**BAB II**

**LANDASAN TEORI**

* 1. **Game**

Kata *game* berasal dari bahasa inggris yang berarti permainan. Permainan adalah sesuatu yang digunakan untuk bermain dan dimainkan dengan aturan – aturan tertentu (Jayandi, 2014). 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 pemain dalam memegang kontrol. Jenis *game arcade* dirancang untuk memancing adrenalin pemain serta tidak membutuhkan jalan cerita yang bagus, melainkan hanya dimainkan dengan bersenang – senang untuk mengejar *point* tertinggi / *highscore*. Ciri – ciri *game arcade* adalah mempunyai level yang singkat, kontrol yang mudah, serta tingkat kesulitan yang bertambah dengan cepat (Ashafidz, 2016).

* 1. **Aritmatika**

Aritmatika adalah cabang dari matematika. Aritmatika disebut juga ilmu hitung. Dalam ilmu hitung dibicarakan tentang sifat – sifat bilangan, dasar – dasar pengerjaan seperti menjumlah, mengurang, membagi dan mengalikan serta menarik akar dan lain sebagainya (Harahap, 1998).

Mental aritmatika adalah kemampuan menghitung yang tidak menggunakan alat bantu seperti mesin hitung, kalkulator, komputer, pena, sempoa, dan lain – lain. Namun hanya dengan menggunakan pemikiran didalam otak atau secara bayangan. Keunggulan mental aritmatika adalah:

1. Mengoptimalkan potensi otak
2. Meningkatan daya ingat dan konsentrasi
3. Membina minat pada pelajaran sekolah terutama matematika
4. Meningkatkan kecepatan berhitung
5. Menimbulkan ketekunan, kreatifitas dan percaya diri.
   1. **Android**

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) android yang didukung langsung oleh Google. 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**

Java adalah bahasa pemrograman yang berorientasi objek dan dapat dijalankan pada berbagai platform sistem operasi. Pada pemrograman android, file berekstensi java dikompilasi menjadi file *bytecode* berekstensi .apk dan kemudian dijalankan pada mesin *interpreter* Dlavik VM.

Java merupakan “bahasa pemrograman yang dikembangkan dari bahasa C++, sehingga bahasa pemrograman ini seperti bahasa C++”. Bahasa Java dapat dijalankan pada sebuah komputer dengan menggunakan software yang disebut dengan Compiler yang berfungsi untuk mengkonversikan kode sumber ke program biner yang berisi bytecode, kemudian interpreter digunakan untuk interpretasi dengan tujuan kode program yang tidak dapat dieksekusi tetapi tetap berjalan pada komputer yang sudah distandarisasikan, yang disebut Java Virtual Machine. Java pertama kali diciptakan oleh James Gosling dan Patrick Naughton pada suatu project dengan Green code di Sun Microsystem. Java kemudian diperkenalkan pada awal tahun 1996 dengan sebutan JDK 1.1 (Java Development Kit versi 1.1). Dalam pengembangan muncul Java 2 yang dikembangkan dari Java JDK 1.1 yang dilengkapi dengan Swing dengan teknologi GUI (Graphical User Interface) yang dapat menciptakan aplikasi berbasis desktop. Java terus dikembangkan oleh Sun, sehingga pada tahun 2006 penamaan platform dengan tujuan untuk mencerminkan tingkat kesempurnaan, stabilitas, skalabilitas serta keamanan atau security yang lebih baik lagi. Pada sebelumnya ialah Java 2 Platform, dengan Standar Edition 5.0 maka sekarang disederhanakan menjadi Java Platform, dengan Standar Edition 6 (Java SE6 atau Java 6) ( Hamdhani, 2015).

* 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. Bagan alir program dibuat dengan menggunakan simbol-simbol yang ditunjukkan pada gambar 2.1.

