

BAB 3

LANDASAN TEORI

Pada bab ini akan dijelaskan mengenai teori dasar yang digunakan oleh penulis sebagai acuan dalam membangun aplikasi.

3.1. Nobar (Nonton Bareng)

Nobar sendiri merupakan istilah yang bisa digunakan untuk semua kegiatan yang bersifat menonton sesuatu bersama. Namun istilah ini lebih erat kaitannya dengan kegiatan menonton pertandingan sepakbola. Kegiatan ini bisa diadakan di mana saja dengan melibatkan banyak orang. Di Indonesia, nobar di lapangan lebih dekat dengan kegiatan menonton film melalui layar tancep, meskipun hal ini tidak disebut nobar. Istilah nobar juga diadaptasi oleh kegiatan-kegiatan lain yang melibatkan sekumpulan orang yang datang ke sebuah tempat umum untuk menonton sesuatu. Selain layar tancep, kegiatan nobar pada masa sekarang juga bisa dikaitkan dengan kegiatan menonton film di bioskop bersama-sama. Kegiatan nobar biasanya merupakan sebuah acara yang disusun dengan kepanitiaan, bukan acara yang diadakan secara mendadak. Intinya istilah nobar bisa dipakai untuk kegiatan apapun yang melibatkan orang banyak untuk menonton sesuatu (Febriani, 2011).

3.2. Aplikasi Mobile

Aplikasi *mobile* adalah sebuah sistem perangkat lunak yang memungkinkan setiap pemakai melakukan mobilitas dengan perlengkapan PDA-asisten digital

perusahaan pada telepon genggam atau seluler. Android dan iOS merupakan sistem operasi *mobile* yang untuk saat ini merajai pasaran. Aplikasi *mobile* juga dikenal sebagai *web app*, *online app*, *iPhone app* atau *smartphone app* (Khan, 2016).

Ada beberapa jenis aplikasi *mobile* pada platform android, yaitu *native application*, *web application*, dan *hybrid application*. Aplikasi *native* dibangun dan ditanam langsung didalam device dan dikembangkan dengan bahasa pemrograman Java dan SDK milik Android. Sedangkan *mobile Web Application* adalah aplikasi *mobile* yang dijalankan menggunakan *browser* pada device, dikembangkan menggunakan bahasa pemrograman web, seperti html5, php, atau ASP. *Hybrid Application* adalah penggabungan aplikasi *mobile* jenis *native* dengan aplikasi *mobile* jenis web dan dijadikan 1 paket kemudian diinstall pada device.

Dengan memperhitungkan kelebihan dan kekurangan dari jenis-jenis aplikasi *mobile* yang ada, penulis memilih untuk mengembangkan aplikasi *mobile native* dikarenakan penulis ingin membuat aplikasi yang berjalan di satu platform saja yaitu android dan aplikasi dapat di distribusikan di Google Play Store.

3.3. Sistem Layanan Berbasis Lokasi

Layanan berbasis lokasi merupakan suatu layanan yang menggunakan lokasi sebagai target nilai dari layanannya dimana target adalah sebuah entitas yang akan diketahui lokasinya. Poin pentingnya adalah ketika posisi target, di mana sebuah target bisa jadi adalah pengguna LBS itu sendiri atau entitas lain yang tergabung dalam suatu layanan (Kupper, 2005).

LBS mengarah pada aplikasi - aplikasi dan layanan-layanan yang menyediakan informasi lokasi mengenai suatu pengguna *mobile*, untuk menyediakan sebuah layanan atau informasi khusus pada pengguna. Untuk menentukan lokasi geografis dari pengguna diantaranya dengan menggunakan *Global Positioning System (GPS)*, yang mana adalah sistem yang berfungsi sebagai sistem navigasi global yang dapat menerima informasi dari sistem satelit. Satelit GPS ini memancarkan sinyal yang memungkinkan penerima sinyal GPS untuk mendapatkan informasi berupa lokasi penerima, arah, dan kecepatan (Wicaksono, 2008). Gambar 3.1. menggambarkan sistem GPS yang mempunyai 3 segmen utama, yaitu:

