BAB II

LANDASAN TEORI

2.1 Aplikasi

Menurut Dhanta (2009:32), aplikasi *(application)* adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas- tugas tertentu, misalnya Microsoft Word, Microsoft Excel. Sedangkan menurut Anisyah (2000:30), aplikasi adalah penerapan, penggunaan atau penambahan Dari pengertian diatas, dapat diambil kesimpulan bahwa aplikasi adalah perangkat lunak yang dibuat untuk mempermudah suatu pekerjaan.

2.2 *Mobile*

Mobile adalah perangkat elektronik portabel yang berfungsi sebagaimana pesawat telepon normal, yang dapat bergerak pada suatu area yang luas. (bandingkan dengan *cordless phone*). Aplikasi bergerak merupakan istilah yang digunakan untuk suatu aplikasi yang dijalankan pada perangkat *mobile* yang cenderung memiliki ukuran kecil, portabel dan dapat terkoneksi melalui jaringan nirkabel. Perangkat *mobile* memiliki banyak jenis dalam hal ukuran, desain dan *layout*, tetapi mereka memiliki kesamaan karakteristik yang sangat berbeda dari sistem *desktop*. (Dikutip dalam jurnal Elsa 2013-01-10)

2.3 Pengertian Reservasi dan Tiket

Adapun pengertian reservasi dan tiket dapat dijabarkan sebagai berikut.

2.3.1. Reservasi

Reservasi adalah perjanjian pemesanan, penyediaan atau pembukuan pada suatu tempat dan periode waktu tertentu bagi calon pelanggan berikut dengan produk jasa tertentu. Pelayanan reservasi akan memberikan kemudahan dan kenyamanan kepada para pelanggan. Pelayanan reservasi akan memberikan jaminan kepastian mendapatkan tempat duduk yang merupakan tuntutan utama dari para calon penumpang, karena hal ini akan memberikan kepastian dan ketenangan bagi para calon penumpang untuk mencapai tujuan dalam menempuh perjalanan. Disamping itu bila ada penumpang yang memerlukan pelayanan permintaan fasilitas-fasilitas tambahan (seperti kursi roda, menu makanan khusus, dan lainnya) akan dapat terlayani dengan baik. Pada umumnya kegiatan reservasi dapat dilaksanakan dengan dua metode, yaitu reservasi dengan sistem manual dan dengan komputer. (Nurma Tiasaputri. 2010. Manfaat Gabrielle System Pada Reservasi Dan Tiketing Sriwijaya Air Distrik Solo. Laporan Tugas Akhir.)

2.3.2. Tiket

Tiket adalah suatu dokumen perjalanan yang dikeluarkan oleh suatu perusahaan yang berisi rute, tanggal, harga, data penumpang yang digunakan untuk melakukan suatu perjalanan. Menurut Rahmat Darsono, tiket adalah salah satu dokumen perjalanan yang dikeluarkan oleh maskapai penerbangan dan merupakan kontrak tertulis satu pihak yang berisikan ketentuan yang harus dipenuhi oleh penumpang selama memakai jasa penerbangan, dan data

penerbangan penumpang yang mempunyai masa periode waktu tertentu (Darsono, 2004 :15).

2.4 Android

Android adalah *operating system* atau OS berbasis linux yang diperuntukan khusus untuk *mobile device* seperti *smartphone* atau *PC table*, persis seperti *symbian* yang dipergunakan oleh Nokia dan BlackBerryOS, jelasnya seperti Microsoft Windows yang sangat dikenal baik oleh para pengguna komputer dan laptop, jika kita analogikan, Android adalah windowsnya sedangkan *smartphone* atau *handphone* atau tablet adalah unit komputernya.

2.4.1 Sejarah Android

The first truly open and comprehensive platform for mobile devices, all of the software to run a mobile phone but without the proprietary obstacles that have hindered mobile innovation. Android adalah sistem operasi untuk perangkat seluler yang awalnya dikembangkan oleh Android Inc. Perusahaan ini kemudian dibeli oleh Google pada tahun 2005. Android dibuat berdasarkan kernel Linux yang dimodifikasi.

2.4.2 Aplikasi android

Aplikasi Android ditulis dengan bahasa *Java*, dengan terminologi yang dapat dijelaskan sebagai berikut :

1. Aktifitas

   Aktifitas adalah sebuah layar tunggal dari *user interface* (UI) yang dapat memberikan fungsi UI sesuai kebutuhan pengguna. Sebagai contoh,

aplikasi email, dengan dua aktifitas, satu untuk daftar semua email yang sudah kita terima *(inbox)* dan aktifitas lainya untuk menulis dan mengirim email. Kedua aktifitas ini saling terkait tetapi bekerja secara bersamaan dalam satu aplikasi.

