

## **BAB 3**

### **LANDASAN TEORI**

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

#### **3.1. Perpustakaan**

Perpustakaan adalah sebuah gedung atau akomodasi fisik tempat menyimpan buku dan media non-buku, digital maupun analog. Perpustakaan sebagai akumulasi buku dalam arti luas serta forum yang merupakan titik temu antara pemakai informasi dengan pustakawan sebagai sumber yang menyediakan jasa temu balik yang efisien dan efektif (Cahyaningtyas & Iriyani, 2015).

#### **3.2. Online Public Access Catalog (OPAC)**

*OPAC* merupakan sebuah sistem katalog perpustakaan yang menggunakan komputer. Basis datanya biasanya dirancang dan dibuat sendiri oleh perpustakaan dengan menggunakan perangkat lunak komersial atau buatan sendiri. Katalog ini memberikan informasi bibliografis dan dimana letak koleksinya. Katalog ini telah dirancang untuk mempermudah pengguna sehingga pengguna tidak perlu bertanya dalam menggunakannya (*user friendly*). Sebelum teknologi informasi masuk dalam dunia perpustakaan, katalog yang dikenal hanya dalam bentuk kartu atau lembaran kertas. Sekarang katalog tidak saja dibuat dalam bentuk kartu, tetapi juga dalam bentuk digital. Katalog dalam bentuk digital biasanya disimpan dalam *harddisk* komputer atau media penyimpanan lainnya, seperti disket, *CD-ROM*, dan *DVD*.

*OPAC* bekerja berdasarkan konsep jaringan, baik berupa LAN maupun WAN. LAN digunakan untuk keperluan hubungan kerja dalam satu ruangan atau bangunan, sedangkan WAN untuk keperluan kerja dalam lingkup yang lebih luas, yaitu antarwilayah. Dengan adanya *OPAC* yang terkomunikasi melalui internet, jangkauan pengguna perpustakaan menjadi lebih luas. Hal tersebut menjadikan nilai guna informasi jauh lebih tinggi (Fitriastuti, 2009).

### 3.3. Android

Android adalah *software* untuk perangkat *mobile* yang mencakup sistem operasi, *middleware* dan aplikasi kunci. Pengembangan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Serangkaian aplikasi inti Android antara lain email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses informasi lokasi, menjalankan *background services*, mengatur alarm, tambahkan pemberitahuan ke status bar, dan banyak lagi. Android bergantung pada versi Linux 2.6 untuk layanan sistem inti seperti keamanan, manajemen memori, manajemen proses, *network stack*, dan *model driver*. Kernel juga bertindak sebagai lapisan abstraksi antara *hardware* dan seluruh *software stack* (Fadjar Efendy Rasjid, 2010).

### 3.4. Aplikasi Mobile

Aplikasi *mobile* adalah sebuah sistem perangkat lunak yang memungkinkan setiap pemakai melakukan mobilitas pada telepon genggam atau seluler. Android dan iOS merupakan sistem operasi *mobile* yang untuk saat ini sangat populer. Aplikasi *mobile* merupakan peningkatan dari sistem perangkat lunak terpadu yang ada pada *PC Desktop*. Aplikasi *mobile* itu sendiri dibagi menjadi 3 jenis yaitu :

#### 1. Native App

Native adalah sebuah aplikasi yang ditujukan pada perangkat *mobile* Android, iOS, Symbian, Windows Phone, WebOS dan lainnya. Aplikasi ini dirancang dengan menggunakan bahasa pemrograman seperti Objective-C atau Swift dan Java, kemudian juga alat *Software Development Kit* (SDK) yang ditanam (*install*) langsung kedalam perangkat *mobile*. Aplikasi *Native* bisa berfungsi baik untuk anda yang menginginkan penggunaan GPS, kamera, mikrofon, kompas, daftar kontak, swipe gestures (layar sensor) dan sebagainya dimana juga bisa aktif ketika koneksi internet tiada karena

memiliki sistem notifikasi pada platform. Aplikasi ini bisa anda dapati di Google Play atau App Store.

