

BAB II

LANDASAN TEORI

A. Aplikasi

Pengertian Aplikasi Menurut Kamus Besar Bahasa Indonesia (Pradyta, 1998:52) adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna.

B. *Smartphone*

Pengertian yang menerangkan dari *Smartphone*, (Hariyanto, 2013) perangkat telekomunikasi serbaguna. *Smartphone* atau yang disebut juga piranti pintar dewasa ini menjadi trend yang mewabah di Indonesia, digemarinya *smartphone* ini juga bukan tanpa alasan, tapi karena fitur yang ditawarkan sangat menarik dan mengubah hobi pengguna untuk *browsing*, chatting dan semacamnya yang awalnya dilakukan secara statis, sekarang dengan *smartphone* bisa dilakukan dengan *mobile* atau bergerak.

Seperti halnya pengguna atau user yang semakin cerdas memilih perangkat pintarnya, vendorpun bersaing menawarkan fitur dan interface yang menarik dan mudah untuk penggunaannya, mereka menawarkan OS (*Operating System*) dan aplikasi yang memanjakan penggunaannya, RIM menawarkan OS *Blackberry*, Apple dengan *Iphonenya*, *Samsung*, *Sony*, *Huawei*, dan beberapa vendor lainnya

mengadopsi Android sebagai OSnya, ada juga yang memakai jasa Windows phone untuk OS, seperti yang dilakukan oleh Nokia.

Fisik dan geografis menjadi semakin tak berjarak, secara sosial pun masyarakat semakin tak bersekat. Sebuah kemajuan yang sangat patut disyukuri, *smartphone* merupakan pengembangan dari mobile phone atau lebih dikenal dengan handphone yang diciptakan awal mula sekitar 40 tahun lalu oleh Martin Cooper dari Motorola. Ponsel yang berbentuk besar sekali itu mempunyai berat sekitar 1,15 kg dan panjang sekitar 10 inch serta hanya bisa digunakan selama 20 menit saja sebelum baterainya habis. Namun dalam perkembangannya hingga jadi *smartphone* pada saat ini, dengan dimensi yang lebih elegan dan indah juga bisa lebih multi fungsi sehingga tidak hanya digunakan untuk telepon dan sms saja, kekuatan baterainya pun bisa lebih tahan lama hingga dapat bertahan beberapa hari *standby* dengan sekali *full charge*.

C. *Android Studio*

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup *system* operasi, *middleware* dan aplikasi. (Algindra, 2013) Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, Google Inc membeli Android Inc. Yang merupakan pendatang baru yang membuat piranti lunak untuk ponsel/*smartphone*.

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, Nvidia*.

Pengembangan aplikasi android membutuhkan *software development kit (SDK)*. Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk mengembangkan aplikasi pada *platform* Android yang menggunakan bahasa pemrograman Java (Nazruddin, 2012). Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang di-*release* oleh Google. Saat ini disediakan Android SDK sebagai alat bantu dan API untuk mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java.

Android memiliki 4 komponen aplikasi. Setiap komponen memiliki peran yang berbeda-beda dan berdiri sendiri meskipun terkadang salah satu komponen bergantung pada komponen lainnya (Mulyadi, 2010). Masing-masing komponen memiliki daur hidup masing-masing, yang mengatur kapan komponen tersebut dibuat dan bagaimana komponen tersebut dihapus. Untuk mengaktifkan komponen-komponen tersebut diperlukan sebuah pesan asinkron yang disebut *Intent* (Gargenta, 2011). Berikut adalah 4 tipe komponen dalam aplikasi android:

1. *Activity*, merupakan suatu komponen yang merepresentasikan sebuah tampilan yang dilengkapi dengan *user interface*.
2. *Service*, merupakan suatu komponen yang berjalan di balik layar untuk menjalankan operasi yang menghabiskan banyak waktu.

3. *Content Provider*, merupakan suatu komponen yang mengatur sekumpulan data aplikasi.
4. *Broadcast receiver*, merupakan komponen yang merespon terhadap siaran (*broadcast*) pengumuman yang dikeluarkan oleh sistem.

Android Studio adalah perangkat lunak baru yang dikembangkan oleh *Google* untuk menghasilkan aplikasi android. Perangkat lunak ini memiliki banyak fitur yang lebih besar tentang *google map* terkait kegiatan dari ECLIPSE. Bahasa java merupakan dasar platform untuk pembuatan aplikasi.

