

BAB II KERANGKA TEORITIS

A. Tinjauan Objek Penelitian

Penelitian “Penerapan *Web Service* Untuk Meningkatkan Performa Kecepatan Data Pada Sistem Tes Potensi Akademik” dilakukan di perguruan tinggi swasta di kota Bogor yaitu Universitas Binaniaga Indonesia. Universitas Binaniaga Indonesia adalah sebuah PTS (Perguruan Tinggi Swasta) yang berlokasi di Bogor yang merupakan penggabungan dari beberapa perguruan tinggi yang berada di bawah naungan Yayasan Binaniaga yakni STIKOM Binaniaga (berdiri pada tahun 1992), STIE Binaniaga (berdiri pada tahun 1993), dan AMIK Bogor (berdiri pada tahun 2004) yang telah disetujui oleh Kementerian Pendidikan dan Kebudayaan Republik Indonesia dengan SK No. 499/M/2020 tanggal 30 April 2020.

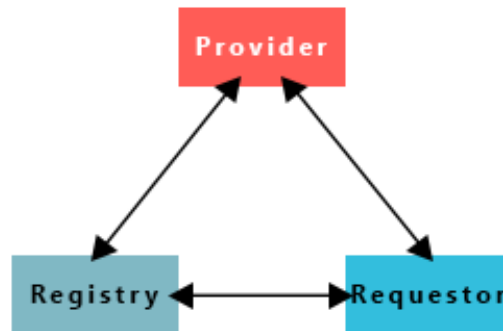
Dalam penerimaan mahasiswa baru, Universitas Binaniaga Indonesia mengadakan kegiatan tes potensi akademik. Kegiatan tes potensi akademik tersebut sudah dibantu dengan menggunakan sistem berbasis *desktop*. Sistem tersebut terkoneksi pada jaringan lokal yang terhubung ke *database* sistem TPA Universitas Binaniaga melalui perantara *middleware* *odbc*.

B. Landasan Teori

Dalam rangka untuk menyelesaikan suatu persoalan dalam penelitian ini, diperlukan landasan teori yang bersifat ilmiah. Dalam landasan teori ini akan dijelaskan kerangka-kerangka acuan komperhensif tentang prinsip, teori, maupun konsep yang digunakan sebagai landasan dalam menyelesaikan persoalan yang dihadapi. Berikut penjelasan teori yang berkaitan dalam penelitian ini:

1. Pengertian *Web Service*

Web service merupakan sebuah perangkat lunak yang dibangun untuk mendukung interaksi antara sistem yang satu dengan yang lainnya pada sebuah jaringan komputer. *Web service* digunakan untuk menyediakan layanan dalam bentuk informasi kepada sistem lain, sehingga sistem lain dapat berkomunikasi dengan *web service* melalui layanan yang disediakan. Dalam menyimpan data *web service* menggunakan format standar seperti HTTP, XML, SSL, SMTP, SOAP, dan JSON. Berikut adalah arsitektur *web service* secara umum pada Gambar 2.1: (Rofiq, dan Susanto, 2017)



Gambar 2.1 Struktur Web Service

2. Pengertian Web Server

Web server adalah perangkat lunak yang menjadi tulang punggung dari *www* (*World Wide Web*) di mana *web server* menerima permintaan dari *client* yang kemudian permintaan tersebut akan diproses dan kemudian *web server* akan menampilkan hasil proses tersebut berupa data yang diinginkan oleh *client*. Data yang ditampilkan memiliki format standar yang disebut dengan *Standard Generalized Markup Language* (SGML) namun beralih ke format *Hypertext Markup Language* (HTML) karena lebih sederhana digunakan dan dapat dipelajari dengan mudah. *Web server* menggunakan protokol yang disebut dengan *Hypertext Transfer Protocol* (HTTP) untuk berkomunikasi dengan *client*. (Nurmiati, 2012).

3. Pengertian JSON

JavaScript Object Notation (JSON) adalah format pertukaran data yang mudah dibaca dan ditulis oleh manusia, mudah diterjemahkan oleh mesin dan juga ringan. JSON menggunakan gaya bahasa yang umum dipakai oleh programmer dari keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python, dan lain-lain. Properti inilah yang membuat JSON tepat untuk format pertukaran data.

