

BAB II

TEORI PENUNJANG

2.1 Sistem Operasi Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* berbasis linux yang mencakup sistem operasi, *middleware*, dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi. Android dipuji sebagai *platform* mobile pertama yang lengkap, terbuka, dan bebas[1].

Sejauh ini Android telah melalui cukup banyak pembaruan sejak pertama dirilis. Tabel 2.1 menunjukkan berbagai versi Android beserta dengan tanggal rilis, nama kode dan level API (*Application Programming Interface*).

Tabel 2.1 Versi Android

Versi Android	Tanggal Rilis	Nama Kode	Level API
1.0	23 September 2008	-	1
1.1	9 Februari 2009	-	2
1.5	30 April 2009	Cupcake	3
1.6	15 September 2009	Donut	4
2.0/2.0.1/2.1	26 Oktober 2009	Éclair	5/6/7
2.2	20 Mei 2010	Froyo	8
2.3/2.3.2/2.3.3/2.3.7	6 Desember 2010	Gingerbread	9/10
3.0/3.1/3.2	22 Februari 2011	Honeycomb	11/12/13
4.0/4.0.2/4.0.3/4.0.4	19 Oktober 2011	Ice Cream Sandwich	14/15
4.1/4.2/4.3	9 Juli 2012	Jelly Bean	16/17/18
4.4	31 Oktober 2013	Kitkat	19
5.0	3 November 2014	Lollipop	21
6.0	September 2015	Marshmallow	23
7.0	23 Agustus 2016	Nougat	24
8.0	21 Maret 2017	Oreo	26

Level API menyatakan suatu bilangan unik yang digunakan untuk mengidentifikasi *application programming interface* (API) yang digunakan pada suatu versi Android. Dengan perkataan lain, setiap versi Android ditandai dengan sebuah level API. Peranti yang digunakan untuk mengembangkan aplikasi Android mencakup *Java Development Kit* (JDK), Android SDK dan Android Studio.

2.1.1 Java Development Kit

Java Development Kit (JDK) merupakan perangkat pengembangan aplikasi java. JDK mutlak diperlukan untuk membuat aplikasi Android, dimana kita ketahui aplikasi Android itu berbasis Java. Java juga merupakan suatu bahasa pemrograman yang biasa digunakan untuk membuat aplikasi. Namun tidak semua pustaka dalam Java digunakan di Android. sebagai contohnya android tidak menggunakan *swing*[2].

2.1.2 Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk android *development* yang diperkenalkan Google pada acara Google I/O 2013. Sebelumnya Eclipse lebih dahulu diperkenalkan. Namun seiring dengan adanya *Android Studio* maka dilakukan pengembangan dari Eclipse IDE yang dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. *Android Studio* berupa Eclipse, *Android Studio* memiliki lebih banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, *Android Studio* menggunakan *Gradle* sebagai *build environment*. Fitur fitur lainnya adalah sebagai berikut[3]:

- a. Menggunakan *Gradle-based build system* yang fleksibel.
- b. Bisa mem-*build multiple* APK (*Application Package*).
- c. *Template support* untuk *Google Service* dan berbagai macam tipe perangkat.
- d. *Layout editor* yang lebih bagus.
- e. *Build-in support* untuk *Google Cloud Platform*, sehingga mudah untuk integrasi dengan *Google Cloud Messaging* dan *App Engine*.
- f. *Import library* langsung dari *Maven repository*.
- g. *Emulator* yang cepet dan kaya fitur.
- h. Mendukung C++, NDK Kotlin.

2.1.3 Android SDK

Android SDK adalah kumpulan *Software* yang berisi mengenai pustaka, *debugger* (alat pencari kesalahan program), *emulator* (peniru perangkat bergerak), dokumentasi, kode contoh, dan panduan[4]. Keberadaan *emulator* dapat membuat dan menguji aplikasi Android tanpa harus mempunyai perangkat keras berbasis Android.

2.2 Pemrograman Berorientasi Objek

Pemrograman Berorientasi Objek (*Object Oriented Programming*) merupakan suatu paradigma pemrograman yang berbeda dengan pemrograman procedural. Pemrograman Berbasis Objek (disingkat dengan nama PBO) memiliki fokus pada konsep pemrograman objek yang memiliki atribut dan fungsi yang yang digunakan untuk melakukan manipulasi terhadap atribut tersebut.

