

## **BAB II**

### **TINJAUAN PUSTAKA**

#### 2.1 Penelitian Terdahulu

Penelitian terdahulu menjadi salah satu acuan penulis dalam pembuatan laporan akhir ini, sehingga penulis dapat memperkaya teori yang digunakan dalam mengkaji penelitian aplikasi pelaporan kerusakan fasilitas kampus berbasis *android*.

Rujukan penelitian pertama yaitu dari Ultach Enri dan Chaerur Rozikin dalam jurnal yang berjudul Sistem Pengaduan Kerusakan Fasilitas Ruang Kelas Berbasis *Android*. Dalam penelitiannya, bahwa teknologi berbasis *android* bisa menjadi solusi yaitu dengan membuat sistem untuk menyelesaikan permasalahan tersebut. Aplikasi yang akan dibuat dapat menerima laporan berbagai kerusakan peralatan pendukung ruang kelas kemudian laporan tersebut akan sampai kepada pihak yang berwenang yang menanganinya (Enri dan Rozikin, 2019: pp 116-129).

Penelitian selanjutnya yaitu jurnal dari Galih Nalendro yang berjudul Sistem Informasi Pengaduan Layanan Universitas. Dalam penelitiannya bahwa ilmu pengetahuan teknologi (IPTEK) dapat berperan besar dalam mempermudah pekerjaan tim pengelola layanan universitas untuk dapat mengetahui semua keluhan dan aduan setiap warga universitas. Sebuah sistem informasi berbasis *mobile* dapat dibangun sebagai sarana bagi semua warga universitas agar dapat dengan mudah mengadukan semua permasalahan di lingkungan universitas. Semua keluhan dan aduan akan ditampung oleh sistem informasi sehingga dapat diproses secepat mungkin (Nalendro, 2019: 53-54).

Sedangkan penelitian yang akan dilakukan penulis tidak jauh berbeda dengan penelitian sebelumnya yaitu Rancang Bangun Aplikasi Pelaporan Kerusakan Sarana dan Prasarana berbasis *Android* yang Terintegrasi dengan *Web* Kampus Pintar, yang digunakan oleh mahasiswa dan dosen sebagai pengguna aplikasi.

## 2.2 Pengertian Aplikasi

Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti sistem perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hampir dilakukan manusia (Pramana, 2012:17). Perangkat lunak aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya. Jika ingin mengembangkan program aplikasi sendiri, maka untuk menulis program aplikasi tersebut, dibutuhkan suatu bahasa pemrograman, yaitu *language software*, yang dapat berbentuk *assembler*, *compiler* ataupun *interpreter*.

Jadi *language software* merupakan bahasanya dan program yang ditulis merupakan program aplikasinya. *Language software* berfungsi agar dapat menulis program dengan bahasa yang lebih mudah, dan akan menterjemahkannya ke dalam bahasa mesin supaya bisa dimengerti oleh komputer. Sekarang, banyak sekali program-program aplikasi yang tersedia dalam bentuk paket-paket program. Ini adalah program-program aplikasi yang sudah ditulis oleh orang lain atau perusahaan-perusahaan perangkat lunak.

Beberapa perusahaan perangkat lunak telah memproduksi paket-paket perangkat lunak yang mempunyai reputasi internasional. Program-program paket tersebut dapat diandalkan, dapat memenuhi kebutuhan pemakai, dirancang dengan baik, relatif bebas dari kesalahan-kesalahan, *userfriendly* (mudah digunakan), mempunyai dokumentasi manual yang memadai, mampu dikembangkan untuk kebutuhan mendatang, dan didukung perkembangannya.

Akan tetapi, bila permasalahannya bersifat khusus dan unik, sehingga tidak ada paket-paket program yang sesuai untuk digunakan, maka dengan terpaksa harus mengembangkan program aplikasi itu sendiri (Hartono, 2004, 397-398).

### 2.3 Pengertian *Android*

*Android* adalah sebuah sistem operasi yang bersifat *open source*(terbuka) yang dimiliki oleh *Google Inc* (Triadi, 2013). *Android* menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri sehingga dapat digunakan oleh berbagai macam piranti bergerak (Wahana, 2012). Sifat *Android* yang terbuka menjadi jawaban dari keberagaman pengguna sehingga tiap-tiap pengguna dapat menciptakan sendiri aplikasi sesuai kebutuhan. Berkat fiturnya yang selalu *update*, *Android* berkembang secara signifikan dari tahun ke tahun. Samapi saat ini sudah terdapat beberapa versi *Android* sejak sistem operasi ini dirilis secara resmi pada tahun 2007. Nama versi *Android* dari generasi ke generasi sangat menarik karena menggunakan nama makanan yang diurutkan secara alfabetik seperti *Cupcake*(Versi 1.5), *Donut*(Versi 1.6), *Éclair*(Versi 2.0-2.1) hingga versi terbaru saat ini *Oreo*(Versi 8.0).