JSON memakai struktur seperti berikut:

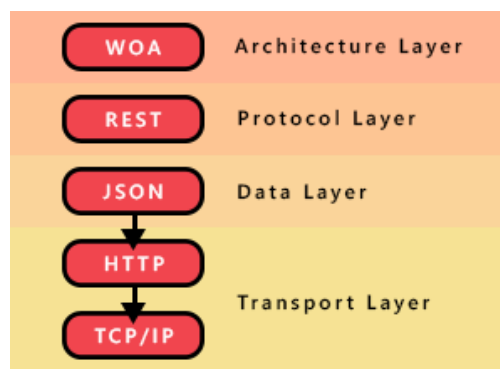
- a. *Object* (objek) adalah sepasang nama/nilai yang tidak berurut. Sebuah objek dimulai dengan tanda "{" (kurung kurawal buka) dan diakhiri dengan "}" (kurung kurawal tutup). Setiap nama objek diikuti dengan tanda : (titik dua) dan setiap pasangan nama/nilai dipisahkan dengan , (koma).
- b. *Array* adalah kumpulan nilai yang berurut. *Array* dimulai dengan "[" (kurung siku buka) dan diakhiri dengan "]" (kurung siku tutup). Nilai-nilai dipisahkan dengan tanda "," (koma).

- c. *Value* (nilai) dapat berupa *string* dalam tanda kutip dua (“) atau angka, atau *boolean* (*true/false*) atau *null*, atau sebuah *object* atau *array*. Struktur ini dapat disusun secara bertingkat.
- d. *String* adalah urutan dari karakter *Unicode* nol atau lebih yang dibungkus dengan tanda kutip dua (“), dan dapat menggunakan garis miring terbalik (\) untuk membuat karakter khusus.
- e. *Number* (angka) pada JSON serupa dengan *number* pada bahasa C dan Java, kecuali tidak menggunakan format *octal* dan *hexadecimal*.
- f. *Whitespace* (spasi kosong) dapat dimasukkan diantara pasangan tanda-tanda tersebut, kecuali beberapa detail *encoding* yang mendeskripsikan bahasa pemrograman yang bersangkutan. (<https://www.json.org/json-en.html>, diakses tanggal 11 Desember 2020).

4. Pengertian REST

Representational State Transfer (REST) adalah arsitektur baru untuk *web service* yang melibatkan aturan atau arsitektur beorientasi *resource* yang mendukung kesederhanaan dari teknologi *web*, menggunakan standar seperti *Hypertext Transfer Protocol* (HTTP), *Uniform Resource Identifier* (URI), dan *Extensible Markup Language* (XML). Arsitektur REST memiliki fitur unggul di antaranya:

- a. Data ditransmisikan dari *client* ke *server* melalui URI.
- b. Operasi yang dilakukan oleh *server* pada data dijabarkan dengan metode HTTP secara langsung.
- c. Apapun yang bisa ditandai sebagai *object* (konkrit atau abstrak) akan secara otomatis menjadi *resource* (sumber daya).
- d. URI untuk setiap sumber daya mengandung nama dan alamat. Umumnya URI ini identik dengan *Uniform Resource Locator* (URL).
- e. Inti dari sumber daya tidak menggambarkan data, tetapi hanya sebuah ide pada struktur dari *data service*. (Dospinescu, dan Perca, 2013).



Gambar 2.2 Lapisan REST

5. Pengertian PHP

PHP merupakan bahasa yang dirancang untuk membuat konten HTML. Rasmus Lerdort pertama kali menyusun PHP pada tahun 1994. Awal mulanya PHP dirancang untuk membuat konten *web* dinamis. Bahasa PHP dipengaruhi oleh bahasa pemrograman yang lain seperti Perl dan C. PHP dapat digunakan dengan tiga cara utama seperti:

- a. *Server-side scripting* – PHP dibuat untuk mengembangkan konten *web* yang bersifat dinamis. Diperlukan PHP *parser* dan *web server* untuk membuat HTML yang digunakan untuk mengirimkan dokumen kode.
- b. *Command-line scripting* – PHP dapat menjalankan *script* dari *command line* sama halnya dengan Perl, AWK, atau Unix shell.
- c. *Client-side GUI applications* – Dengan PHP-GTK dapat membuat aplikasi antar platform berbasis GUI. (Tatroe, dkk, 2013).

6. Pengertian JavaScript

Pada tahun 1995 JavaScript adalah bahasa pemrograman yang pertama kali diperkenalkan sebagai cara untuk menambahkan program ke situs halaman *web* pada *browser* Netscape Navigator. JavaScript memungkinkan aplikasi *web* untuk berinteraksi langsung tanpa harus memuat ulang halaman untuk setiap kegiatan. JavaScript juga digunakan untuk menyediakan berbagai bentuk interaksi. Setelah diadopsi di luar Netscape, sebuah standar dokumen ditulis untuk mendeskripsikan cara kerja JavaScript yang disebut standar ECMAScript. (Haverbeke, 2018).

