

## BAB III

### LANDASAN TEORI

Bab ini akan membahas mengenai landasan teori tentang futsal, aplikasi mobile, android, *location based service*, *google maps*, *Application Programming Interface*, dan metode *Breadth First Search*.

#### 3.1. Futsal

Futsal adalah permainan bola yang dimainkan oleh dua tim, yang masing-masing beranggotakan lima orang. Tujuannya adalah memasukkan bola ke gawang lawan, dengan memanipulasi bola dengan kaki. Selain lima pemain utama, setiap regu juga diizinkan memiliki pemain cadangan. Tidak seperti permainan sepak bola dalam ruangan lainnya, lapangan futsal dibatasi garis, bukan net atau papan. Futsal diciptakan di Montevideo, Uruguay pada tahun 1930, oleh Juan Carlos Ceriani. Keunikan futsal mendapat perhatian di seluruh Amerika Selatan, terutamanya di Brazil. Ketrampilan yang dikembangkan dalam permainan ini dapat dilihat dalam gaya terkenal dunia yang diperlihatkan pemain-pemain Brasil di luar ruangan, pada lapangan berukuran biasa. Pele, bintang terkenal Brazil, contohnya, mengembangkan bakatnya di futsal. (Zagalaz, et al., 2012)

#### 3.2. Aplikasi Mobile

Aplikasi *mobile* adalah proses pengembangan aplikasi untuk perangkat genggam seperti PDA, asisten digital perusahaan atau telepon genggam. Aplikasi ini sudah ada pada telepon selama manufaktur, atau di *download* oleh pelanggan dari toko aplikasi dan dari distribusi

perangkat lunak *mobile platform* yang lain (Safaat, 2012). Aplikasi *mobile* memiliki tiga jenis aplikasi, yakni aplikasi *native*, aplikasi *hybrid*, aplikasi *web*.

Aplikasi *native* adalah aplikasi yang dibangun dengan bahasa pemrograman yang spesifik untuk *platform* tertentu. Contoh populernya yakni penggunaan bahasa pemrograman *Objective-C* atau *Swift* untuk *platform* iOS (*Apple*). Adapun *platform* Android yang menggunakan bahasa pemrograman *Java*. Aplikasi *hybrid* adalah aplikasi *web* yang ditransformasikan menjadi kode *native* pada *platform* seperti iOS atau Android. Aplikasi *hybrid* biasanya menggunakan *browser* untuk mengizinkan aplikasi *web* mengakses berbagai fitur di *device mobile* seperti *Push Notification*, *Contacts*, atau *Offline Data Storage*. Beberapa *tools* untuk mengembangkan aplikasi *hybrid* antara lain *Phonegap*, *Rubymotion* dan lain-lain. Aplikasi *mobile* yang dijalankan menggunakan *browser* yang ada di handphone, menggunakan bahasa pemrograman *web* seperti *PHP* & *HTML5*. Aplikasi *web* bisa menjadi pilihan yang terbaik untuk membangun produk yang diinginkan, karena benar-benar minim persyaratan, dengan kata lain akses ke fitur dari perangkat *mobile* seperti *Push Notification* tidak diperlukan. Aplikasi *web* bisa menjadi pilihan yang murah dari sisi *budget*. Namun sayangnya, aplikasi *web* tidak bisa didistribusikan melalui toko aplikasi *native* seperti *App Store* atau *Google Play*. Untuk aplikasi pencarian pertandingan berbasis *mobile* yang akan penulis bangun menggunakan jenis aplikasi *native*, karena aplikasi yang akan dibangun tidak menggunakan bahasa pemrograman *website* dan tidak menggunakan *browser* yang

ada di perangkat *mobile*, dan akan murni menggunakan bahasa pemrograman java.