1. Segmen Kontrol

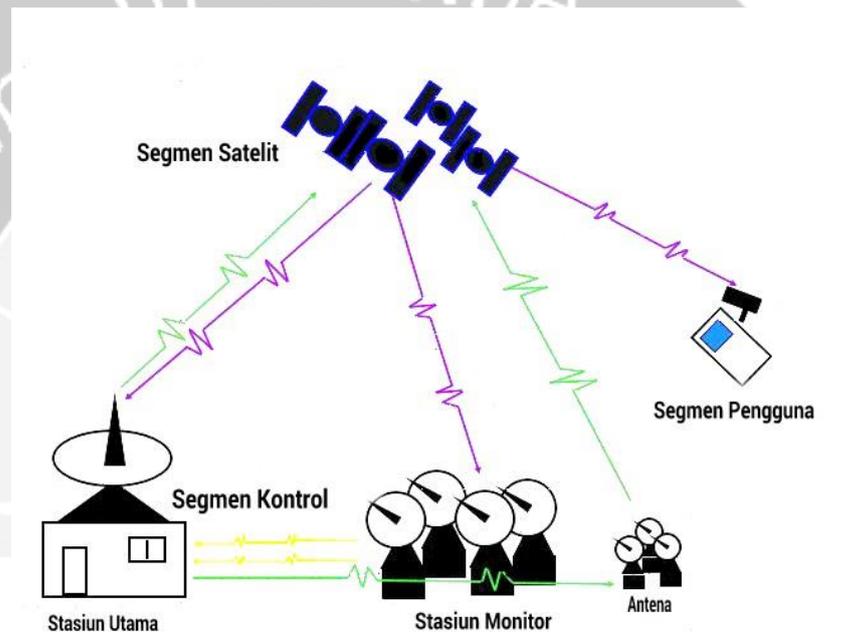
Segmen kontrol terdiri dari suatu sistem yang mengontrol kegiatan satelit GPS yang tersebar diseluruh dunia.

2. Segmen Satelit

Segmen satelit terdiri dari satelit-satelit GPS. Satelit bertugas untuk menerima dan menyimpan data yang ditransmisikan oleh segmen kontrol dan memancarkan data secara terus-menerus ke segmen pengguna. Konstelasi GPS terdiri dari 24 satelit yang menempati 6 buah bidang orbit yang berbentuk mendekati lingkaran, dengan masing-masing bidang orbit ditempati 4 buah satelit. Kedudukan masing masing satelit dalam tiap orbitnya diatur sedemikian rupa dengan jarak interval diantaranya tidak sama. Hal ini untuk meminimalkan pengaruh keadaan dimana ada satelit yang tidak berfungsi.

3. Segmen Pengguna

Segmen pengguna terdiri dari para pengguna yang tersebar diseluruh permukaan bumi dan alat penerima GPS. Segmen pengguna bertugas menerima data dari satelit dan memprosesnya untuk menentukan posisi, arah, jarak, dan waktu yang diperlukan.



Gambar 3.1. Ilustrasi Sistem Layanan Berbasis Lokasi

3.4. Metode Pencarian Lokasi

Ada beberapa metode pencarian lokasi yang umum digunakan, di mana masing-masing berbeda karena memiliki kelebihan dan kelemahan masing-masing tergantung penggunaannya. Sebagai contoh, metode yang satu bisa saja memerlukan biaya lebih besar dari yang lain namun memiliki tingkat akurasi yang tinggi. Adapun metode yang sering digunakan, (Deitel et al, 2002), yaitu:

1. *Device-based*, metode ini memerlukan perangkat tambahan berupa modul GPS yang didasarkan pada sistem satelit global.
2. *Network-based*, metode ini didasarkan pada sistem jaringan seluler. Ada beberapa jenis mekanisme pada metode ini, yakni:
 - a. *Cell of Origin*

Pada metode ini penentuan lokasi didasarkan pada *cell* jaringan di mana posisi ponsel pengguna berada, di mana masing-masing *cell* dijangkau oleh satu BTS (*Base Tower Station*). Informasi lokasi ditentukan oleh BTS terdekat yang menjangkau posisi ponsel user. Metode ini merupakan metode paling sederhana dan memiliki tingkat akurasi yang rendah.
 - b. *Angle of Arrival*

Metode ini memiliki tingkat akurasi yang lebih baik. Penentuan lokasi didasarkan pada sudut yang dibentuk antara posisi ponsel dengan dua atau lebih BTS yang menjangkaunya. *Angle of Arrival* menentukan informasi lokasi dengan triangulasi (pembentukan segitiga). Metode ini mudah terkena gangguan pada daerah perkotaan. Oleh karena itu baik digunakan pada daerah dengan populasi yang sedikit.
 - c. *Time Difference of Arrival*