7. Pengertian jQuery

jQuery merupakan sebuah *library* yang menyediakan kegunaan umum lapisan abstraksi untuk *scripting* umum *website*. Fitur utama dari jQuery adalah:

- a. **Mengakses bagian dari sebuah halaman.** Tanpa *library* JavaScript, banyak baris kode yang harus ditulis untuk melintasi pohon **Document Object Model (DOM)**, dan melacak porsi spesifik dari struktur dokumen HTML.
- b. **Memodifikasi tampilan sebuah halaman.** jQuery dapat mengubah *class* atau *style property* individual yang diterapkan ke bagian dari dokumen bahkan setelah halaman selesai dirender.
- c. **Mengambil informasi dari server tanpa me-refresh halaman.** Pola kode ini biasa dikenal dengan **Asynchronous JavaScript and XML**

(AJAX), dan mempermudah pengembang *web* dalam membuat halaman yang responsif, dan kaya akan fitur. (Chaffer dan Swedberg, 2007).

8. Pengertian MySQL

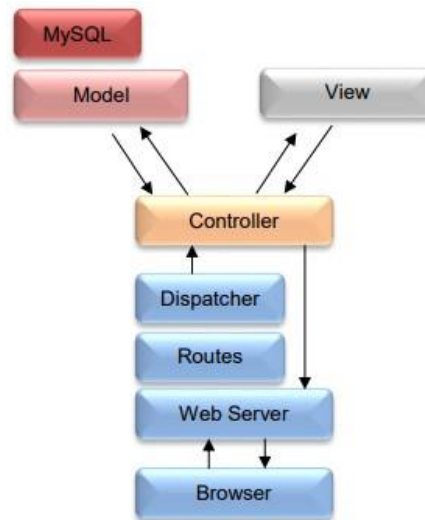
MySQL (dieja My-Ess-Que-Ell) adalah sebuah *Relational Database Management System* (sistem manajemen *database* relasional) yang cepat dan kuat. MySQL memungkinkan pengguna untuk menyimpan data, mencari data, menyortir dan mengambil data dengan efisien pada *database*. MySQL mengontrol data untuk memastikan banyak pengguna bisa bekerja secara bersamaan untuk menyediakan akses secara cepat dan untuk memastikan hanya pengguna yang diizinkan yang dapat memperoleh akses, maka dari itu MySQL itu *multiuser, multithreaded server*. MySQL menggunakan bahasa *query* standar yakni *Structured Query Language* (SQL). (Welling, dan Thomson, 2008).

9. Pengertian Apache Web Server

Apache *Web Server* adalah *web server* yang berbasis *open source* dengan basis *unix*. Awal mulanya Apache dikembangkan berbasis kode pada NCSA HTTPD 1.3, lalu dirancang ulang menjadi *web server* yang pada saat ini banyak dipakai. Apache merupakan *web server* terpopuler dan banyak digunakan lebih dari 42% dari berbagai domain *website*. Apache memiliki fitur seperti performa, fungsionalitas, efisiensi, serta kecepatan yang tinggi. (Chandra, 2019)

10. Pengertian CodeIgniter

CodeIgniter adalah sebuah kerangka (*framework*) *web application* yang berbasis *open source* yang digunakan untuk mengembangkan aplikasi php yang bersifat dinamis. Tujuan dari dikembangkannya CodeIgniter adalah untuk membantu pengembang aplikasi untuk membangun aplikasi lebih cepat. CodeIgniter menggunakan konsep pola pengembangan *Model-View-Controller* (MVC). Cara kerja pada CodeIgniter yakni *web browser* berinteraksi melalui *controller*. *Controller* kemudian akan menerima dan membalas semua permintaan dari *web browser*. *Controller* akan meminta data ke *Model* jika *controller* memerlukan data. Untuk tampilan ke *user* akan ditangani oleh *View*. Hal ini ditunjukkan pada Gambar 2.3: (Subari, dkk, 2018).

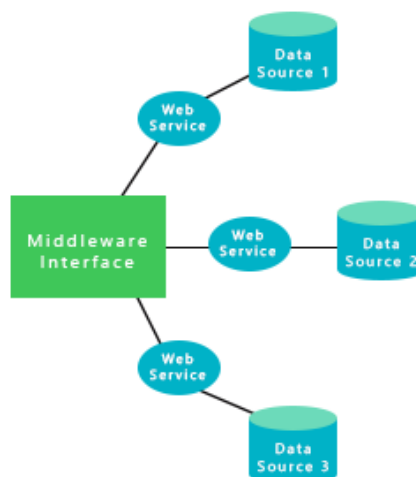


Gambar 2.3 Alur Data CodeIgniter

C. Subjudul Teoritis

1. Pengertian Middleware

Middleware merupakan perangkat lunak komputer yang memberikan layanan untuk menghubungkan bagian-bagian yang berbeda dari sebuah aplikasi. *Middleware* pada umumnya dipakai untuk sistem terdistribusi dengan tujuan untuk mempermudah pengembangan perangkat lunak. Berikut adalah struktur *middleware* pada Gambar 2.4: (Hasibuan, dkk, 2020).