D. *Location Based Service (LBS)*

Teknologi *Location Based service (LBS)* merupakan salah satu bagian dari implementasi mobile GIS yang lebih cenderung memberikan fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial dibanding fungsionalitas pada teknologi GIS populer untuk Field Based GIS (Haryanto, 2013). Dua unsur utama LBS adalah:

1. *Location Manager (API Maps)*: Menyediakan *tools/source* untuk LBS, *Application Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta feature lainnya seperti tampilan satelit, street (jalan), maupun gabungannya. Paket ini berada pada *com.google.android.maps*.
2. *Location Providers (API Location)*: Menyediakan teknologi pencarian lokasi yang digunakan oleh *device/perangkat*. *API Location* berhubungan dengan data GPS (*Global Positioning System*) dan data lokasi *real-time*. *API Location* berada pada paket android yaitu *android.location*.

Dengan *location manager*, kita dapat menentukan lokasi kita saat ini dan rute menuju tempat tertentu.

E. *Google Maps*

Google Maps adalah layanan gratis *Google* yang cukup populer. Fitur *Google Maps* dalam web dapat ditambahkan sendiri dengan *Google Maps API*. (Udin, 2016) *Google Maps API* adalah library *JavaScript*. Menggunakan atau memprogram *Google Maps API* sangat mudah. Yang dibutuhkan adalah pengetahuan tentang *HTML* dan *JavaScript*, serta koneksi internet. Dengan menggunakan *Google Maps API*, dapat menghemat waktu dan biaya untuk membangun aplikasi peta digital yang handal, sehingga dapat fokus hanya pada data-data. Biarkan data peta-peta dunia menjadi urusan *Google* saja.

Saat ini versi terakhir *Google Maps API* adalah versi 3. Versi ini tampil lebih cepat dari versi sebelumnya khususnya untuk browser ponsel. Hal ini bisa membangun situs web yang dilengkapi peta untuk iPhone dan ponsel dengan sistem operasi Android.

Cara mulai menulis program *Google Maps API* dengan urutan sebagai berikut:

1. Memasukkan *Maps API JavaScript* ke dalam *HTML* kita.
2. Membuat element div dengan nama *map_canvas* untuk menampilkan peta.
3. Membuat beberapa objek literal untuk menyimpan *property-property* pada peta.
4. Menuliskan fungsi *JavaScript* untuk membuat objek peta.
5. Menginisiasi peta dalam *tag body HTML* dengan *event onload*.

F. UML (*Unified Modelling Language*)

UML (*Unified Modeling Language*) adalah sebuah bahasa yang berdasarkan grafik/gambar untuk memvisualisasi, menspesifikasikan, membangun, dan pendokumentasian dari sebuah sistem pengembangan software berbasis OOP (*Object-Oriented programming*) Eriza Siti (2010:23).

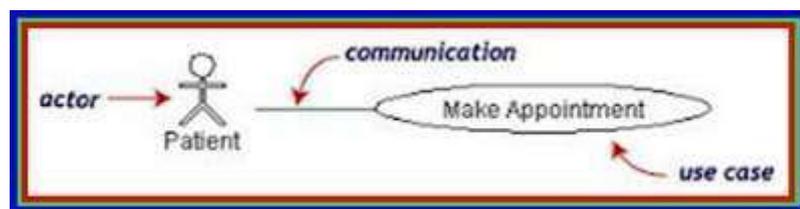
Beberapa diagram UML:

1. Diagram *Use Case*

Diagram *use case* terdiri atas tiga komponen, yaitu *use case*, aktor, dan relasi, yang menunjukkan interaksi antara *use case* dan aktor dalam suatu sistem. Pengertian aktor disini adalah seseorang atau apa saja yang berhubungan dengan sistem yang sedang dibangun. Adapun *use case* menggambarkan semua yang ada dalam ruang lingkup sistem dan bagaimana seseorang menggunakan sistem. Aktor dan *use case* dihubungkan dengan relasi.

Jenis relasi secara umum terbagi menjadi tiga, yakni :

- a. Relasi asosiasi, yakni relasi antara aktor dengan *use case*.
- b. Relasi *include*, dan *extend*, yakni relasi antar *use case*.
- c. Relasi generalisasi, yakni relasi antar aktor.