2. Layanan

   Sebuah layanan berjalan secara tidak tampak *(background progress)* yang tidak memiliki UI. Salah satu contoh layanan adalah ketika sistem beritahukan ada sms atau email yang diterima melalui layar.

3. Penyedia Konten

   Dengan penyedia konten kita dapat mengakses ke data yang disimpan dalam bentuk aplikasi perangkat lain atau bahkan membentuk aplikasi kita. Data ini dapat disimpan dalam *file* atau *database SQLite*.

4. *Layout*

   Layout adalah elemen penampung dalam UI. Setiap kali kita mendefinisikan sebuah *file xml* dengan UI tertentu, kita perlu menggunakan elemen ini untuk menahan struktur hirarki *tree* dari tampilan. Kita dapat menemukan berbagai jenis tata letak, tergantung sifat dan hubungan dengan unsur-unsur yang berlaku.

5. *View*

   Sebuah *view* adalah objek tunggal dari UI. Tampilan ini ditempatkan didalam *layout* dan dengan *view* kita bisa menciptakan UI mulai dari yang sederhana sampai yang paling komplek. Android memberi kita pilihan bekerja dengan *xml* untuk membuat UI.

6. *Event* (Pada klik atau Sentuh)

   Event adalah sesuatu yang terjadi dari luar dimana sesuatu yang terjadi diluar kendali atau kita tidak tahu kapan akan terjadi. Kita harus mendefinisikan apa yang diperintahkan oleh setiap pengguna. Untuk itu kita perlu menggunakan cara pengendalian untuk setiap kemungkinan yang akan terjadi dengan mendefinisikan bagaimana sistem melakukan respon. *Event* yang paling umum terjadi adalah *Event on Touch* atau ketika layar disentuh.

7. Sumber Data/*Resource*

   Sumber data adalah unsur-unsur eksternal yang digunakan dalam aplikasi. *File-file* sumber data ditempatkan dalam *folder* "res" bisa berupa gambar, musik dan video. Di *folder* ini disimpan *file xml*. UI, *styles* serta *string* yang kita gunakan dalam App. Setelah semua *file-file* tersebut tesedia maka akan membentuk sebuah *file* dengan nama *R.java, file R.java* merupakan *auto generate* yang tidak perlu dimodifikasi.

Aplikasi Android dijalankan di atas mesin virtual bernama *Dalvik Virtual Machine*. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang nantinya bisa digunakan oleh berbagai macam peranti bergerak. Untuk pengembang android, Google sendiri telah menyediakan SDK *(system development kit),* dan API *(application program interface)* yang membantu para developer untuk membangun aplikasi sendiri.

Keunggulan Android diantaranya :

1. Keterbukaan

Android menyediakan akses ke fungsi dasar perangkat *mobile* menggunakan standar panggilan ke API.

2. Penghancuran perbatasan

   Anda dapat menggabungkan informasi dari Internet ke dalam telepon, seperti informasi kontak, atau data pada lokasi geografis untuk mendapatkan kesempatan baru.

3. Kesamaan aplikasi

   Untuk Android ada perbedaan antara telepon utama aplikasi dan perangkat lunak lain, anda bahkan dapat mengubah program untuk memutar nomor, atau *screen saver*.

4. Cepat dan mudah perkembangan

   Dalam SDK memiliki semua yang anda butuhkan untuk membuat dan menjalankan aplikasi Android, termasuk simulator ini instrumen, dan alat *debugging* maju.

Google mengibaratkan Android sebagai sebuah tumpukan software. Setiap lapisan dari tumpukan ini menghimpun beberapa program yang mendukung fungsi-fungsi spesifik dari sistem operasi. Berikut ini susunan dari lapisan – lapisan tersebut jika di lihat dari lapisan dasar hingga lapisan teratas:

a. Linux Kernel

   Tumpukan paling bawah pada arsitektur Android ini adalah kernel.

b. Android Runtime

   Lapisan setelah Kernel Linux adalah *Android Runtime.Android Runtime* ini berisi *Core Libraries* dan *Dalvik Virtual Machine*. *Core Libraries*

mencakup serangkaian inti *library Java*, artinya Android menyertakan satu *set library-library* dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada *library-library* dasar bahasa pemrograman *Java*.

c. Libraries

Bertempat di level yang sama dengan *Android Runtime* adalah *Libraries*. Android menyertakan satu *set library- library* dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android.