Gambar 2.4 Struktur Middleware

2. Pengertian Response Time

Menurut kamus (Rigdon, 2016) Definisi dari *response time* (waktu respon) ialah jumlah waktu yang dibutuhkan oleh program untuk melakukan suatu pekerjaan mulai dari awal sampai dengan selesai. Dalam lingkungan *client/server*, pengukuran waktu ini dilakukan dari sisi *client*.

3. Pengertian Performance

Menurut kamus (Rigdon, 2016) *Performance* (performa) adalah sebuah istilah yang digunakan untuk mengetahui sejauh mana suatu produk atau layanan dapat menjalankan fungsinya yang sudah ditentukan.

4. Pengertian Tes Potensi Akademik

Tes potensi adalah evaluasi terhadap kemampuan potensial seseorang yang dibuat spesifik untuk memperkirakan peluang kesuksesan untuk belajar di suatu perguruan tinggi. Tes potensi akademik menggunakan konsep *Graduate Record Examinations* (GRE) yang terdiri dari bagian *Verbal Reasoning* (V), *Quantitative Reasoning* (Q), dan *Analytical Writing* (AW). Di Indonesia tes potensi akademik terdiri dari tiga tahap tes yaitu tes verbal, tes kuantitatif, dan tes penalaran. Tes potensi akademik tidak disusun berdasarkan silabus mata pelajaran, karena itu kesuksesan dalam menjawab soal tes ini lebih bergantung pada penguasaan materi tertentu. Hal ini dikarenakan soal-soal pada tes potensi dikembangkan sedemikian rupa untuk dijawab berdasarkan pada daya penalaran (*reasoning*) baik logis (*logical*) maupun analitis (*analytical*). (Azwar, 2008).

D. Subjudul Permasalahan

1. Penyebab Lambatnya Kinerja Aplikasi

Menurut artikel yang dipublikasikan oleh riverbed.com (2017) ada 5 penyebab kenapa aplikasi berjalan dengan lambat:

- a. ***Sluggish client* (Klien yang lamban)** – Aplikasi berbasis *web* saat ini cenderung untuk mendorong kerja interaksi pengguna yang seringkali disertai dengan banyak data ke *client*. Dari situ, kode JavaScript memproses ratusan atau ribuan kolom data yang bisa menyebabkan jeda beberapa detik sebelum menampilkan pembaruan pada *client*.
- b. ***Slow server* (Server yang lambat)** – Penyebab paling umum dari lambatnya performa aplikasi adalah dari aplikasi atau dari *server* itu sendiri. Aplikasi modern biasanya menggunakan infrastruktur *multi-tier*.

- (1) *Front-end web server* berinteraksi dengan aplikasi *server* yang berinteraksi dengan *middleware server* yang mengakses satu atau beberapa *database server*.
 - (2) *Server* tersebut mungkin berinteraksi dengan *DNS server* untuk mencari alamat IP atau memetakannya kembali ke *server names*.
- Jika terjadi satu *link* yang lemah maka akan memperlambat seluruh aplikasi.
- c. ***Diminutive databases (Database yang kecil)*** – Aplikasi yang dikembangkan di jaringan LAN dengan jumlah data yang kecil terlihat akan beroperasi dengan lancar di lab. Tetapi ketika sudah masuk tahap produksi, semuanya akan berubah menjadi sebaliknya. *Database* akan terus bertumbuh dan begitu juga *downtime*-nya.
 - d. ***Chatty conversations (Percakapan yang cerewet)*** – Masalah umum lainnya yang menyebabkan lambatnya aplikasi adalah *chatty conversations*. *Server* atau mungkin *client* itu sendiri akan membuat banyak permintaan kecil untuk mengeksekusi sebuah transaksi atas nama orang yang menjalankan aplikasi.
 - e. ***Slow network services (Layanan jaringan yang lambat)*** – Terakhir, lambatnya layanan jaringan bisa membuat kinerja aplikasi menjadi lambat yang tidak melibatkan jaringan itu sendiri, tetapi layanan yang paling bergantung pada aplikasi berbasis jaringan. Bayangkan sebuah aplikasi yang membuat *query* ke *DNS server* utama yang tidak ada. Dengan tidak adanya respon, aplikasi akan kehabisan waktu pada permintaan pertama sebelum melakukan permintaan ke *DNS server* yang kedua. Pada situasi itu aplikasi akan melambat secara perlahan, tapi akan tetap berjalan dengan baik.