### **3.3. Android**

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat *mobile* layar sentuh seperti *smartphone* dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya *Open Handset Alliance*, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Android menyediakan *platform* yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Dengan diberlakukannya *platform* terbuka, lisensi perijinan Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pengembang, operator nirkabel, dan pembuat perangkat. Android memungkinkan penggunaanya untuk memasang aplikasi pihak ketiga yang dapat diperoleh dari toko aplikasi seperti Google *Play*, Amazon, ataupun dengan mengunduh dan memasang berkas *Application Package File* (APK) dari situs pihak ketiga. Sistem operasi ini biasanya ditanamkan pada *smartphone* atau komputer tablet. (Suprianto & Agustina, 2012)

### **3.4. Location Based Service**

*Location based services* (LBS) adalah layanan berbasis lokasi atau istilah umum yang sering digunakan untuk menggambarkan teknologi yang digunakan untuk

menemukan lokasi perangkat yang pengguna gunakan. Layanan ini menggunakan teknologi *global positioning service* (GPS) dan *cell-based location* dari Google. Selain itu, LBS tersebut terdiri dari beberapa komponen, diantaranya *mobile devices*, *communication network*, *position component*, dan *service and content provider*. *Mobile devices* merupakan komponen yang sangat penting. Piranti *mobile* tersebut diantaranya adalah *smartphone*, PDA, dan lainnya yang dapat berfungsi sebagai alat navigasi atau seperti halnya alat navigasi berbasis GPS. Komponen *communication network* ini berupa jaringan telekomunikasi bergerak yang memindahkan data pengguna dari perangkat ke penyedia layanan. *Position component* yang dimaksud adalah posisi pengguna harus ditentukan. Posisi ini dapat didapatkan dengan jaringan telekomunikasi atau dengan GPS. Sedangkan *service and content provider* adalah penyedia layanan yang menyediakan layanan berbeda ke pengguna seperti pencarian rute, kalkulasi posisi, dan lainnya.

### **3.5. Google Maps**

*Google Maps* adalah layanan gratis yang diberikan oleh Google dan sangat populer. *Google Maps* adalah suatu peta dunia yang dapat digunakan untuk melihat suatu daerah. Dengan kata lain, *Google Maps* merupakan suatu peta yang dapat dilihat dengan menggunakan suatu *browser* atau aplikasi *mobile*. Pengembang dapat menambahkan fitur *Google Maps* dalam *web* dan aplikasi *mobile* yang sedang dikembangkan atau pada blog yang berbayar maupun gratis sekalipun dengan *Google Maps API*. *Google Maps API* adalah suatu *library* yang berbentuk *JavaScript* (Mufti, 2015). *Google Maps* dibuat dengan menggunakan

kombinasi dari gambar peta, basis data, serta objek-objek interaktif yang dibuat dengan bahasa pemrograman *HTML*, *Javascript*, dan *AJAX*, serta beberapa bahasa pemrograman lainnya. Gambar-gambar peta yang muncul pada layar merupakan hasil komunikasi dari pengguna dengan *database* pada *web server* google untuk menampilkan gabungan dari potongan-potongan gambar yang diminta. Seluruh citra yang ada diintegrasikan ke dalam suatu *database* pada *google server*, yang nantinya akan dipanggil sesuai kebutuhan permintaan. Bagian-bagian gambar peta yang merupakan gabungan dari gambar-gambar yang berukuran 256 x 256 pixel. Tiap-tiap 256 x 256 *tile* mewakili gambar tertentu dalam *longitude*, *latitude*, dan *zoom level* tertentu. (Ichtiara, 2008)

### **3.6. Application Programming Interface**

*Application Programming Interface* (API) adalah sekumpulan perintah, fungsi, komponen, dan protokol yang disediakan oleh sistem operasi ataupun bahasa pemrograman tertentu yang dapat digunakan oleh pengembang aplikasi saat membangun perangkat lunak. Dalam API terdapat fungsi-fungsi atau perintah-perintah untuk menggantikan bahasa yang digunakan dalam *system calls* dengan bahasa yang lebih terstruktur dan mudah dimengerti oleh pengembang aplikasi. API yang digunakan dalam aplikasi ini adalah *Google Maps*. *Maps* yang disediakan dari google akan digunakan untuk menerapkan lokasi-lokasi futsal yang akan digunakan di aplikasi ini. Dengan disediakannya API oleh *Google Maps*, aplikasi ini akan terintegrasi dengan baik antara pengguna aplikasi (Chow, 2008).

Keuntungan menggunakan API :

1. Probabilitas

API dapat digunakan untuk bahasa pemrograman ataupun untuk sistem operasi mana saja asalkan paket-paket API sudah terpasang.

2. Lebih Mudah Dimengerti

API menggunakan bahasa yang lebih terstruktur dan mudah dimengerti daripada bahasa *system call*. Hal ini sangat penting dalam hal editing dan pengembangan.

3. Mudah Dikembangkan

Dengan adanya API, memudahkan para pengembang untuk mengembangkan suatu sistem.