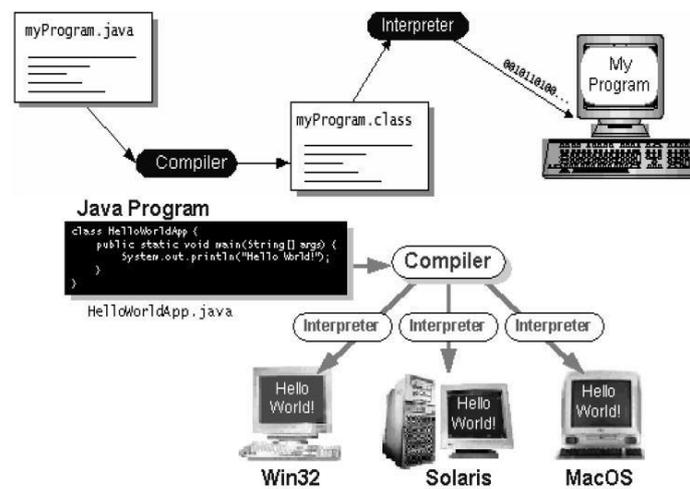
Spesifiknya, pengertian berorientasi objek itu sendiri berarti mengorganisasi perangkat lunak sebagai kumpulan objek tertentu yang memiliki struktur data dan perilakunya[5]. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berubung secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu:

1. Identifikasi
2. Klasifikasi
3. *Polymorphism* (polimorfisme)
4. *Inheritance* (pewarisan)

2.3 Pemrograman Java

Java adalah program yang sangat populer di lingkungan pendidikan saat ini, hampir di seluruh kampus memiliki satuan ajar pemrograman ini. Hal yang paling mendasar dan harus dikuasai oleh Programmer Java adalah memasang *library* (kelas dalam bentuk biner). Karena java yang dirancang secara orientasi objek, tentunya program yang terdiri atas kesatuan antar objek yang membentuk aplikasi. Sehingga sebuah objek tertentu bisa digunakan dalam beberapa aplikasi yang berbeda. Objek

ini biasanya bersifat umum penggunaannya, misalkan objek kalender, objek *driver* basis data, objek report dan lain-lain[6].



Gambar 2.1 Cara Kerja Pemrograman Java

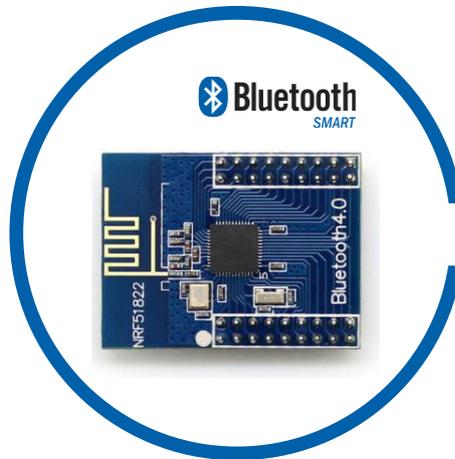
2.4 *Bluetooth Low Energy (BLE)*

Berdasarkan *Bluetooth Low Energy (BLE)* adalah teknologi nirkabel yang muncul dan dikembangkan oleh *Bluetooth Special Interest Group (SIG)* yang berguna untuk komunikasi jarak pendek. Berbeda dengan versi *Bluetooth* sebelumnya, BLE telah dirancang sebagai solusi *Bluetooth* berdaya rendah untuk aplikasi kontrol dan pemantauan.

Perangkat *Bluetooth Low Energy* bekerja pada frekuensi 2,4 GHz, sehingga memiliki karakteristik propagasi dalam ruangan yang sama dengan transceiver WiFi 2,4 GHz. Mode beaconing, atau advertising, yang diizinkan dalam standar BLE dapat mengirim pesan yang sangat singkat secara broadcast dengan kecepatan update yang sangat fleksibel. Pesan ini dapat digunakan untuk memungkinkan perangkat mendeteksi kedekatan dengan lokasi tertentu berdasarkan Kekuatan Sinyal yang Diterima (RSS). Dengan cara ini, lokasi spesifik, iklan, *voucher* dan informasi dapat diberikan kepada pengguna dengan cepat dan mudah.

Berbeda dengan Bluetooth biasa, *Bluetooth Low Energy* memiliki kelebihan pada konsumsi dayanya yang rendah, Meskipun kelemahannya yang sulit untuk melakukan komunikasi data yang besar, namun sangat berguna apabila digunakan untuk bertukar data kecil secara berkala[7]. Berbeda dengan Bluetooth pada

umumnya yang dapat menangani banyak data tetapi menghabiskan banyak daya dengan cepat dan juga biaya yang lebih banyak. *Bluetooth Low Energy* digunakan untuk aplikasi yang tidak perlu bertukar data dalam jumlah besar, karena itu dapat berjalan dengan daya baterai yang tahan lama dan biaya yang lebih murah[8].



