

BAB III

LANDASAN TEORI

Pada bab ini akan dibahas berbagai teori yang melandasi dalam membangun sistem ini.

3.1 Sistem Informasi

Menurut Hall (2006, p6), sistem informasi adalah serangkaian prosedur formula dimana data dikumpulkan, diproses menjadi informasi dan didistribusikan kepada pengguna.

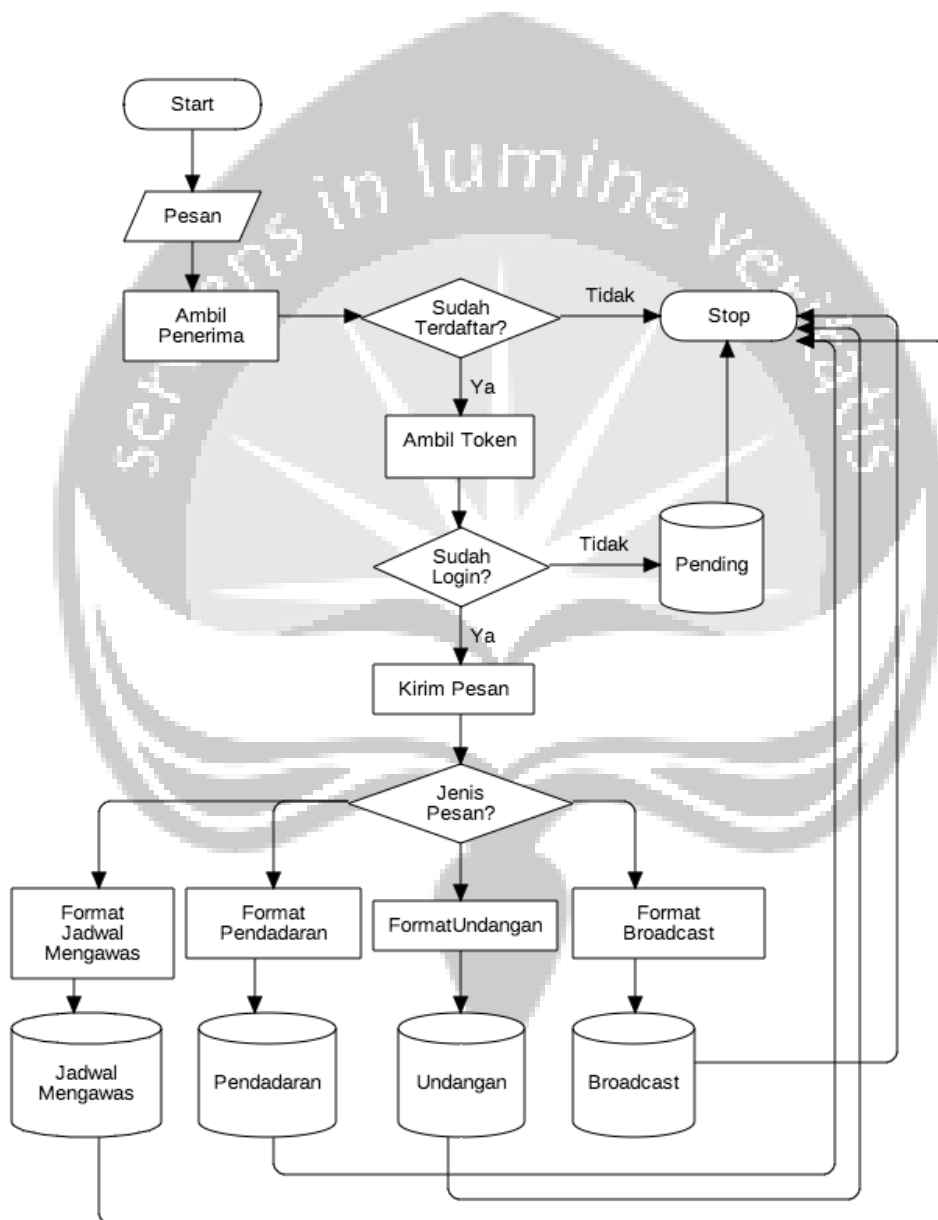
Menurut Satzinger, Jackson dan Burd (2010 : 7) sistem informasi merupakan sekumpulan komponen terpisah yang berfungsi untuk mengumpulkan, mengolah, menyimpan dan menyediakan tugas-tugas dalam bisnis.

Berdasarkan definisi diatas maka sistem informasi merupakan kumpulan komponen-komponen yang saling berkolaborasi untuk mengumpulkan, menyimpan, memproses dan menyebarkan informasi untuk tujuan tertentu di dalam organisasi.

