

BAB II

TINJAUAN PUSTAKA DAN LANDASAN TEORI

2.1 Tinjauan Pustaka

Layanan Portal Mahasiswa dalam dunia kampus sangatlah penting. Layanan ini berfungsi sebagai pengambilan mata kuliah, melihat kewajiban pembayaran, serta jadwal kurikulum. Layanan Portal Mahasiswa ini sudah terkoneksi dengan *internet* sehingga kerap disebut sebagai KRS Online.

Sebelumnya, di UMY sendiri sudah ada layanan portal mahasiswa atau KRS Online seperti ini namun masih dalam bentuk *website*/situs. Maka penulis melakukan pengembangan dengan membuat KRS Online berbasis aplikasi yang menggunakan sistem operasi Android. Penulis juga menggunakan situs resmi KRS Online sebagai referensi pembuatan aplikasi ini, karena untuk layanannya tidak berbeda jauh dengan yang ada di situs tersebut.

Dalam skripsi yang berjudul "PENGEMBANGAN APLIKASI KRS ONLINE PADA SMARTPHONE BERBASIS ANDROID", telah membuat aplikasi serupa dengan milik penulis, ia mengatakan tujuan dari penelitiannya ini antara lain yaitu untuk membangun sebuah sistem pengisian Kartu Rencana Studi (KRS) berupa layanan online berbasis Android sebagai media alternatif kepada mahasiswa untuk melakukan proses pengisian Kartu Rencana Studi (KRS) kapanpun dan dimanapun sehingga lebih praktis dan efisien. (Laledak, 2015)

Penulis meninjau dari aplikasi KRS Online sebelumnya, bahwa fitur tidak jauh berbeda dengan fitur yang ada di website KRS Online Universitas Muhammadiyah Yogyakarta. Maka, penulis akan mengembangkan lagi aplikasi ini dengan memperbaiki tampilan agar lebih *user-friendly* sehingga memudahkan mahasiswa yang mengaksesnya karena sudah familiar dengan fitur yang ada. Selain itu juga penulis menambahkan dialog peringatan pada halaman KRS Online jika pada waktu pembayaran telah tiba maka akan diberikan sebuah pemberitahuan.

2.2 Landasan Teori

2.2.1 Perangkat Lunak (*Software*)

Perangkat lunak (*Software*) merupakan sekumpulan data digital yang diatur dan disimpan dalam komputer berisi instruksi yang berfungsi sebagai penunjang atau pendukung kinerja *hardware* dalam menjalankan suatu perintah. Yang mana dapat diartikan bahwa *software* sebagai penghubung antara *user* dan *hardware* agar dapat berjalan dengan baik. (Roger S., 2002)

Bahasa komputer dengan bahasa manusia sangat berbeda. Komputer hanya mengenali 0 atau 1 atau Ya dan Tidak. Jadi untuk menjalankan sebuah perintah, perlu penerjemah dari bahasa manusia ke bahasa komputer atau sering disebut *compiler* sehingga komputer dapat mengenali bahasa atau kode perintah tersebut.

Dalam skripsi yang ditulis oleh mahasiswa fakultas Teknik Elektronika Universitas Negeri Yogyakarta mengelompokkan perangkat lunak ke dalam dua kategori:

1. Sistem operasi yang mengatur cara kerja komputer. Seperti mengatur konfigurasi dan mengontrol penggunaan hardware.
2. *Application Software*, yaitu suatu *software* yang dirancang untuk maksud tertentu. *Application Software* bekerja di dalam suatu system operasi (Nafngan, 2015).

2.2.2 Mobile Application

Mobile Application adalah *software* yang dikembangkan untuk perangkat *mobile* atau perangkat nirkabel seperti *smartphone* dan *tablet*. *Mobile Application* juga memungkinkan penggunanya mengakses suatu layanan dengan mudah dimanapun dan kapanpun.

David Pogue yang merupakan seorang pakar teknologi mengatakan bahwa *smartphone* yang lebih baru dapat dikatakan sebagai "telepon app" untuk membedakan dari perangkat yang kurang canggih sebelumnya. Yang artinya, *mobile application* adalah *software* yang dijalankan di dalam telepon genggam atau *mobile phone* (Pogue, 2009).

2.2.3 Android

Android adalah salah satu sistem operasi yang berjalan di atas *platform* telepon pintar atau *smartphone* berbasis Linux dan bersifat *open source*. Sistem operasi ini memungkinkan pengguna mengoperasikan aplikasi pada *smartphone* mereka (Android Overview, 2012).