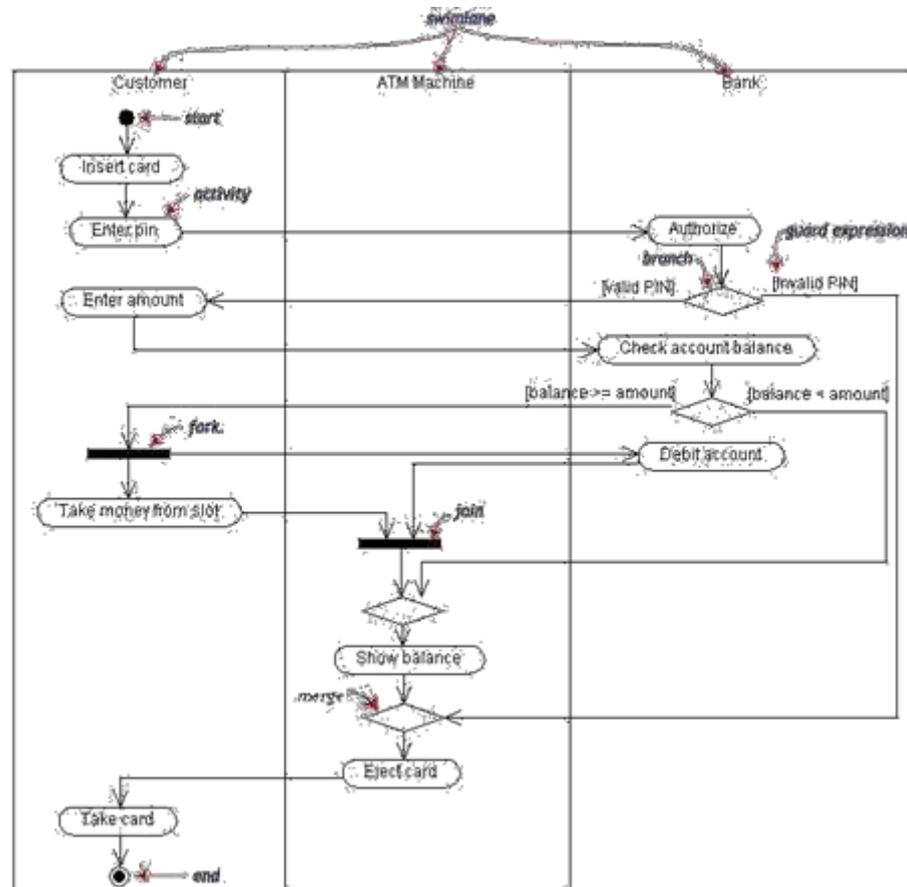
Gambar 2.1 Diagram *Use Case*

Diagram *Use Case* berguna dalam tiga hal:

- a. Menjelaskan fasilitas yang ada (*requirement*)
- b. Komunikasi dengan klien
- c. Membuat test dari kasus-kasus secara umum

2. Diagram *Activity*

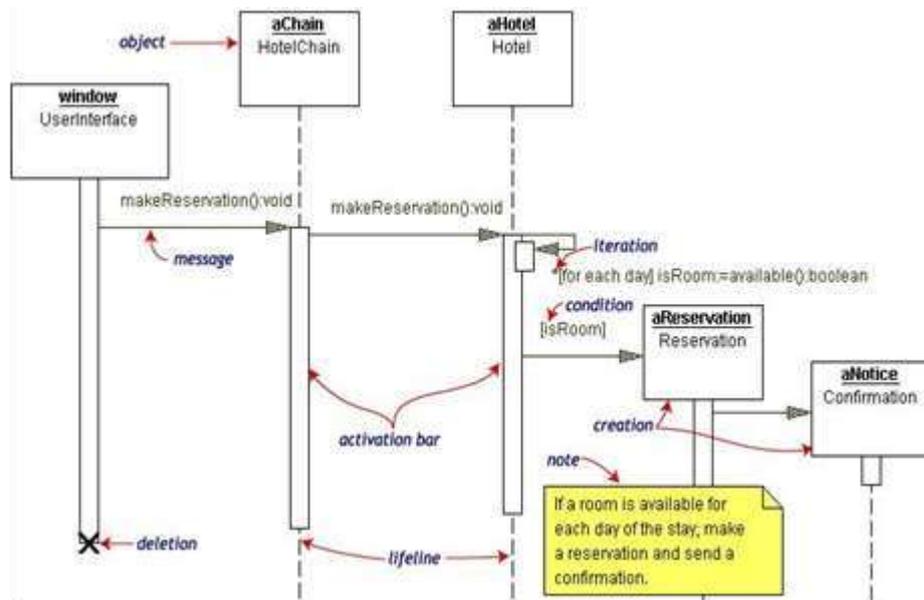
Pada dasarnya diagram *activity* sering digunakan oleh flowchart. Diagram ini berhubungan dengan diagram *statechart*. Diagram *statechart* berfokus pada obyek yang dalam suatu proses (atau proses menjadi suatu obyek), diagram *activity* berfokus pada aktifitas-aktifitas yang terjadi yang terkait dalam suatu proses tunggal. Jadi dengan kata lain, diagram ini menunjukkan bagaimana aktifitas-aktifitas tersebut bergantung satu sama lain.



