

BAB III LANDASAN TEORI

3.1 Web server

Web server adalah *software* yang memberikan layanan data yang mempunyai fungsi untuk menerima permintaan HTTP (*HyperText Transfer Protocol*) atau HTTPS yang dikirim oleh klien melalui *web browser* dan mengirimkan kembali hasilnya dalam bentuk halaman *web* yang umumnya berbentuk dokumen HTML (*HyperText Markup Language*). *Web server* berguna sebagai tempat aplikasi *web* dan sebagai penerima *request* dari *client* (Indra Warman & Zahni, 2013). Pada umumnya *web server* telah dilengkapi pula dengan mesin penerjemah bahasa skrip yang memungkinkan *web server* menyediakan layanan situs *web* dinamis dengan memanfaatkan pustaka tambahan seperti PHP (*PHP: Hypertext Preprocessor*) dan ASP (*Active Server Pages*).



Gambar 3.1 Arsitektur *Web server*

Gambar 3.1 merupakan arsitektur dari *web server*. *Client* melakukan HTTP *request* ke *web server* dan *web server* akan mengembalikan *request* berupa halaman *website* meliputi HTML, *image*, CSS, dan javascript. *Server* juga dapat melakukan *query* atau *request* data ke *database* jika *client* ingin mengelola data. *Database* akan mengembalikan *request* dari *server* berupa data dan *server* menampilkannya berupa halaman *web* ke *client*. Dua contoh *web server* yang sering digunakan adalah Apache

dan IIS. Sedangkan *database* yang digunakan adalah MySQL, MySQL merupakan software sistem manajemen *database* (DBMS) yang sangat populer atau banyak digunakan untuk membangun aplikasi *web* sebagai sumber data. MySQL bersifat *open source*, mudah, dan cepat dalam mengeksekusi *query*.

3.2 Situs Berita Online

Situs berita *online* merupakan *website* penyedia informasi berupa berita. Salah satu yang sering diaplikasikan dalam praktik jurnalistik *modern* adalah situs berita (Aswari, 2013). Perkembangan teknologi yang menjadi penyebab banyaknya situs berita *online* di Indonesia. Situs berita termasuk dalam media *online*. Portal pada media *online* mempunyai variasi dalam pemberitaan yang menarik dengan gambar dan video (Astari, 2014). Variasi pemberitaan dan tampilan yang menarik membuat pembaca lebih senang membaca berita pada situs berita *online*.



Gambar 3.2 Contoh Situs Berita Online

Gambar 3.2 merupakan dua buah contoh situs berita *online* terbesar di Indonesia, yaitu : kompas dan detik.

Situs berita mempunyai desain yang berbeda-beda. Terdapat iklan di situs berita tersebut disamping kanan dan di atas. Selain iklan dalam bentuk banner, ada pula iklan dalam bentuk *pop up* sehingga pembaca harus mengklik tanda *close* untuk menutup iklan. *Slider* juga menjadi pemanis dalam situs berita yang biasanya menampilkan berita-berita terbaru yang menjadi *trending* topik.

3.3 Web Service

Web service adalah sistem yang dirancang untuk mendukung interoperabilitas dan interaksi antar sistem pada suatu jaringan. *Web service* mempublikasikan aplikasi dalam bentuk layanan dan mudah ditemukan dan diakses oleh aplikasi lain secara *online* tanpa tergantung pada *platform* yang digunakan (Kuswandi, et al., 2007).

Secara umum, *web service* dapat diidentifikasi dengan menggunakan URL seperti hanya *web* pada umumnya. Namun yang membedakan *web service* dengan *web* pada umumnya adalah interaksi yang diberikan oleh *web service*. *Web service* bersifat terbuka untuk kepentingan integrasi data kolaborasi data yang diakses secara *online* oleh berbagai pihak dengan teknologi yang dimiliki oleh masing-masing pengguna. (Sutanta & Mustofa, 2012). *Web service* tidak mempunyai tampilan karena *web service* termasuk dalam *Bussiness-Service tier*. Pada *web service* hubungan antara *client* dan *server* tidak terjadi secara langsung.