Pada 2005 Google membeli sistem operasi Android dari Android, Inc. dan merilisnya pada tahun 2007 serta menjual ponsel pertamanya di bulan Oktober 2008. Android telah digunakan oleh beberapa perangkat keras seperti *Smartphone*, TV, perangkat mobil, jam tangan, *game console*, dan kamera.

Di bulan April hingga Mei 2013 Android telah digunakan oleh 71% *mobile app developer* dan menjadi *platform* terpopuler bagi beberapa *developer*. Dan mengalami kenaikan dari 583 juta pengguna aktif menjadi satu miliar lebih pada tahun 2014 (Elgin, 2005).

Sistem operasi ini juga telah mengalahkan Symbian pada tahun 2010 berkat kontribusi perkembangan Android. Banyak perusahaan di bidang teknologi yang menggunakan Android karena memiliki biaya yang rendah dalam pengembangannya, mudah dalam mengkustom sistemnya, serta tidak terlalu berat dalam pengoperasiannya (Victor, 2011)

Android mulai menguasai pasar teknologi di bulan November 2013 dengan beragam versi android sejumlah 11.868 perangkat. Dan 48 miliar perangkat lunak juga telah terinstal melalui Google Play serta diaktifkan oleh lebih dari 900 juta perangkat berbasis Android (Arthur, 2013)

2.2.4 SDLC (*Software Developing Life Cycle*)

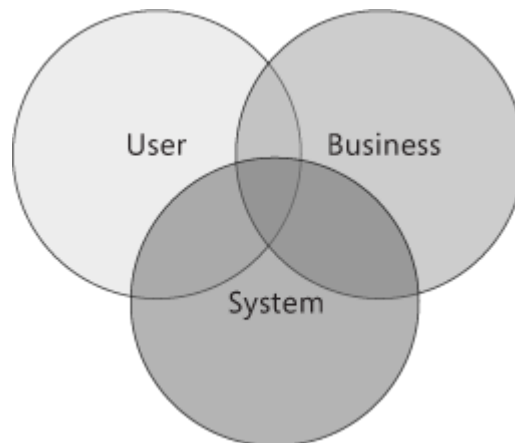
SDLC (*System Developing Life Cycle*) proses dalam mengembangkan suatu *software* atau perangkat lunak yang melalui tahapan-tahapan tertentu seperti Rencana (*Planning*), Analisa (*Analysis*), Rancangan (*Design*), Implementasi (*Implementation*), Uji Coba (*Testing*), dan Pengelolaan (*Maintenance*). SDLC memungkinkan pengembang agar dalam proses pengembangan *software* dapat terstruktur mengikuti alur sesuai tahapan yang ada.

2.2.5 Software Architecture

“The highest-level breakdown of a system into its parts; the decisions that are hard to change; there are multiple architectures in a system; what is architecturally significant can change over a system’s lifetime; and, in the end, architecture boils down to whatever the important stuff is.”

Bisa disimpulkan bahwa *Software Architecture* adalah sebuah proses untuk mendefinisikan struktur dari suatu aplikasi yang dapat memenuhi seluruh kriteria dari sisi teknis dan juga operasional, dengan pertimbangan kualitas seperti *performance*, *security*, dan *manageability* (Clements, et al., 2010)

Pada pembuatan sebuah aplikasi ada 3 komponen yang sangat penting dan komponen-komponen itu akan saling terkait satu sama lain. Contohnya seperti Gambar 2.1.



Gambar 2. 1 Komponen Utama

Tujuan dibuatnya *Software Architecture* adalah menghubungkan bisnis dengan kebutuhan teknis melalui *Use Case* yang telah dibuat dan menemukan cara untuk mengimplementasikan *use case* tersebut. *Software Architecture* yang baik akan meminimalisir kemungkinan resiko/kegagalan yang diciptakan oleh para *developer*. Selain itu, *software architecture* yang baik juga dapat beradaptasi dengan perubahan yang akan terjadi seiring waktu dalam teknologi perangkat keras dan perangkat lunak, serta dalam skenario dan persyaratan pengguna. Seorang arsitektur juga harus mempertimbangkan hasil rancangan arsitektur yang sudah dibuat agar memiliki *performance* dan *security* yang baik.