Metode ini menentukan jarak dengan menggunakan triangulasi dari sinyal perjalanan. Informasi lokasi ditentukan dari perpotongan jarak busur lingkaran yang dibentuk atas perbedaan waktu kedatangan yang dihasilkan.

d. *Enhanced Observed Time Difference*

Pada metode ini, tiga BTS akan mengamati perbedaan sinyal waktu perjalanan ponsel user, dengan mentransmisikan sinyal ke ponsel. Metode ini menyerupai TDOA, namun membutuhkan perangkat tambahan pada ponsel.

e. *Location-Pattern Matching*

Pada metode ini, penentuan lokasi didasarkan pada kondisi daerah di mana ponsel user berada. Faktor penting yang digunakan adalah tekstur area yang didapat dari pemantulan gelombang radio ponsel terhadap bangunan atau penghalang yang dilewatinya. Pola pemantulan sinyal akan dianalisis oleh BTS untuk dicocokkan dengan pola area yang tersimpan di *database*. Metode ini tidak membutuhkan perangkat tambahan, namun membutuhkan pengelolaan *database* dan cocok digunakan di area perkotaan.

3. *Hybrid method*, metode ini mengkombinasikan metode GPS dengan *Network-based* untuk menghasilkan informasi yang cepat dan akurat.

3.5. Google Maps API

Google Map merupakan layanan aplikasi peta *online* yang disediakan oleh Google secara gratis. Layanan peta *Google Map* secara resmi dapat diakses melalui situs <http://maps.google.com>. Pada situs tersebut dapat dilihat informasi geografis pada hampir semua permukaan di bumi kecuali daerah kutub utara dan selatan. Layanan ini di buat sangat interaktif, karena di dalamnya peta dapat digeser sesuai keinginan pengguna, mengubah level *zoom*, serta mengubah tampilan jenis peta. *Google Map*

mempunyai banyak fasilitas yang dapat di pergunakan misalnya pencarian lokasi dengan memasukkan kata kunci, kata kunci yang dimaksud seperti nama tempat, kota atau jalan, fasilitas lainnya yaitu perhitungan rute perjalanan dari satu tempat, ke tempat lain (Anwar, et al., 2014).

3.6. Android

Android bukan hanya untuk perangkat *mobile* saja, android merupakan sebuah sistem operasi yang dikemas sedemikian rupa sehingga dapat digunakan untuk berbagai perangkat yang menggunakan layar (Simmonds, 2010). Android adalah sebuah sistem berbasis java yang berjalan pada Kernel Linux 2.6. Android dirilis oleh Google, dibawah Open Handset Alliance, pada November 2007. Bersamaan dengan peluncuran tersebut, Google membuat pusat Development Tool dan panduan untuk menjadi pengembang pada sistem tersebut. File panduan Software Development Kit (SDK), dan komunitas pengembang dapat diperoleh pada website resmi Google Android. Android menawarkan sebuah lingkungan yang berbeda untuk pengembang. Setiap aplikasi memiliki tingkatan yang sama. Android tidak membedakan antara aplikasi inti dengan aplikasi pihak ketiga. Application Programming Interface (API) yang disediakan menawarkan akses ke hardware, maupun data-data ponsel sekalipun, atau data sistem sendiri. Bahkan pengguna dapat menghapus aplikasi inti dan menggantikannya dengan aplikasi pihak ketiga (Munir, et al., 2015).

Gambar 3.2. menggambarkan arsitektur android yang terdiri dari beberapa lapisan, yaitu:

1. *Linux Kernel*

Android bukan Linux, tetapi android dibangun diatas Linux Kernel versi 2.6.

2. *Libraries*

Android menyertakan satu set *libraries* C atau C++ yang digunakan dalam berbagai komponen sistem android.