Hubungan antara *client* dan *server* dijumpai oleh file *web service* dalam format tertentu. Sehingga

akses terhadap *database* akan ditangani tidak secara langsung oleh *server*, melainkan melalui perantara yang disebut sebagai *web service* (Pattera, et al., 2014). *Web service* dibagi menjadi dua macam yaitu SOAP (*Simple Object Access Protocol*) dan REST (*Representational State Transfer*).

a. SOAP

SOAP merupakan singkatan dari *Simple Object Access Protocol*. SOAP adalah protokol untuk pertukaran informasi dengan disentralisasi dan terdistribusi. SOAP dibangun dengan menggunakan protokol komunikasi HTTP, sehingga SOAP dapat berkomunikasi dengan berbagai aplikasi meskipun terdapat perbedaan sistem operasi, teknologi, dan bahasa pemrogramannya. Peran SOAP adalah sebagai protokol pemaketan untuk pesan yang digunakan secara bersama oleh aplikasi-aplikasi penggunanya. Spesifikasi yang digunakan adalah dengan berbasis XML (*Extensible Markup Language*) untuk informasi yang ditransfer, serta sekumpulan aturan bagi translasi aplikasi dan tipe-tipe data *platform* yang spesifik menjadi bentuk XML.

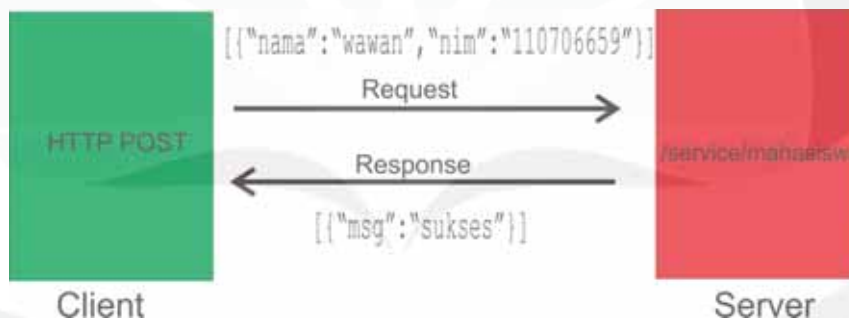
b. REST

REST merupakan singkatan dari *Representational State Transfer*. REST adalah teknik di arsitektur *software* untuk sistem terdistribusi seperti WWW (*World Wide Web*). REST tidak memerlukan *parsing* XML dan tidak memerlukan *header* pesan ke dan dari penyedia layanan. Hal tersebut dapat mengurangi penggunaan *bandwidth*. REST bekerja dengan bernavigasi melalui *link-link* HTTP untuk melakukan aktivitas tertentu seakan-akan terjadi perpindahan *state* satu dengan

yang lain. Perintah HTTP yang biasa digunakan adalah fungsi GET, POST, PUT, atau DELETE. Pada pengaplikasiannya, REST lebih banyak digunakan untuk *web service* yang berorientasi pada *resource*.

3.4 JSON (*Java Script Object Notation*)

JSON merupakan singkatan dari *Java Script Object Notation*. JSON adalah format pertukaran data yang ringan dan formatnya sederhana. JSON sangat mudah dimengerti dan diimplementasikan oleh manusia dan mudah bagi komputer untuk melakukan parsing data. Format JSON digunakan untuk mentransmisikan data yang terstruktur melalui koneksi jaringan pada proses yang disebut serialisasi (Kusumawaty, 2012).



