan garis berpanah tertutup pada salah satu ujungnya yang menunjukkan lebih umum. Cara penggambarannya secara vertical dengan inheriting usecase dibawah parent usecasenya.