E. Tinjauan Pustaka

Ada beberapa penelitian yang sebelumnya sudah dilakukan yang dapat dijadikan sebagai acuan untuk penelitian yang akan dilakukan. Acuan yang dimaksud antara lain:

1. **“Penerapan Web Service (XML dan JSON) Untuk Meningkatkan Performance Pada Informasi Pembayaran Uang Kuliah”**, Program Studi Teknik Informatika, STIKOM Binaniaga Bogor. Oleh Anggra Triawan dan Muhammad Adi Prasetyo (2018)

Pada penelitian ini permasalahan yang dibahas adalah diperlukan sebuah sistem yang dipakai oleh BAUM (Biro Administrasi Umum) untuk mempercepat kegiatan dalam pembayaran uang kuliah. Ada 2 pilihan sistem

yang nantinya akan dipakai untuk staf keuangan yang dilihat dari kinerja yang lebih cepat antara XML dan JSON. Dari hasil penelitian yang dilakukan, disimpulkan bahwa:

- a. *Web service* mempercepat menampilkan menu pada sistem pembayaran uang kuliah
- b. Hasil kecepatan XML untuk *localhost* mempunyai rata-rata waktu 1,143533793363636 detik, sedangkan yang sudah di-*hosting* 0,0145821137818182 detik. Kecepatan JSON untuk *localhost* mempunyai rata-rata waktu 1,113626610136364 detik, sedangkan yang sudah di-*hosting* 0,0088517232272727 detik.

2. **“ANALISIS PERBANDINGAN PERFORMA WEB SERVICE REST MENGGUNAKAN FRAMEWORK LARAVEL, DJANGO DAN RUBY ON RAILS UNTUK AKSES DATA DENGAN APLIKASI MOBILE”**, Magister Teknik Informatika, Universitas AMIKOM Yogyakarta, Fakultas Ilmu Komputer, Universitas Indonesia. Oleh Danandjaya Saputra dan Rizal Fathoni Aji (2018).

Pada penelitian ini masalah yang dibahas adalah kampus STT Indonesia Tanjungpinang sudah memiliki sistem informasi akademik berbasis *web* bernama Portal E-Kampus. Portal E-Kampus di STT Indonesia Tanjungpinang akan dikembangkan ke aplikasi *mobile*. Dalam proses integrasi antara sistem yang sudah ada dengan sistem yang baru yang menggunakan platform *mobile* diperlukan penggunaan *web service*. *Web service* dapat dibuat dengan menggunakan *web framework* yang berbeda, pemilihan *web framework* dapat berpengaruh pada kinerja *web service*, *web framework* yang dimaksud adalah Laravel, Ruby on Rails dan Django. Berdasarkan penelitian tersebut, terdapat kesimpulan bahwa:

- a. Analisis perbandingan dilakukan dengan cara memanggil proses-proses *web service* yang dibuat menggunakan Laravel, Ruby on Rails dan Django melalui aplikasi *mobile* yang dihubungkan dengan jaringan lokal.
- b. Setiap proses pada *web service* yang di-*request* melalui aplikasi *mobile* dicatat *response time*, CPU *usage* dan RAM *usage*-nya untuk menentukan *web service* dengan performa terbaik.
- c. *Web service* menggunakan *framework* Django diprediksi memiliki waktu respon yang cepat, penggunaan CPU dan memori yang paling sedikit sehingga direkomendasikan kepada pihak kampus untuk membangun *web service* dengan menggunakan *framework* Django.

3. **“WEB SERVICE FOR SUPPORTING OF WEBSITE PERFORMANCE”**, Jurusan Teknik Informatika, Sekolah Tinggi Teknologi Adisutjipto (STTA) Yogyakarta. Oleh Sinta Tri Pamungkas, Agus Basukesti, dan Dwi Nugraheny (2012).

Pada penelitian ini masalah yang dibahas adalah diperlukan suatu sarana komunikasi yang efisien serta efektif antar sistem serta antar *platform*, maka dari itu dikembangkanlah *web service* untuk memenuhi kebutuhan tersebut. *Web service* dikembangkan sebagai sarana supaya pemilik suatu *website/blog* yang berjumlah banyak dapat terkoordinasi dengan baik. Hasil dari penelitian yang sudah dilakukan antar lain:

- a. *Web service* dapat dibangun untuk mendukung kinerja *website*.
- b. Mempermudah dalam pengelolaan data komentar yang berjumlah banyak.