Gambar 3.3 Penggunaan JSON REST

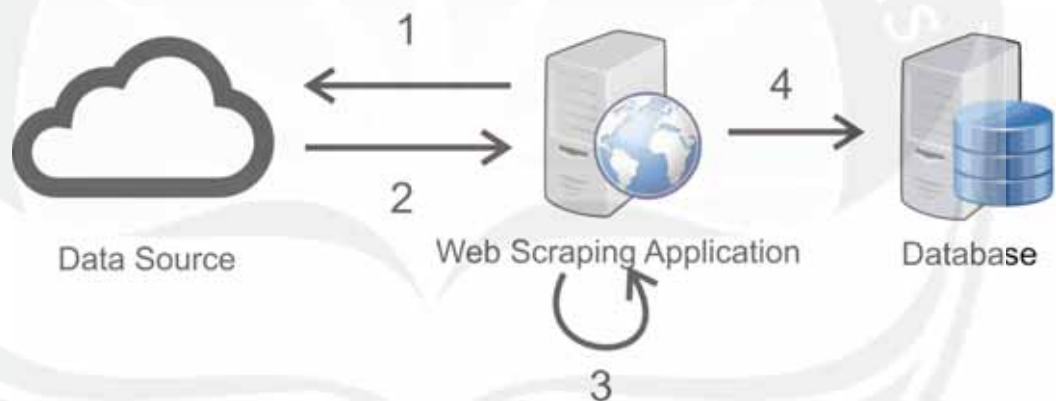
Gambar 3.3 adalah penggunaan JSON REST dengan method POST. Data yang dikirimkan oleh *client* adalah nama : wawan dan nim : 110706659. *Server* kemudian menjalankan fungsi `'/service/mahasiswa'` untuk memasukkan data mahasiswa ke dalam *database*. Fungsi tersebut kemudian akan me-*return*kan data `msg : sukses`.

3.5 Web scraping

Web scraping merupakan proses untuk mendapatkan sejumlah informasi untuk sebuah situs *website*. *Web*

scraping erat kaitannya dengan pengindeksan *website* yang merupakan teknik pencarian informasi yang mengadopsi teknik pada mesin pencarian.

Fokus dari *web scraping* adalah pada transformasi data yang tidak terstruktur pada *website*, biasanya dengan format HTML, menjadi data terstruktur yang bisa disimpan dan dianalisa dalam *database* (Vargiu & Urru, 2012). Metode pengambilan konten yang bisa digunakan adalah *fetching* dan *cURL*. Metode *cURL* dapat mengirim informasi lengkap seperti sebuah *web browser* sehingga membuat *web server* menganggap permintaan dilakukan oleh pengguna dengan menggunakan *web browser*. (Utomo, 2012).



Gambar 3.4 Proses Web Scraping

Gambar diatas merupakan contoh proses pengambilan data dengan *web scraping*. Langkah ke 1 adalah proses *request* dari aplikasi ke data yang ingin di ambil yaitu dari sebuah *website*. Proses ke 2 adalah mendapatkan data dari data *source* atau proses *response*. Data yang didapatkan kemudian di ekstrak sehingga menjadi data terstruktur pada proses ke 3. Ekstraksi data tersebut kemudian disimpan ke dalam *database* pada proses ke 4.

3.6 Automatic Text Summarization

Peringkasan teks otomatis adalah pembuatan bentuk yang lebih singkat dari sebuah teks secara otomatis dengan memanfaatkan aplikasi. Peringkasan teks secara otomatis sangat membantu dalam mengetahui inti dari sebuah teks atau banyak teks. Ringkasan merupakan teks yang dihasilkan dari sebuah teks atau banyak teks yang mengandung isi informasi dari teks asli dan panjangnya tidak lebih dari setengah teks aslinya (Hovy & Lin, 2001).

3.7 Metode TF-IDF

TF-IDF (*Term Frequency-Inverse Document Frequency*) merupakan metode yang digunakan untuk memberikan nilai pada kalimat yang dianggap sebagai inti dokumen (Evan, et al., 2014). Metode ini menggabungkan dua konsep untuk perhitungan bobot, yaitu *Term Frequency* (TF) merupakan frekuensi kemunculan kata (t) pada kalimat (d). *Document Frequency* (DF) adalah banyaknya kalimat dimana satu kata (t) muncul. Bobot hubungan antara sebuah kata dan sebuah dokumen akan tinggi apabila frekuensi kata tersebut tinggi di dalam dokumen dan frekuensi keseluruhan dokumen yang mengandung kata tersebut rendah pada kumpulan dokumen (Intan & Defeng, 2006). Langkah-langkah peringkasan Dokumen dengan Metode TF-IDF adalah sebagai berikut :

1. Dokumen dipecah menjadi kalimat-kalimat dan menghilangkan tanda hubung yang terdapat di masing-masing kalimat, seperti : titik, koma, tanda seru, tanda petik dan sebagainya. Kalimat-kalimat tersebut kemudian diubah menjadi *lowercase*.