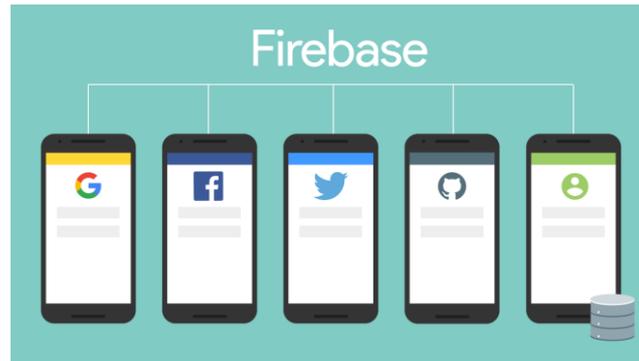
Gambar 2.2 *Bluetooth Low Energy*

2.5 Firebase

Pengertian Firebase pertama kali didirikan pada tahun 2011 oleh Andrew Lee dan James Tamplin. Produk yang pertama kali dikembangkan adalah *Realtime Database*, di mana *developer* dapat menyimpan dan melakukan sinkronisasi data ke banyak *user*. Kemudian berkembang menjadi layanan penyedia pengembangan aplikasi. Pada Oktober 2014, perusahaan tersebut diakuisisi oleh Google. Berbagai fitur terus dikembangkan hingga diperkenalkan pada Mei 2016 di Google I/O.

Firebase merupakan teknologi yang memungkinkan untuk membuat aplikasi web tanpa pemrograman pada *server* sehingga pembangunannya lebih mudah. Dengan menggunakan Firebase, kita tidak perlu menekan *server over-provisioning* atau membangun REST API namun dengan sedikit konfigurasi dapat melakukan fungsi memverifikasi pengguna, penyimpanan data, dan menerapkan aturan akses[9].

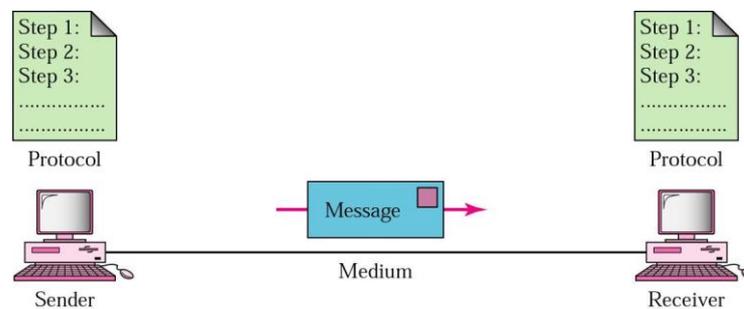
Firebase dapat digunakan pada sistem operasi seperti iOS, web, OS X, dan android. Aplikasi yang menggunakan Firebase dapat mengontrol dan menggunakan data tanpa perlu memikirkan bagaimana data akan di simpan, dan di sinkronkan di berbagai contoh aplikasi secara *real time*.



Gambar 2.3 Firebase

2.6 Komunikasi Data

Komunikasi data adalah hubungan atau interaksi (pengiriman dan penerimaan) antar *device* yang terhubung dalam sebuah jaringan, baik yang dengan jangkauan sempit maupun dengan jangkauan yang lebih luas. Efektivitas dari komunikasi data tergantung pada empat karakteristik mendasar, yaitu pengiriman, akurasi, ketepatan waktu, dan jitter [10].



Gambar 2.4 Ilustrasi Komunikasi

Komponen komunikasi data meliputi:

1. Pengirim, perangkat yang mengirimkan data.
2. Penerima, perangkat yang menerima data.
3. Data, informasi yang akan dikondisikan.
4. Media pengiriman, media atau perantara yang digunakan untuk melakukan pengiriman data.
5. Protokol, aturan-aturan yang berfungsi sebagai penyelarasan hubungan.

2.7 iBeacon

iBeacon merupakan perangkat yang digunakan untuk penentuan posisi dalam ruangan, yang dikeluarkan oleh Apple Corp pada bulan September 2013[11].

Mekanisme kerja dari iBeacon adalah perangkat komunikasi yang dilengkapi dengan *Bluetooth Low Energy* (BLE) mengirimkan data berupa sinyal menggunakan *Id* BLE, dan setelah menerima *Id* tersebut, perangkat terhubung dengan Internet dan akan melakukan tindakan sesuai dengan data yang dikirim[12].

Penyebaran iBeacon terdiri dari satu sampai beberapa perangkat iBeacon yang mentransmisikan nomor identifikasi unik mereka ke area lokal. Perangkat lunak pada perangkat penerima kemudian dapat mencari iBeacon dan melakukan berbagai fungsi, seperti memberi tahu pengguna. Penerima perangkat juga dapat terhubung ke iBeacons untuk mengambil data dari layanan iBeacon GATT (generic attribute profile)[13]. iBeacons tidak mendorong pemberitahuan ke perangkat penerima (selain identitas mereka sendiri). Namun, perangkat lunak mobile dapat menggunakan sinyal yang diterima dari iBeacons untuk memicu notifikasi push mereka sendiri.



Gambar 2.5 Module iBeacon