4. **“Penerapan Teknologi Web Service Untuk Integrasi Layanan Puskesmas dan Rumah Sakit”**, Program Studi Ilmu Komputer, Universitas Gadjah Mada. Oleh Rokhmat Hidayat dan Ahmad Ashari (2013).

Pada penelitian ini masalah yang dibahas adalah dibutuhkannya sistem informasi yang saling mendukung dan saling terkait sehingga setiap kegiatan dan program kesehatan yang dilaksanakan dan dirasakan oleh masyarakat dapat diketahui, dipahami, diantisipasi dan dikelola dengan sebaik-baiknya. Maka dari itu dibutuhkan sekali dibangunnya sistem informasi kesehatan yang terintegrasi baik di dalam sektor kesehatan (antar program dan antar jenjang), dan di luar sektor kesehatan yakni dengan sistem jaringan informasi rumah sakit pemberi rujukan maupun rumah sakit rujukan. Hasil dari penelitian yang sudah dilakukan antara lain:

- a. *Prototype* sistem berhasil dibangun berupa integrasi sistem layanan informasi kesehatan berbasis *web service* dapat yang dapat menyatukan beberapa sistem yang berbeda dari segi bahasa pemrograman (ASP.NET, PHP) dan *database*-nya (SQL Server, MySQL).
- b. *Prototype* sistem yang dikembangkan dapat memberikan informasi yang terintegrasi melalui sebuah aplikasi berbasis *web* yang dikembangkan dengan menggunakan PHP sebagai *client* dari dua *web service* yang ada.
- c. Implementasi teknologi *web service* yang berfungsi sebagai *middleware* mampu melakukan pertukaran pesan dengan memanfaatkan protokol HTTP melalui sebuah jaringan antara aplikasi dan *database*.

5. **“PEMANFAATAN TEKNOLOGI WEB SERVICES PADA PERTUKARAN DATA KATALOG ANTAR PERPUSTAKAAN”**, Jurusan Teknik Informatika, Universitas Kristen Krida Wacana. Oleh Endah Kristiani (2012).

Pada penelitian ini masalah yang dibahas adalah kemampuan untuk mengintegrasikan aplikasi yang berbeda di antara perpustakaan dalam jaringan kerjasama. Maka dari itu diperlukan teknologi *web service* dimana *web service* dapat mengintegrasikan aplikasi, bahasa pemrograman dan platform yang berbeda sehingga aplikasi-aplikasi tersebut dapat saling berkomunikasi dan bertukar data. Hasil penelitian yang dilakukan dapat disimpulkan bahwa:

- a. Penerapan *web service* dikembangkan dengan menggunakan PHP NuSOAP untuk bertukar pesan SOAP.
- b. *Web service* dapat menjembatani pertukaran data antar perpustakaan yang berbeda sistem dan *database*-nya.
- c. Simulasi program data katalog perpustakaan A dengan *database* MySQL dan data katalog perpustakaan B dengan *database* Visual Foxpro serta konektor ODBC membuktikan bahwa katalog kedua perpustakaan ini dapat diintegrasikan.
- d. Hasil tes fungsionalitas menggunakan *tool* JMeter membuktikan bahwa fungsionalitasnya berfungsi dengan baik.
- e. Hasil tes performa menggunakan *tool* JMeter menunjukkan bahwa peningkatan jumlah *client* mempunyai *nonlinear impact* pada waktu respon *server* sehingga dapat dikatakan bahwa meningkatkan jumlah *client* 2 atau 3 kali tidak mempunyai pengaruh yang signifikan terhadap respon *server*.

6. **“PEMANFAATAN REPRESENTATIONAL STATE TRANSFER DAN JSON WEB TOKEN PADA WEB SERVICE UNTUK PENINGKATAN KINERJA DAN KEAMANAN DATA PADA APLIKASI E-COMMERCE”**, Program Studi Teknik Informatika, Universitas 17 Agustus 1945 Surabaya Oleh Yusril Fadillah dan Agus Hermanto (2020).

Pada penelitian ini masalah yang dibahas adalah industri kesehatan di bida pemasaran ingin meningkatkan proses bisnisnya dengan membangun sebuah aplikasi *e-commerce* berbasis *mobile*. Penelitian ini menerapkan *web service* dengan menggunakan arsitektur REST (*Representational State Transfer*) dengan mengimplementasikan RESTful API supaya dapat mengkomunikasikan data dari dua *platform* yang berbeda. Kesimpulan dari penelitian yang sudah dilakukan adalah sebagai berikut:

- a. Kinerja aplikasi *m-commerce* meningkat dengan menggunakan arsitektur RESTful API.
- b. Berdasarkan hasil implementasi arsitektur RESTful API, pada proses *login* yang membutuhkan kode OTP untuk memverifikasi nomor HP menghasilkan sebuah *response* JSON berupa kode token.