2.2.6 KRS Online

Kita sering mendengar istilah KRS dalam dunia kampus. KRS memiliki kepanjangan Kartu Rencana Studi. Kartu Rencana Studi adalah layanan yang membantu mahasiswa dalam memilih mata kuliah yang akan diambil untuk ditempuh dalam satu semester. Jadi KRS Online merupakan layanan akademik dengan sistem *online* yang berfungsi membantu mahasiswa dalam mengambil mata kuliah untuk satu semester ke depan hanya dengan menggunakan *gadget* yang dimiliki mahasiswa semisal, *laptop*, *smartphone*, *tablet*, atau *personal computer*.

2.3 Basis Data

Basis data atau biasa dikenal dengan *database* adalah berbagai informasi yang saling berhubungan dan tersimpan dalam komputer secara terstruktur dan dapat dibaca, diubah, serta dapat dihapus dengan perangkat lunak yang dikenal dengan Database Management System atau DBMS dengan menggunakan *query database* (Whitten & Bentley, 2007)

Suatu *database* sangat penting dalam pengembangan perangkat lunak, karena bertugas sebagai penyimpan data dan juga mempermudah dalam mencari dan mengelola data dengan *software* yang dikenal sebagai *Database Management System* (DBMS). Selain itu juga untuk menjamin keamanan data karena data tersebut terpusat pada satu server dan dapat digunakan siapa saja yang terkoneksi dengan *server* tersebut dalam suatu perusahaan.

2.3.1 Microsoft SQL Server

Microsoft SQL Server merupakan sebuah *Relation Database Management System* (RDBMS) yang dikembangkan oleh Microsoft. Sebagai *server* basis data, aplikasi ini berfungsi untuk menyimpan dan menyediakan data.

Basis data yang digunakan penulis sama dengan basis data yang digunakan pada website KRS Online. Jadi, penulis hanya membuat suatu *service* yang berfungsi untuk menghubungkan aplikasi dengan *database*, sehingga informasi data dapat ditampilkan dalam aplikasi.

Penulis menggunakan aplikasi SQL Server versi 17.8.1, yang nantinya akan menjadi sumber data aplikasi KRS Online ini. Untuk menghubungkan aplikasi dengan basis data, diperlukan sebuah *web service*.

2.3.2 Web Service

Web Service merupakan teknologi, proses, dan perangkat lunak yang memberikan *support* untuk integrasi bisnis terutama melalui lingkungan yang berbasis *internet*. Umumnya, dengan menggunakan URL dapat mengidentifikasi *web service* seperti halnya web. Perbedaan antara *web service* dengan *web* secara umum merupakan interaksi dari *web service* itu sendiri. URL dari *web service* berisi kumpulan data, *command*, serta *configuration* dan/atau penulisan *syntax* yang berfungsi untuk membangun fungsi atau fitur dari perangkat lunak tersebut.

Dalam pengembangan software ini, penulis menggunakan *web service* jenis REST (*Representational State Transfer*). REST adalah gaya arsitektur yang mendefinisikan satu set batasan yang akan digunakan untuk membuat *web service*. *Web Service* yang sesuai dengan gaya arsitektur REST menyediakan interoperabilitas antara sistem komputer di Internet (Benslimane & Dustdar, 2008)

2.3.3 Framework Laravel

Laravel adalah *web framework* PHP *open source* yang dibuat oleh Taylor Otwell dan ditujukan untuk pengembangan aplikasi *web* mengikuti pola arsitektur MVC. MVC adalah Model-View-Controller, dimana Model merupakan penghubung ke database, View untuk memunculkan tampilan, dan Controller sebagai code yang memfungsikan View dan Model.

Alasan penulis menggunakan Laravel adalah untuk menghindari *repetitive work* atau bisa disebut pekerjaan yang berulang. Jadi untuk beberapa fitur *web*, penulis menyerahkan ke library lain bukan hasil buatan sendiri. Selain itu juga untuk meningkatkan keamanan. Dan yang paling penting adalah, *framework* ini *open source* yang berarti semua pengguna dapat menggunakan tanpa ada biaya apapun. Namun dari kelebihan-kelebihan tadi, ada kelemahan yang dimiliki oleh Laravel. Dikarenakan banyaknya *file* dan *asset* yang harus dimuat, maka *framework* ini menjadi lebih berat daripada *framework* lainnya seperti Codeigniter dan lain-lain.