## 2. *Mobile Web App*

Aplikasi ini sedikit berbeda dengan Native apps, yaitu pada bahasa pemrograman dan hanya dapat diakses ketika koneksi internet aktif. Aplikasi ini juga tidak tersedia di Google Play dan App Store. Bahasa pemrograman yang digunakan pada aplikasi *mobile website* adalah html5, javascript, dan *Active Server Pages* (ASP) ditambahkan komponen seperti php, *database*, CSS (*Cascading Style Sheets*) ataupun lain sebagainya.

## 3. *Hybrid App*

Aplikasi Hybrid adalah sebuah aplikasi yang juga populer pada perkembangan *mobile*. Komponen aplikasi ini terdiri dari kedua perangkat aplikasi sistem yang telah disebutkan sebelumnya yaitu *Native app* dan *Mobile Web App*, contohnya html5, css, javascript dan php beserta alat lain seperti Apache Cordova, Rubymotion, Appcelerator Titanium yang bakal dibungkus menjadi satu aplikasi *mobile* bernama Hybrid (Khan, 2016).

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.

### 3.5. **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 *me-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 *me-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).

### 3.6. MySQL

MySQL adalah *multiuser database* yang menggunakan bahasa *Structured Query Language (SQL)*. MySQL dalam operasi *client server* melibatkan *server daemon* MySQL disisi *server* dan berbagai macam program serta *library* yang berjalan disisi *client*. MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TEX, mengaku mampu

menyimpan data lebih dari 40 *database*, 10.000 tabel, dan sekitar 7.000.000 baristotalnya kurang lebih 100 *Gigabyte* data (Hendrianto, 2014).

### **3.7. PHP**

PHP (Hypertext Preprocessor), merupakan bahasa pemrograman pada sisi server yang memperbolehkan programmer menyisipkan perintah-perintah perangkat lunak *web server* (Apache, IIS, atau apapun) akan dieksekusi sebelum perintah itu dikirim oleh halaman ke browser yang *me-request*-nya, contohnya adalah bagaimana memungkinkannya memasukkan tanggal sekarang pada sebuah halaman *web* setiap kali tampilan tanggal dibutuhkan. Sesuai dengan fungsinya yang berjalan di sisi server maka PHP adalah bahasa pemrograman yang digunakan untuk membangun teknologi *web application* (Februariyanti & Zuliarso, 2012).

### **3.8. Bahasa Pemrograman Java**

Java dikembangkan oleh Sun Microsystems pada Agustus 1991. Java disebut juga merupakan hasil perpaduan sifat dari sejumlah bahasa pemrograman, yaitu C dan C++. Pemrograman Java bersifat tidak bergantung pada *platform*, yang artinya, java dapat dijalankan pada sembarang komputer dan bahkan pada sembarang sistem operasi. Sebagaimana halnya C++, salah satu bahasa yang mengilhami Java, Java juga merupakan bahasa pemrograman berorientasi objek. Sebagai bahasa pemrograman berorientasi objek, Java menggunakan kelas untuk membentuk suatu objek. Karakteristik Java antara lain adalah berorientasi objek (*object-oriented*), terdistribusi (*distributed*), sederhana (*simple*), aman (*secure*), *interpreted*, *robust*, *multithreaded*, dan dinamis (Tuteja & Dubey, 2012).

### **3.9. Android studio**

Android Studio adalah Lingkungan Pengembangan Terpadu Integrated Development Environment (IDE) untuk pengembangan aplikasi Android, berdasarkan IntelliJ IDEA. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android Studio menawarkan fitur lebih banyak

untuk meningkatkan produktivitas Anda saat membuat aplikasi Android (Android Studio, 2013).

### **3.10. Laravel**

Laravel adalah sebuah framework PHP yang dirilis dibawah lisensi MIT, dibangun dengan konsep MVC (*Model View Controller*). Laravel adalah pengembangan *website* berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu (Cloud Hosting Indonesia, 2016).

Pada bab sudah dijelaskan mengenai teori dasar yang digunakan oleh penulis sebagai acuan dalam membangun aplikasi. Pada bab selanjutnya, akan dijelaskan mengenai analisis dan perancangan aplikasi yang akan dibangun.