7. **“Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS”**, Teknik Elektro, Politeknik Negeri Ujung Pandang. Oleh Ahsan Mubariz, Dahlia Nur, Eddy Tungadi, dan Muhammad Nur Yasir Utomo (2020).

Pada penelitian masalah yang dibahas adalah sistem ujian masuk Politeknik Negeri Ujung Pandang memiliki kendala pada waktu respon saat banyaknya akses secara bersamaan. Untuk itu sistem dibangun ulang menggunakan arsitektur REST dan platform Node.JS. Node.JS memiliki keunggulan pada teknik *non-blocking* yang memungkinkan operasi-operasi dijalankan secara paralel, sehingga memungkinkan banyak *request* dapat diselesaikan secara paralel. Dari penelitian yang dilakukan dapat disimpulkan bahwa:

- a. Penelitian tersebut berhasil membangun *back-end server* untuk sistem ujian masuk Politeknik Negeri Ujung Pandang dengan kinerja dan fungsionalitas yang baik.
- b. Rata-rata waktu respon *server* meningkat seiring jumlah *virtual user* yang mengakses sistem secara bersamaan diperbesar dengan rata-rata peningkatan sebesar 52% pada skenario yang diujikan.
- c. Sistem yang dibuat memiliki kinerja yang lebih baik dibandingkan sistem sebelumnya berdasarkan hasil uji dengan rata-rata selisih waktu respon sebesar 3222,5 ms.

8. **“APLIKASI PENCARIAN PENJUALAN LAPTOP MENGGUNAKAN TEKNOLOGI WEB SCRAPING”**, Jurusan Teknologi Informasi, Politeknik Negeri Malang. Oleh Yan Watequlis Syaifudin, Arida Ferti Syafiandini, dan Hafiz Rizqi Prasadana (2018).

Pada penelitian ini masalah yang dibahas adalah pembeli laptop kerap kesulitan dalam memilih barang yang dicari dikarenakan telah banyak *website e-commerce* yang dibangun dan digunakan oleh para penjual laptop. Maka dari itu dibutuhkan sebuah aplikasi yang dapat memudahkan orang dalam pencarian barang dengan menggunakan teknologi *web scraping* dalam pencarian objek laptop yang berbasis *web* dan dengan menerapkan arsitektur

Representational State Transfer (REST). Setelah dilakukan penelitian dan pengembangan, maka disimpulkan dari penelitian ini bahwa:

- a. Metode *web scraping* digunakan untuk mendapatkan data harga barang, data *rating* barang, nama barang dan gambar barang.
- b. Pemanfaatan API Yahoo Service untuk mendapatkan data barang berupa laptop yang kemudian diolah agar sesuai dengan kebutuhan sistem yang dibangun.
- c. Dalam proses penyajian data pada aplikasi dengan *website e-commerce* yang digunakan, perlu didefinisikan kategori secara spesifik untuk hasil pencarian di *website e-commerce* yang digunakan.

9. **“Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token”**, Program Studi Teknik Informatika, Universitas Komputer Indonesia. Oleh Muhammad Iqbal Perkasa, Eko Budi Setiawan (2018).

Pada penelitian ini masalah yang dibahas adalah pemerintah belum menggunakan data *public* secara maksimal untuk keperluan administrasi. Pemanfaatan data dengan populasi besar ini yang menjadi tujuan dibangunnya sistem aplikasi *web service* dengan REST API di mana data ini akan terbuka dan dapat diakses kepada siapa yang memiliki akses. Salah satu instansi yang menggunakan layanan ini adalah Dinas Ketenagakerjaan dan Transmigrasi di mana sistem ini dapat membuat staf dinas lebih efisien dalam membuat dan mendaftarkan kartu pencarian kerja menggunakan data yang tersedia. Dari hasil penelitian yang dilakukan, dapat ditarik kesimpulannya sebagai berikut:

- a. *Web service* mampu dan dapat mempercepat pendaftaran dengan keandalan *server* yang baik.
- b. Administrator data penduduk dapat dimudahkan dalam memonitor penggunaan data penduduk dan mengatur hak akses masing-masing *access token*.

10. **“Analisis Perbandingan Performa Web Service Menggunakan Bahasa Pemrograman Python, PHP, dan Perl pada Client Berbasis Android”**, Program Studi Teknik Informatika, Universitas Brawijaya. Oleh Achmad Fauzi Harismawan, Agi Putra Kharisma, Tri Afirianto (2018).