d. Application Framework

Lapisan selanjutnya adalah *application framework*, yang mencakup program untuk mengatur fungsi-fungsi dasar *smartphone*. *Application Framework* merupakan serangkaian *tool* dasar seperti alokasi *resource smartphone*, aplikasi telepon, pergantian antar – proses atau program, dan pelacakan lokasi fisik telepon.

e. Application

Dilapisan teratas bertempat pada aplikasi itu sendiri. Dilapisan inilah anda menemukan fungsi-fungsi dasar *smartphone* seperti menelepon dan mengirim pesan singkat, menjalankan *web browser*, mengakses daftar kontak, dan lain-lain. Bagi rata-rata pengguna, lapisan inilah yang paling sering mereka akses. Mereka mengakses fungsi- fungsi dasar tersebut melalui *user interface*. Macam-macam versi Android:

a. Android versi 1.1
b. Android versi 1.5 (Cupcake)
c. Android versi 1.6 (Donut)

d. Android versi 2.0/2.1 (Eclair)

e. Android versi 2.2 (Froyo: Frozen Yoghurt)

f. Android versi 2.3 (Gingerbread)

g. Android versi 3.0/3.1 (Honeycomb)

h. Android versi 4.0 (ICS:Ice Cream Sandwich)

2.4.3 *Software* yang digunakan

Dalam membuat aplikasi *mobile* reservasi tiket *tour and travel* pada Anagata Taravel Bogor berbasis android ini diperlukan *software Eclipse* yang sudah terinstal *System development kit* (SDK), *Java development kit* (JDK), *Android development tool* (ADT) didalamnya, serta paket mySQL, dan paket XAMPP.

(Dikutip dari : Reto Meier, *professional android application development* hal 28 dalam Jurnal Dewi Rossa Fortina).

2.5 Pemograman Berorientasi Objek *(Object Oriented)*

Bahasa C++, Java, VB.NET dan sejenisnya kita kenal sebagai bahasa pemograman berorientasi objek. Tetapi hanya dengan menggunakan bahasa tersebut bukan berarti kita sudah pasti membuat program yang berorientasi objek.

2.5.1 Pengertian *Unified Modelling Language (UML)*

Unified Modelling Language adalah bahasa pemodelan standar. *(*Chonoles, 2003: bab1) mengatakan sebagai bahasa, berarti UML memiliki sintaks dan semantik. Ketika kita membuat model menggunakan konsep UML ada aturan-aturan yang harud diikuti. Bagaiamana elemen pada model-model yang kita buat berhubungan satu dengan yang lainnya dan harus mengikuti standar

yang ada. UML bukan sekedar diagram tetapi juga menceritakan konteksnya. Ketka pelanggan memesan sesuatu dari sistem, bagaimana transaksinya? Bagaimana keamanan terhadapa sistem buat? Dan sebagainya dapat dijawab dengan UML.

2.5.2 Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa UML menyediakan Sembilan jenis diagram, ada juga yang menyebutkan delapan jenis diagram. Namun demikian model-model itu dapat dikelompokkan berdasarkan sifatnya dinamis atau statis, diantaranya :

a. Diagram Kelas (*Class Diagram*)

   Bersifat statis, diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serat relasi-relasi. Diagram ini umum dijumpai pada pemodelan berorientasi objek. Meskipun statis, sering pula diagram ini memuat kelas aktif.

b. Diagram Paket (*Package Diagram*)

   Bersifat statis, diagram ini memeprlihatkan kumpulan kelas-kelas merupakan bagian dari diagram komponen.

c. Diagram *Use-Case*

   Bersifat statis, diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor.

d. Diagram *Sequences* (Urutan)

   Bersifat dinamis, diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalm waktu tertentu.

e. Diagram Komunikasi (*Communication Diagram*)

Bersifat dinamis, diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi structural dari objek-objek yang menerima serta mengirim pesan.

f. Diagram *Statechart (Statechart Diagram)*

Bersifat dinamis, diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status, transisi, kejadian serta aktivitas.

g. Diagram Aktivitas *(Activity Diagram)*

Bersifat dinamis, diagram aktivitas adalah tipe khusus dari diagram status yang memeperlihatkan aliran dari suatu aktivitask ke aktivitas lainnya dalam suatu sistem.

h. Diagram Komponen *(Component Diagram)*

Bersifat dinamis, diagram komponen ini memperlihatkan organisasi serta kebergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya.

i. Diagram *Deployment (Deployment Diagram)*

Bersifat statis, diagram ini memperlihatkan konfigrasi saat aplikasi dijalankan (*run time*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan.

(Dikutip dalam buku Prabowo Pudjo Widodo dan Herlawati, Oktober 2011, "*Menggunakan UML*".).