*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 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 (Safaat, 2010).

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*. *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*.

Arsitektur *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).

#### 2.4 Pengertian *Web Service*

*Web Service* adalah aplikasi yang dibuat agar dapat dipanggil dan diakses oleh aplikasi lain melalui internet dengan menggunakan format pertukaran data sebagai format pengiriman pesan (Kasman, 2016). *Web Service* digunakan sebagai suatu fasilitas yang menyediakan layanan (dalam bentuk informasi atau data) kepada sistem lain, sehingga dapat berinteraksi dengan sistem tersebut melalui layanan-layanan yang disediakan. *Web Service* menyimpan data informasi dalam format JSON atau XML, sehingga data ini dapat diakses oleh sistem lain walaupun berbeda platform, sistem operasi, dan bahasa pemrograman. Sebuah *Web Service* dapat dipanggil oleh aplikasi lain dengan menggunakan bantuan HTTP (*Hyper Text Transfer Protocol*). Karena *Web Service* menggunakan protokol HTTP, tentu PHP sebagai bahasa pemrograman web menjadi salah satu kekuatan dalam bahasa pemrograman yang mengelola *Web Services*.

#### 2.5 Pengertian JSON

JSON (*Javascript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer, menurut Json Org. JSON(2005). JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh

karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran data (Rahman dkk, 2013).

## 2.6 Arsitektur *Client-Server*

Arsitektur *client – server* merupakan suatu desain dari sebuah aplikasi yang terdiri dari *client* dan *server* di mana saling berkomunikasi satu sama lain ketika mengakses *server* dalam satu jaringan( Eka, 2015). *Client* sendiri merupakan mesin pengguna yang menyediakan *user interface* dan *local processing* di mana melakukan suatu permintaan data atau layanan ke *server*. Sedangkan *server* merupakan mesin yang menyediakan data atau layanan yang diminta oleh *client*.

## 2.7 Pengertian Postman

Postman merupakan sebuah *tool REST client* berbasis *web* yang tersedia dalam bentuk ekstensi pada Google Chrome. Berguna untuk melakukan pengujian dari *Web Service* yang sedang dikembangkan (Rohman, 2017).

## 2.8 Penegertian *Representational State Transfer*

*Representational State Transfer* (REST) adalah salah satu rancang bangun arsitektural sistem *Web Service*. REST merupakan salah satu model design rancangbangun *Web Service* selain SOAP dan WSDL. REST mulai berkembang pesat karena sistemnya yang lebih sederhana daripada SOAP dan WSDL. Sisi sederhana dari REST yang membuat menarik adalah REST dapat dibangun dengan sedikit *tools* (Rozali, 2011). Selain itu untuk melakukan *testing* terhadap REST *service* dapat dilakukan secara sederhana pada *web browser* tanpa harus melakukan simulasi *client – server*. *Web API* merupakan sebuah perkembangan dalam *Web Service* yang menggunakan REST sebagai dasar arsitekturnya. *Web API* memiliki kemampuan untuk melakukan kombinasi terhadap lebih dari satu *Web Service* yang nantinya akan menghasilkan sebuah aplikasi baru yang diberi nama *mashup*. Ketika digunakan dalam pengembangan *web*, *web API* membuat

rancangan request. HTTP lengkap dengan pendefinisian struktur dari pesan, biasanya dibentuk dalam format XML atau JSON. Ketika sedang menjalankan beragam *Web Service*, masing –masing *sub service* tersebut berjalan secara autonomous. *User* tidak memiliki kontrol terhadap masing – masing *service* tersebut. Dengan segala kemudahan yang ada, web API juga memiliki kekurangan yaitu: penyedia *Web Service* dapat merubah, menghilangkan, ataupun mengupdate program tanpa notifikasi. Reliabilitasnya yang kurang begitu didukung. Kegagalan terkadang muncul ketika dieksekusi.

## 2.9 Pengertian *Flutter*

*Flutter* ditunjukan untuk mempermudah dan mempercepat proses pengembangan aplikasi *mobile* yang dapat berjalan di atas *android* dan iOS tanpa harus mempelajari dua bahasa pemrograman sekaligus. Aplikasi *android* dan iOS dapat dibuat dengan basis pemrograman yang sama yaitu *Dart*. (Raharjo, 2019). *Flutter* adalah teknologi untuk membangun suatu *mobile apps* yang dibuat oleh *Google*. *Flutter* menggunakan bahasa *Dart* yang juga dibuat oleh *Google* dan merupakan bahasa *general purpose*. Artinya, *Dart* mampu berjalan di banyak platform. *Flutter* merupakan *cross-platform framework*, alias aplikasi yang dapat digunakan di lebih dari satu platform. Aplikasi yang dibuat dengan menggunakan *Flutter* dapat dijalankan baik di platform *Android* maupun iOS (Wiguna, 2019).