Pada penelitian ini masalah yang dibahas adalah efisiensi dalam penggunaan sumber daya dalam pengembangan aplikasi perangkat bergerak karena perangkat bergerak sangatlah sensitif terhadap *delay*. Salah satu faktor yang menyebabkan *delay* adalah proses yang dilakukan pada *web*

service. Kecepatan proses *web service* bergantung pada bahasa pemrograman yang digunakan, bahasa pemrograman juga dapat mempengaruhi penggunaan CPU dan *memory* pada *server*. Oleh karenanya perlu dilakukan analisis performa *web service* untuk mengetahui perbedaan penggunaan CPU, *memory*, dan kecepatan eksekusi bahasa pemrograman yang digunakan pada *web service*. Dari hasil penelitian yang dilakukan dapat ditarik kesimpulannya sebagai berikut:

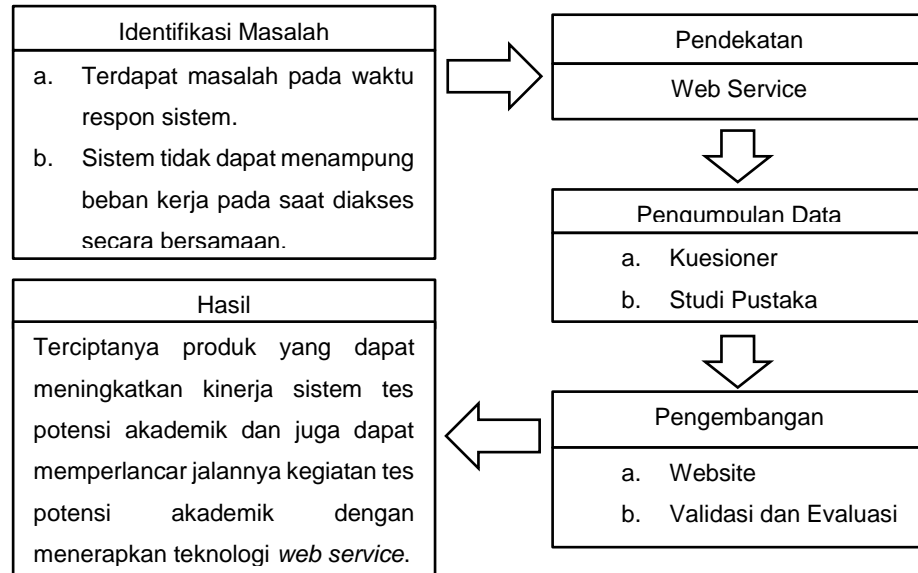
- a. Analisis dilakukan dengan cara menjalankan operasi *string concatenation* pada bahasa pemrograman Python, PHP, dan Perl dan diimplementasikan pada jaringan lokal.
- b. Setelah dilakukan pengujian data sebanyak 5.000 dan 20.000 data, bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat, sedangkan bahasa pemrograman Python memiliki penggunaan CPU dan *memory* paling sedikit.
- c. Pada pengujian dengan menggunakan 10.000 dan 15.000 data, bahasa pemrograman Python memiliki penggunaan *memory* paling sedikit.
- d. Pada pengujian dengan menggunakan 40.000 data, bahasa pemrograman Perl memiliki kecepatan eksekusi paling cepat dan penggunaan CPU serta *memory* paling sedikit.

F. Kerangka Pemikiran

Berdasarkan landasan teori yang didapat dari pencarian teori yang dijadikan rujukan penelitian, maka dapat disusun kerangka pemikiran sebagai berikut:

1. Identifikasi masalah untuk menetapkan tujuan penelitian.
2. Melakukan penerapan *web service* pada permasalahan yang sudah teridentifikasi.
3. Melakukan pengumpulan data berdasarkan setiap pengguna sistem.
4. Melakukan pengembangan melalui tahap perancangan, tahap implementasi, dan tahap pengujian terhadap sistem sehingga bisa menarik kesimpulan dari penelitian yang dilakukan.

5. Melakukan evaluasi pada sistem yang dikembangkan sehingga bisa diketahui kekurangannya untuk dijadikan bahan evaluasi.



Gambar 2.5 Kerangka Pemikiran

Gambar 2.5 menerangkan kerangka pemikiran mulai dari identifikasi masalah, metode pendekatan yang dilakukan, teknik pengumpulan data, pengembangan, sampai hasil yang diperoleh.

G. Hipotesis Penelitian

Berdasarkan kerangka berpikir dan penelitian teoritis serta rujukan penelitian yang diambil, penelitian ini menetapkan hipotesis penelitian bahwa penerapan *web service* diduga menjadi solusi untuk meningkatkan performa pada sistem tes potensi akademik.