3.2 Aplikasi Pengumuman Terpadu

Sistem informasi pengumuman merupakan aplikasi yang ditujukan untuk mempermudah distribusi informasi perkuliahan dari dosen ke mahasiswa ataupun dari pihak Tata Usaha fakultas ke dosen. Dengan adanya aplikasi ini setiap pengumuman atau berita mengenai perkuliahan akan lebih mudah dan cepat sampai kepada penerima karena pesan yang dikirimkan bersifat *realtime*. Pesan yang dikirimkan berupa teks. Yang dapat mengirim pesan adalah dosen dan TU. Dosen hanya dapat mengirim pesan ke mahasiswa sedangkan TU dapat mengirimkan pesan ke dosen dan mahasiswa. Pesan yang dikirimkan ke mahasiswa

hanya berupa pengumuman biasa yang sifatnya umum. Ketika dosen atau TU mengirim pesan ke mahasiswa maka server aplikasi akan mengelola pesan tersebut dan mengirimkannya ke GCM Service dan kemudian dikirimkan ke mahasiswa. Proses pengiriman pesan dapat dilihat pada *flowchart* di gambar 3.1.



Gambar 3. 1 Flowchart Pengiriman Pesan

Pesan yang dikirimkan ke dosen dapat berupa pengumuman biasa, undangan, undangan pendadaran, jadwal dan jadwal mengawas ujian. Pesan yang dikirimkan ke dosen berasal dari TU kemudian diinputkan ke sistem lalu diolah dan dikategorikan, kemudian dikirim ke GCM Service untuk kemudian dikirimkan ke dosen. Pesan yang masuk ke *device* penerima akan diolah sesuai dengan jenis pesannya dan ditampilkan ke pengguna.

3.3 Aplikasi Mobile

Aplikasi mobile dapat diartikan sebagai sebuah produk dari sistem komputasi *mobile*, yaitu sistem komputasi yang dapat dengan mudah dipindahkan secara fisik dan yang komputasi kemampuan dapat digunakan saat mereka sedang dipindahkan. Contohnya adalah *personal digital assistant* (PDA), *smartphone* dan ponsel (B'Far, 2005:3).

Berdasarkan jenisnya, Fling (2009:70) membagi aplikasi mobile menjadi beberapa kelompok yaitu:

a. *Short Message Service* (SMS)

Merupakan aplikasi *mobile* paling sederhana, dirancang untuk berkirim pesan dan berguna ketika terintegrasi dengan jenis aplikasi mobile lainnya.

b. *Mobile Websites* (Situs Web Mobile)

Merupakan situs web yang dirancang khusus untuk perangkat *mobile*. Situs web *mobile* sering memiliki desain yang sederhana dan biasanya bersifat memberikan informasi.

c. *Mobile Web Application* (Aplikasi Web Mobile)

Aplikasi web mobile merupakan aplikasi mobile yang tidak perlu diinstal atau dikompilasi pada perangkat target. Menggunakan XHTML, CSS, dan JavaScript,

aplikasi ini mampu memberikan pengguna pengalaman layaknya aplikasi *native/asli*.

d. Native Application (Aplikasi Asli)

Merupakan aplikasi mobile yang harus diinstal pada perangkat target. Aplikasi ini dapat disebut aplikasi platform, karena aplikasi ini harus dikembangkan dan disusun untuk setiap *platform mobile* secara khusus.

3.4 Android

Mulyadi (2010:5) menerangkan Android merupakan subset perangkat lunak untuk perangkat *mobile* yang meliputi sistem operasi, *middleware* dan aplikasi yang di rilis oleh Google. Mulyadi (2010:8) menjelaskan arsitektur Android sebagai berikut.

a. Aplikasi

Level aplikasi ini nantinya memuat aplikasi yang dikembangkan oleh para *programmer*.

b. *Application framework*

Pengembang aplikasi mempunyai akses penuh menuju *framework API* yang sama dengan yang digunakan oleh aplikasi inti. Arsitektur aplikasi dirancang agar komponen aplikasi dapat digunakan kembali dengan mudah.

c. *Libraries*

Android menyertakan *libraries C/C++* yang digunakan oleh berbagai komponen dari sistem *Android*. Kemampuan *libraries* dapat diakses oleh pengembang aplikasi *Android* melalui *Application Framework Android*.

d. *Android runtime*

Android terdiri dari satu set *core libraries* yang menyediakan sebagian besar fungsi yang sama dengan *core libraries* bahasa pemrograman *Java*.