Gambar 2.2 Diagram Activity

3. Diagram Sequence

Diagram *sequence* merupakan salah satu diagram *interaction* yang menjelaskan bagaimana suatu operasi itu dilakukan; *message* (pesan) apa yang dikirim dan kapan pelaksanaannya. Diagram ini diatur berdasarkan waktu. Obyek-obyek yang berkaitan dengan proses berjalannya operasi diurutkan dari kiri ke kanan berdasarkan waktu terjadinya dalam pesan yang terurut.



Gambar 2.3 Diagram Sequence

'Reservation window' mengirim pesan *makeReservation* ke 'HotelChain'. Kemudian 'HotelChain' mengirim pesan yang sama ke 'Hotel'. Bila 'Hotel' punya kamar kosong, maka dibuat 'Reservation' dan 'Confirmation'.

Lifeline adalah garis dot (putus-putus) vertikal pada gambar, menerangkan waktu terjadinya suatu obyek. Setiap panah yang ada adalah pemanggilan suatu pesan. Panah berasal dari pengirim ke bagian paling atas dari batang kegiatan (*activation bar*) dari suatu pesan pada *lifeline* penerima. *Activation bar* menerangkan lamanya suatu pesan diproses.

Pada gambar diagram, terlihat bahwa 'Hotel' telah melakukan pemanggilan diri sendiri untuk pemeriksaan jika ada kamar kosong. Bila benar, maka 'Hotel' membuat 'Reservation' dan 'Confirmation'.

Pemanggilan diri sendiri disebut dengan iterasi. *Expression* yang dikurung dengan “[]”, adalah condition (keadaan kondisi).

4. *Class Diagram*

Tujuan utama dari *class diagram* adalah untuk menciptakan sebuah kosa kata yang digunakan oleh analis dan pengguna. Diagram kelas biasanya merupakan hal-hal, ide-ide atau konsep yang terkandung dalam aplikasi. Misalnya, jika anda sedang membangun sebuah aplikasi penggajian, diagram kelas mungkin akan berisi kelas yang mewakili hal-hal seperti karyawan, cek, dan pendaftaran gaji. Diagram kelas juga akan menggambarkan hubungan antara kelas.

G. DBMS (*Database Management System*)

DBMS merupakan suatu perangkat lunak komputer yang dirancang secara khusus untuk memudahkan dalam pengelolaan database (Kadir: 2003). DBMS (*Database Management System*) merupakan perantara bagi pemakai dengan basis data dalam media penyimpanan. Bahasa basis data dibagi menjadi dua bentuk, yaitu:

1. *Data Definition Language (DDL)*

Jenis bahasa ini menspesifikasikan struktur/skema data yang menggambarkan/mewakili desain basis data secara keseluruhan. Dengan bahasa ini kita dapat membuat tabel baru, membuat indeks, mengubah tabel, dan lain-lain. Hasil kompilasi perintah DDL adalah kumpulan tabel yang disimpan dalam file khusus yang disebut kamus data.

Kamus data adalah suatu metadata (superdata) yaitu data yang mendeskripsikan data sesungguhnya.

2. *Data Manipulation Language* (DML)

Merupakan bentuk bahasa basis data yang berguna untuk melakukan manipulasi dan pengambilan data pada suatu basis data. *Data Manipulation Language* (DML) merupakan bahasa yang memudahkan pemakai dalam mengakses data yang dipresentasikan oleh model data.

DML terdiri dari dua jenis, yaitu:

3. Prosedural

Jenis ini mensyaratkan agar pemakai menentukan data apa yang diinginkan serta bagaimana cara mendapatkannya.

4. Non procedural

Jenis ini membuat pemakai dapat menentukan data apa yang diinginkan tanpa menyebutkan bagaimana cara mendapatkannya.

5. *Database Server* dengan MySQL

MySQL merupakan *software DBMS* yang sangat populer di kalangan pemrograman web, khususnya yang berbasis *open source*. Hal ini disebabkan karena kemudahannya untuk digunakan. Disamping dukungannya ke dalam berbagai *platform*, kecepatan aksesnya dapat diandalkan. MySQL mudah digunakan (*easy-touse*) sebagai sistem manajemen database relasional (RDBMS) yang digunakan untuk *database* pada *website*.