2. Masing-masing kalimat kemudian di pecah menjadi kata-kata. Proses dari pemecahan kalimat menjadi kata diberikan filter kata sambung. Kemudian hitung nilai TF dari kata tersebut di dalam sebuah dokumen menggunakan persamaan $TF(t) = f_{t,d} / \sum t,d$ dimana $f_{t,d}$ merupakan frekuensi sebuah kata (t) muncul di dalam dokumen d , sedangkan $\sum t,d$ merupakan total keseluruhan kata yang terdapat di dalam dokumen d .
3. Setelah didapat nilai TF dari masing-masing kata, maka dapat dicari nilai IDF dari masing-masing kata tersebut dengan melihat pada seluruh dokumen dengan menggunakan persamaan $IDF(t) = \log \left(\frac{|D|}{f_{t,D}} \right)$ dimana $|D|$ merupakan jumlah dokumen yang ada di dalam koleksi, sedangkan $f_{t,D}$ merupakan jumlah dokumen dimana t muncul dalam D . IDF merupakan *Inverse Document Frequency*.
4. Masing-masing kata dihitung nilai TF-IDFnya dengan menggunakan persamaan $TF-IDF(t) = TF(t) * IDF(t)$. Dari hasil penghitungan TF-IDF didapatkan kata-kata yang mempunyai nilai berbeda, tergantung seringnya kata tersebut muncul di dokumen.
5. Kemudian lakukan *sentence scoring* dari sebuah dokumen dengan menggunakan nilai TF dari masing-masing kata. Kalimat yang mempunyai skor tertinggi dipilih untuk dijadikan ringkasan dari sebuah dokumen.
6. Ringkasan dari masing-masing dokumen tersebut digabungkan menjadi satu.
7. Supaya dapat terbentuk sebuah ringkasan akhir, dilakukan pemilihan kalimat dengan metode TF-IDF dan

sentence scoring sama seperti memilih kalimat yang akan digunakan sebagai ringkasan pada dokumen tunggal.

3.8 Cron Job

Cron job adalah sebuah *software* untuk penjadwalan yang memungkinkan mengatur tugas secara terjadwal dan berjalan pada *background* Linux server (digitalocean, 2013). Tugas yang biasanya dilakukan oleh *cron job* adalah mengeksekusi *script* per waktu yang telah ditentukan. *Cron job* akan berjalan secara otomatis pada server berdasarkan pengaturannya dan dapat ditentukan interval waktunya (IDWebhost, 2007).

Cron job mempunyai dua macam tampilan yaitu berbasis GUI (*Graphical User Interface*) dan berbasis *Command Line*. *Cron job* yang berbasis GUI (*Graphical User Interface*) biasanya sudah terdapat di *cpanel* hosting dan tinggal digunakan dengan pengaturan waktu dan *script* yang akan dijalankan. Salah satu contoh *command* untuk menjalankan *script* pada *cron job* adalah sebagai berikut :



Gambar 3.5 Contoh Operasi Cron Job

3.9 Aplikasi Mobile

Aplikasi *mobile* adalah perangkat lunak yang didesain untuk dapat dijalankan pada *device smartphone* dan perangkat *mobile* yang lainnya. Permintaan publik dan pertumbuhan aplikasi *mobile* yang meningkat pesat membuat cakupan dari aplikasi *mobile* semakin luas. Menurut (Lee & Scheel, 2004) Pengembangan aplikasi *mobile* dianggap lebih mudah dan lebih cepat.

Aplikasi *mobile* dibagi menjadi 3 yaitu *native application*, *web application*, dan *hybrid application*. Aplikasi *native* dikembangkan dengan bahasa pemrograman yang semestinya misalnya android menggunakan Java dan SDK milik Android, dan iOS menggunakan *Objective-C* dan SDK iOS. Aplikasi *native* tertanam atau diinstal pada *device*. *Mobile Web Application* adalah aplikasi *mobile* yang dijalankan menggunakan *browser* yang ada di handphone, menggunakan bahasa pemrograman *web*. *Hybrid Application* adalah aplikasi *mobile* yang tertanam atau diinstall pada *device* dan dibuat dengan menggunakan teknologi *web* digabung dengan teknologi *native application*.