3. *Android Runtime*

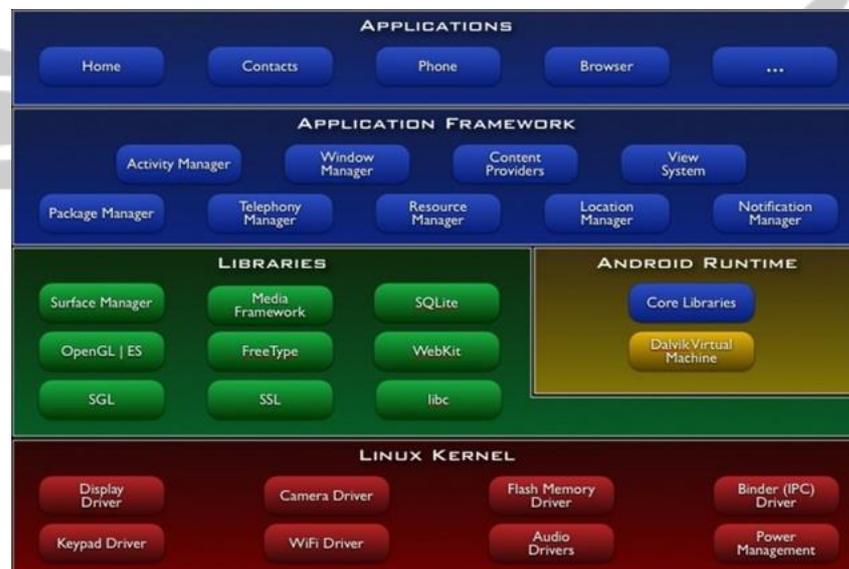
Android terdiri dari satu set perpustakaan inti (*core libraries*) yang menyediakan sebagian besar fungsi yang sama dengan yang terdapat dalam perpustakaan inti dari bahasa pemrograman java.

4. *Application Framework*

Arsitektur aplikasi dirancang agar komponen dapat digunakan kembali (*reuse*) dengan mudah.

5. *Application and Widget*

Pada lapisan ini *developer* menempatkan aplikasi yang dibuat.



Gambar 3.2. Ilustrasi Arsitektur Android (elinux.org)

3.7. Web Service

Web Service adalah sebuah teknologi yang bisa digunakan untuk membuat sebuah aplikasi yang bersifat *platform-independent*. *Web Service* dapat menghubungkan antar *web* yang berbeda, aplikasi yang berbeda, bahkan sampai *device* yang berbeda dalam sistem operasi yang berbeda pula (Hendrawan, 2006).

Web Service dapat menjalankan operasi-operasi termasuk akses data, *update database*, mengatur integritas data, mengirim *respon* dari *request* yang dikirim oleh *user*. *Web service* tidak memiliki tampilan karena *web service* termasuk dalam *Bussiness-Service tier*. Artinya di dalam *web service* hanya tersedia fungsi-fungsi yang nantinya dapat digunakan oleh aplikasi lainnya. Pada penggunaan pada sistem perancangan ini digunakan *web service* yang berupa file ber-ekstensi *.php* untuk dapat menjembatani antara aplikasi *mobile phone* dengan basis data. *Web service* diletakkan tempatnya sama dengan lokasi basis data. Penggunannya dengan menggunakan *HTTP Access* dimana hasil balikan dari *web service* adalah data dengan ekstensi *.json*.

3.8. PHP

PHP adalah bahasa *scripting language* yang dirancang secara khusus untuk penggunaan pada *web*. PHP kependekan dari *HyperText Preprocessor*. Seperti bahasa pemrograman *web* lainnya PHP memproses seluruh perintah yang berada dalam skrip PHP didalam *web browser* dan menampilkan *output*-nya ke dalam *web browser* klien. PHP adalah bahasa *scripting* yang menghasilkan *output* HTML ataupun *output* lain

sesuai keinginan program yang dijalankan pada *server side* (Suryatiningsih, 2010).

3.9. Firebase Cloud Messaging (FCM)

Sebagai salah satu produk Google yang kini banyak digunakan, Firebase menjadi *backend server* yang populer digunakan oleh para *developer*. Selain karena kemudahan dalam penggunaannya, Firebase juga menjadi solusi bagi *developer* yang ingin mengembangkan aplikasi tetapi memiliki masalah dalam pengadaan server untuk aplikasi.