Ada tiga jenis bahasa pemrograman API :

1. Inti resmi Java API, yang terdapat dalam JDK atau JRE, dari salah satu edisi dari Java Platform. Tiga edisi dari Java Platform adalah Java ME (Micro edition), Java SE (*Standard edition*), dan Java EE (*Enterprise edition*).

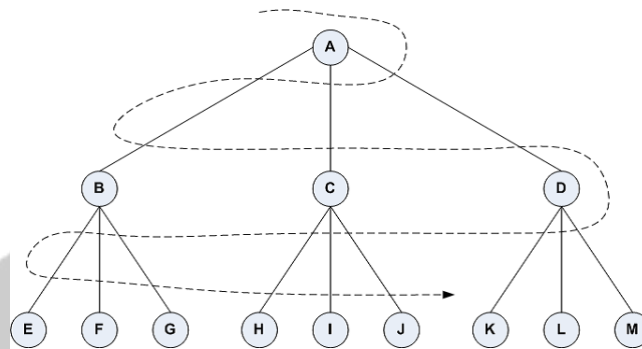
2. Resmi opsional API yang dapat di *download* secara terpisah. Spesifikasi API ini di definisikan sesuai dengan Spesifikasi Java Request (JSR).

3. API tidak resmi, yang dikembangkan oleh pihak ketiga, tetapi tidak berkaitan dengan JSR apapun.

### 3.7. Metode *Breadth First Search*

Algoritma *Breadth First Search* (BFS) atau dikenal juga dengan nama algoritma pencarian melebar adalah algoritma yang melakukan pencarian secara melebar yang mengunjungi simpul secara *preorder* yaitu mengunjungi suatu simpul kemudian mengunjungi semua simpul yang bertetangga dengan simpul tersebut terlebih dahulu. Selanjutnya, simpul yang belum dikunjungi dan bertetangga dengan simpul-simpul yang tadi dikunjungi, demikian seterusnya. Jika graf berbentuk pohon berakar, maka semua simpul pada aras  $d$  dikunjungi lebih dahulu sebelum simpul-simpul pada aras  $d+1$ .

Algoritma ini memerlukan sebuah antrian  $q$  untuk menyimpan simpul yang telah dikunjungi. Simpul-simpul ini diperlukan sebagai acuan untuk mengunjungi simpul-simpul yang bertetangganya. Tiap simpul yang telah dikunjungi masuk ke dalam antrian hanya satu kali. Algoritma ini juga membutuhkan table Boolean untuk menyimpan simpul yang telah dikunjungi sehingga tidak ada simpul yang dikunjungi lebih dari satu kali. *Breadth-first search (BFS)* melakukan proses *searching* pada semua node yang berada pada level atau hirarki yang sama terlebih dahulu sebelum melanjutkan proses *searching* pada node di level berikutnya. Urutan proses *searching* BFS ditunjukkan dalam Gambar 1.1 adalah: A, B, C, D, E, F, G, H, I, J, K, L, M



Gambar 1.1. Diagram pohon dari BFS.

Dengan menggunakan algoritma *BFS*, pencarian pertandingan *squad* memiliki keuntungan karena algoritma *BFS* jauh mengungguli algoritma yang lainnya. Hal ini dikarenakan algoritma *BFS* tidak akan terjebak dalam rute simpul yang salah, dan akan selalu mendapatkan solusi terbaik. Algoritma ini cocok untuk diterapkan jika simpul solusi berada dalam jarak yang tidak terlalu jauh dari simpul awal. Jika dibandingkan dengan algoritma lainnya, algoritma *BFS* lebih unggul karena jika simpul solusi berada di dekat simpul akar, maka solusi akan ditemukan sebelum algoritma tersebut membentuk pohon yang lebar. Kekurangan dari algoritma *BFS* adalah pemakaian memori yang lebih besar dibandingkan dengan algoritma yang lainnya, karena *BFS* selalu membandingkan simpul-simpul yang bertetangga, sehingga jika simpul solusi jauh dari simpul akar, maka memori yang digunakan lebih besar dibandingkan algoritma lainnya yang menggunakan memori yang lebih kecil.

Demikianlah pembahasan mengenai landasan teori tentang futsal, aplikasi mobile, android, *location based service*, *google maps*, *Application Programming Interface*, dan metode *Breadth First Search*.