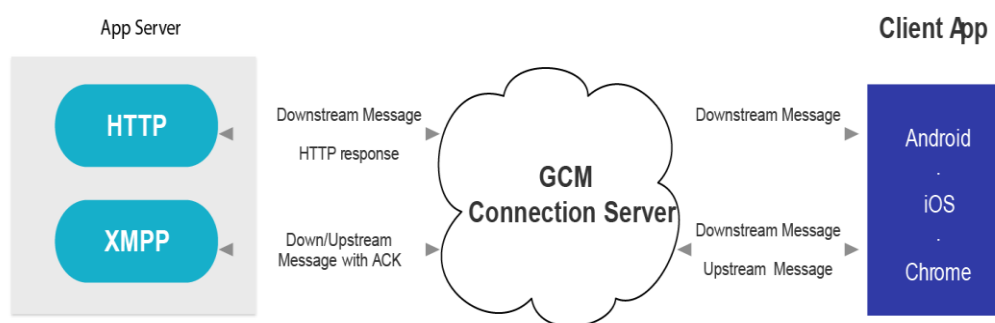
e. *Kernel Linux*

Android bukan *Linux*, akan tetapi *Android* dibangun di atas kernel *Linux 2.6* sehingga kehandalannya dapat dipercaya.

3.5 Google Cloud Messaging

Merupakan layanan gratis yang disediakan oleh Google yang memungkinkan para pengembang aplikasi untuk bertukar pesan antara *server* dan aplikasi di klien. GCM dapat mengirimkan data maksimal sebesar 4kb ke klien. GCM menangani semua aspek mulai dari data dikirim hingga data diterima oleh klien.

Data dikirim menggunakan protokol HTTP atau XMPP melalui *GCM Connection Server* yang disediakan oleh Google. Gambar 3.1 merupakan arsitektur GCM yang dibuat oleh Google.



Gambar 3. 2 Arsitektur GCM

(Sumber : <https://developers.google.com/cloud-messaging/gcm#arch>)

Berikut ini adalah penjelasan dari gambar 3.2.

- a. *Google Connection Server* yang disediakan oleh google akan menerima pesan yang dikirimkan oleh server aplikasi milik kita dan mengirimkan pesan tersebut ke aplikasi di klien.
- b. *App Server* merupakan server yang harus disediakan oleh kita sendiri serta harus menggunakan protokol HTTP ataupun XMPP untuk berinteraksi dengan *GCM Connection Server*.
- c. *Client App* merupakan aplikasi di klien yang akan menerima pesan yang dikirimkan melalui *Google Connection Server* dan untuk mengirim atau menerima pesan dari GCM maka aplikasi ini harus didaftarkan terlebih dahulu ke GCM dan akan mendapatkan kode identifikasi yang bersifat unik atau disebut juga *token*.

3.6 Web Services

Menurut W3C, *web service* merupakan komponen aplikasi yang berkomunikasi menggunakan protokol terbuka. *Web service* dibangun untuk memungkinkan aplikasi web untuk saling bekerja sama. Dengan web service, aplikasi web dapat mempublikasikan fungsi-fungsinya ke seluruh dunia.

Menurut www.IBM.com *Representational State Transfer* (REST) merupakan seperangkat prinsip arsitektur yang dapat digunakan untuk merancang web service yang berfokus pada sumber daya sistem, termasuk bagaimana sumber daya dikirim melalui protokol HTTP oleh berbagai macam klien yang ditulis dalam bahasa

pemrograman yang berbeda beda. REST *web service* memiliki empat prinsip utama yaitu:

- a. Menggunakan HTTP *method* secara eksplisit. Salah satu karakteristik utama dari layanan REST adalah penggunaan HTTP *method* secara eksplisit dengan cara yang mengikuti protokol seperti yang didefinisikan oleh Request for Comments (RFC) 2616.
- b. *Stateless* meningkatkan kinerja dan menyederhanakan desain aplikasi REST *web service*. Hal ini dikarenakan dengan tidak adanya *state server* tidak perlu melakukan sinkronisasi *session* data dengan aplikasi klien.
- c. Mengekspose struktur direktori. REST *web service* harus memiliki *Uniform Resource Identifier* (URI) yang intuitif dan mudah ditebak. URI didefinisikan sebagai semacam dokumentasi diri antarmuka yang memerlukan sedikit, jika ada, penjelasan atau referensi bagi pengembang untuk memahami petunjuk mendapatkan sumber daya yang terkait.
- d. Bertukar data dengan menggunakan XML, *Java Script Object Notation* (JSON) atau kedua nya.