Selain itu, kemudahannya dalam integrasi ke berbagai aplikasi web dapat membantu dalam pengembangan sistem informasi online. MySQL mempunyai dua lisensi, yaitu open source dibawah GNU *General Public License* (GPL) dan *commercial* dibawah MySQLAB. Pada awalnya MySQL hanya dapat dioperasikan pada satu *platform* saja. Namun dalam perkembangannya, MySQL dapat dioperasikan dalam berbagai platform seperti Windows, *FreeBSD*, ataupun Linux (Sanjaya 2005).

H. Java

Java adalah bahasa berorientasi objek yang dapat digunakan untuk pengembangan aplikasi mandiri, (Indra Dwi 2012). Aplikasi berbasis *internet*, serta aplikasi untuk perangkat-perangkat cerdas yang dapat berkomunikasi lewat internet atau jaringan komunikasi. Dalam Java ada 2 (dua) jenis program berbeda, yaitu aplikasi dan *applet*. Aplikasi adalah program yang biasanya disimpan dan dieksekusi dari komputer lokal sedangkan *applet* adalah program yang biasanya disimpan pada komputer yang jauh, yang dikoneksikan pemakai lewat *web browser*. Java bukan turunan langsung dari bahasa pemrograman manapun. OOP (*object oriented programming*) adalah cara yang ampuh dalam pengorganisasian dan pengembangan perangkat lunak.

PHP adalah sebuah bahasa pemrograman yang berjalan dalam sebuah web-server (*server side*). PHP diciptakan oleh *programmer* unix dan Perl yang bernama Rasmus Lerdoft pada bulan Agustus-September 1994. Pada awalnya, Rasmus mencoba menciptakan sebuah *script* dalam *website* pribadinya dengan tujuan untuk memonitor siapa saja yang pernah mengunjungi *websitenya*.

Pada awalnya PHP merupakan kependekan dari *Personal Home Page* (Situs personal). Selanjutnya Rasmus merilis kode sumber tersebut untuk umum dan menamakannya PHP/FI pada sekitar tahun 1995 dan diperkenalkan kepada beberapa programmer pemula dengan alasan bahasa yang digunakan oleh PHP cukup sederhana dan mudah dipahami. Selanjutnya Rasmus menulis ulang PHP dengan bahasa C untuk meningkatkan kecepatan aksesnya.

Mulai bulan September sampai Oktober 1995, kode PHP ditulis ulang dan digabungkan menjadi PHP/FI. Baru di akhir tahun 1995 dirilis bagi umum secara gratis. Rasmus berangapan apabila kode PHP ini berguna bagi dirinya, tentu juga akan bermanfaat untuk orang lain. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman website oleh *browser*. Berdasarkan URL atau alamat *website* dalam jaringan internet, *browser* akan menemukan sebuah alamat dari *webserver*, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh *webserver*. Selanjutnya *webserver* akan mencari berkas yang diminta dan menampilkan isinya di *browser*. *Browser* yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke *web-server*, *web-server* akan memeriksa *type file* yang diminta user. Jika *type file* yang diminta adalah PHP, maka akan memeriksa isi *script* dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung *script* PHP, permintaan user akan langsung ditampilkan ke *browser*, namun jika dalam *file* tersebut mengandung *script* PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan *script-script* PHP dan mengolah *script* tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke *browser user* Andre (2014).

I. *Black Box Testing*

Black-box testing adalah metode pengujian perangkat lunak yang tes fungsionalitas dari aplikasi yang bertentangan dengan struktur internal. Pengetahuan khusus dari kode aplikasi / struktur internal dan pengetahuan pemrograman pada umumnya tidak diperlukan. Uji kasus dibangun di sekitar spesifikasi dan persyaratan, yakni, aplikasi apa yang seharusnya dilakukan. Menggunakan deskripsi eksternal perangkat lunak, termasuk spesifikasi, persyaratan, dan desain untuk menurunkan uji kasus. Tes ini dapat menjadi fungsional atau non-fungsional, meskipun biasanya fungsional. Perancang uji memilih input yang valid dan tidak valid dan menentukan output yang benar. Tidak ada pengetahuan tentang struktur internal benda uji itu. Lina, (2013)

Metode uji dapat diterapkan pada semua tingkat pengujian perangkat lunak: unit, integrasi, fungsional, sistem dan penerimaan. Ini biasanya terdiri dari kebanyakan jika tidak semua pengujian pada tingkat yang lebih tinggi, tetapi juga bisa mendominasi unit testing juga. Metode ujicoba *blackbox* memfokuskan pada keperluan fungsional dari *software*.

Uji coba *black box* berusaha untuk menemukan kesalahan dalam beberapa kategori, diantaranya:

1. Fungsi-fungsi yang salah atau hilang
2. Kesalahan *interface*
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan performa
5. Kesalahan inisialisasi dan terminasi