* + 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. |
| 3. |  | Input / Output; | Digunakan untuk menggambarkan berbagai media input dan output dalam sebuah bagan alir program. |
| 4. |  | Disk Bermagnit | Data disimpan secara permanen pada disk bermagnit. |
| 5. |  | Penghubung Pada Halaman Berbeda | Menghubungkan bagan alir yang berada dihalaman yang berbeda. |
| 6. |  | Pemasukan Data On Line | Entri data alat oleh on line seperti terminal CRT dan komputer pribadi. |
| 7. |  | Pemrosesan Komputer | Sebuah fungsi pemrosesan yang dilaksanakan oleh komputer biasanya menghasilkan perubahan terhadap data atau informasi |
| 8. |  | Arus Dokumen atau Pemrosesan | Arus dokumen atau pemrosesan; arus normal adalah ke kanan atau ke bawah. |
| No | Simbol | Pengertian | Keterangan |
| 9. |  | Keputusan | Sebuah tahap pembuatan keputusan |
| 10. |  | Penghubung Dalam Sebuah Halaman | Menghubungkan bagan alir yang berada pada halaman yang sama. |
| 11. |  | Dokumen Rangkap | Digambarkan dengan menupuk simbol dokumen dan pencetakan nomor dokumen dibagian depan dokumen pada bagian kiri atas. |

* 1. ***Use Case Diagram***

*Use-case diagram* adalah diagram yang membantu dalam menyusun *requirement* sebuah sistem, mengkomunikasikan rancangan dengan klien,dan merancang *test case* untuk semu fitur yang ada pada sistem.

Tabel 2.2 Simbol Use Case

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Pengertian | Keterangan |
| 1 |  | Actor | Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan *use case* |
| 2 |  | Dependency | Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (*independent*) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*depedent*) |
| 3 |  | Generalization | Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari induk |
| 4 |  | Use case | Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil |

* 1. ***Activity Diagram***

*Activity diagram* atau diagram aktifitas diperlukan untuk menggambarkan proses bisnis dan urutan aktifitas dalam suatu proses. Diagram aktifitas juga dipakai dalam *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau *data flow diagram* pada perancangan terstruktur. Diagram aktifitas sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan.

Tabel 2.3 Simbol *Activity Diagram*

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Pengertian | Keterangan |
| 1 |  | Initial | Memulai dan mengakhiri sebuah aktifitas |
| 2 |  | Action | Menunjukkan aktifitas yang dilakukan dalam odul sistem aplikasi |
| 3 |  | Transition | Memperlihatkan urutan eksekusi |
| 4 |  | Fork | Memecah sebuah *behavior* menjadi *activity* atau *action* yang paralel |
| 5 |  | Join | Menggabungkan kembali *activity* atau *action* yang paralel |
| 6 |  | Decision | Menunjukkan aktifitas yang harus dipilih apakah yang pertama atau kedua |

* 1. ***Class Diagram***

*Class diagram* adalah model statis yang menggambarkan struktur deskripsi *class* serta hubungan antara *class*. *Class diagram* mirip ER-Diagram pada perancangan *database*, bedanya pada ER-Diagram tidak terdapat operasi tetapi hanya atribut. *Class* terdiri dari nama kelas, atribut, dan operasi serta hak aksesnya.

Tabel 2.4 Hak Akses Class

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 | - | Private | Hak akses yang hanya bisa diakses oleh class itu sendiri |
| 2 | + | Public | Hak akses yang bisa diakses oleh semua class yang terhubung |
| 3 | # | Protected | Hak akses yang hanya bisa diakses oleh class yang bersangkutan dan turunannya |

Tabel 2.5 Hubungan Antar Class

|  |  |  |  |
| --- | --- | --- | --- |
| No | Simbol | Nama | Keterangan |
| 1 |  | Assosiation | Hubungan antar class dengan makna class 1 menggunakan class lain |
| 2 |  | Dependency | Hubungan antar class dengan makna kebergantungan antar class |
| 3 |  | Aggregation | Hubungan antar class dengan makna semua bagian |

Tabel 2.6 Nilai Kardinal

|  |  |  |
| --- | --- | --- |
| No | Simbol | Keterangan |
| 1 | 0..1 | Nol atau satu |
| 2 | 1 | Hanya satu |
| 3 | 0..\* | Nol atau lebih |
| 4 | 1..\* | Satu atau lebih |
| 5 | n | Lebih dari 1 |
| No | Simbol | Keterangan |
| 6 | 0..n | Nol sampai n |
| 7 | 1..n | Satu sampai n |