Pengguna yang telah menggunakan Firebase akan menerima notifikasi tentang fitur baru yang Google terapkan dalam Firebase. Fitur baru tersebut diklaim dapat membantu *developer* dalam membangun aplikasi secara lebih cepat, dengan solusi penyimpanan file dan alat konfigurasi yang dapat di-*remote* (dipantau dari jauh) untuk melakukan kostumisasi terhadap aplikasi yang dikembangkan. Yang paling membahagiakan adalah adanya integrasi Firebase dengan *Google Cloud Messaging (GCM)* yang dikenal dengan *Firebase Cloud Messaging*. Dengan adanya fitur tersebut Google juga melakukan *update* terhadap *core authentication, database, dan hosting services* yang ada pada Firebase.

Tidak sampai disitu saja, kini Firebase juga mendukung pemantauan siklus penggunaan aplikasi dengan menambahkan fitur *Firebase Analytics*. Dengan adanya fitur ini *developer* dapat memantau penggunaan aplikasi dengan fitur analisis yang dapat digunakan secara gratis dan *unlimited*. Selain itu Google juga merilis *Test Lab* dan *Crash Reporting* yang dapat digunakan *developer* untuk meningkatkan kualitas aplikasi yang dikembangkan dengan Firebase. Fitur tambahan lainnya

yaitu adanya 5 dukungan untuk pengguna aplikasi yang mempermudah peningkatan aplikasi berbasis user yaitu *Notifications, Invites, Dynamic Links, App Indexing*, dan integrasi dengan *Google AdWords* (firebase.google.com).

3.10. Arsitektur MVC (*Model-View-Controller*)

Model-View-Controller (MVC) adalah sebuah konsep yang diperkenalkan oleh penemu Smalltalk (Trygve Reenskaug) untuk mengenkapsulasi data bersama dengan pemrosesan (*model*), mengisolasi dari proses manipulasi (*controller*) dan tampilan (*view*) untuk direpresentasikan pada sebuah *user interface*. MVC mengikuti pendekatan yang paling umum dari *Layering*. *Layering* hanyalah sebuah logika yang membagi kode ke dalam fungsi di kelas yang berbeda. Pendekatan ini mudah dikenal dan yang paling banyak diterima. Keuntungan utama dalam pendekatan ini adalah penggunaan ulang (*reusability*) kode. Definisi teknis dari arsitektur MVC dibagi menjadi tiga lapisan :

- 1) *Model*, digunakan untuk mengelola informasi dan memberitahu pengamat ketika ada perubahan informasi. Hanya *model* yang mengandung data dan fungsi yang berhubungan dengan pemrosesan data. Sebuah *model* meringkas lebih dari sekedar data dan fungsi yang beroperasi di dalamnya. Pendekatan *model* yang digunakan untuk komputer *model* atau abstraksi dari beberapa proses dunia nyata. Hal ini tidak hanya menangkap keadaan proses atau sistem, tetapi bagaimana sistem bekerja. Sebagai contoh, *programmer* dapat menentukan *model* yang menjembatani komputasi

back-end dengan *front-end* GUI (*Graphical User Interface*).

- 2) *View*, bertanggung jawab untuk pemetaan grafis ke sebuah perangkat. *View* biasanya memiliki hubungan 1-1 dengan sebuah permukaan layar dan tahu bagaimana untuk membuatnya. *View* melekat pada *model* dan *render* isinya ke permukaan layar. Selain itu, ketika *model* berubah, *view* secara otomatis menggambar ulang bagian layar yang terkena perubahan untuk menunjukkan perubahan tersebut. Terdapat kemungkinan beberapa *view* pada *model* yang sama dan masing-masing *view* tersebut dapat *render* isi *model* untuk permukaan tampilan yang berbeda.
- 3) *Controller*, menerima input dari pengguna dan menginstruksikan *model* dan *view* untuk melakukan aksi berdasarkan masukan tersebut. Sehingga, *controller* bertanggung jawab untuk pemetaan aksi pengguna akhir terhadap respon aplikasi. Sebagai contoh, ketika pengguna mengklik tombol atau memilih *item* menu, *controller* bertanggung jawab untuk menentukan bagaimana aplikasi seharusnya merespon (Hidayat & Surarso, 2012).

Sekian pembahasan hal-hal yang mendasari dibuatnya aplikasi Ayo Nobar, arsitektur, bahasa pemrograman dan *tools* yang digunakan dalam pembuatan aplikasi Ayo Nobar. Pada bab selanjutnya akan dibahas mengenai analisis dan perancangan aplikasi Ayo Nobar.