

RANCANG BANGUN SISTEM INFORMASI PERPUSTAKAAN MENGGUNAKAN WEB SERVICE PADA JURUSAN TEKNIK KOMPUTER POLSRI

Ali Firdaus¹, Slamet Widodo², Adi Sutrisman³, Sutan Gading Fadhillah Nasution⁴, Rina Mardiana⁵
Jurusan Teknik Komputer, Politeknik Negeri Sriwijaya Palembang

ABSTRAK

Penelitian ini berjudul “Rancang Bangun Sistem Informasi Perpustakaan Menggunakan Web Service Pada Jurusan Teknik Komputer POLSRI”. Sistem informasi yang dibangun ini menggunakan *web service* berbasis REST. Perancangan web service ini menggunakan bahasa pemrograman JavaScript dengan menggunakan *runtimenode.js* untuk berjalan di sisi *server*. Sistem informasi ini menggunakan MongoDB sebagai media penyimpanan database. REST Server ini mempunyai API (*Application Programming Interface*) *endpoint* atau alamat yang berupa URI (*Uniform Resource Identifier*) yang menerima *request* dari *requestor* melalui protokol HTTP dengan berbagai *method* (*PUT, GET, POST, DELETE*). REST Server ini kemudian memproses operasi *database* yang umumnya disebut (*CREATE, READ, UPDATE, DELETE*) dan kemudian mengirim *response* ke *requestor* berupa *HTTP response code* sebagai tanda atau status *request* beserta data dengan format JSON (*JavaScript Object Notation*) yang akan dapat dibaca oleh berbagai bahasa pemrograman. REST Server ini juga dilengkapi dengan sistem koneksi *real-time* dan *push notification* *semmitter* untuk mem-*broadcast* pesan ke semua pengguna aplikasi yang dituju.

Kata Kunci— Sistem Informasi, API, RESTful Web Service, MongoDB, Node.js

I. PENDAHULUAN

1.1 Latar Belakang

Sejauh ini perpustakaan di jurusan Teknik Komputer POLSRI masih menyimpan data pembukuan secara manual. Seiring teknologi semakin berkembang sangat pesat terutama dalam pengembangan Aplikasi, diharapkan perpustakaan Teknik Komputer memiliki aplikasi khusus yang terhubung ke internet dengan tujuan mempermudah mahasiswa atau dosen Teknik Komputer untuk mengakses perpustakaan dari jarak jauh secara online serta memudahkan admin perpustakaan dalam melaksanakan tugas administratifnya. Akan tetapi, teknologi akan terus semakin berkembang. Aplikasi yang sudah dibuat tidak akan bertahan untuk jangka panjang. Akan ada teknologi baru yang bermunculan untuk waktu kedepannya. Untuk menghindari pembuatan ulang data setiap membuat dan mengembangkan aplikasi terbaru, maka Web Service solusinya.

Dengan begitu, ketika akan membuat dan mengembangkan aplikasi perpustakaan terbaru untuk masa kedepannya, tidak lagi diperlukan untuk membuat ulang struktur dan metode pengaksesan

sistem informasinya. Hanya saja diperlukan untuk mengembangkan aplikasi dari segi tampilan dan fitur, sedangkan aplikasi itu akan tetap mengakses data yang sama dari server yang memiliki layanan Web Service. Tidak hanya itu, Web Service juga memiliki peluang besar untuk menyelesaikan masalah dalam pertukaran data, integrasi data dan pengelolaan data. Service yang dibangun dapat dikembangkan atau digunakan oleh beberapa aplikasi dengan platform yang berbeda. Jadi, sistem informasi perpustakaan akan bisa di akses oleh aplikasi atau service lain yang membutuhkan informasi tersebut guna untuk menciptakan sistem yang terintegrasi.

1.2 Tujuan

Adapun tujuan dari *Web Service* ini adalah membuat sistem informasi yang flexible, fungsional dan bisa terintegrasi dengan aplikasi lainnya yang membutuhkan data dari Perpustakaan di jurusan Teknik Komputer.

1.3 Manfaat

Adapun manfaat dibuatnya Web Service ini adalah :

1. Sistem informasi tidak hanya dapat diakses oleh aplikasi Perpustakaan Online di jurusan Teknik Komputer saja, tetapi juga dapat diakses oleh berbagai aplikasi lainnya dari berbagai platform yang membutuhkan sistem informasi Perpustakaan di jurusan Teknik Komputer

2. Sistem yang akan dibangun bersifat flexible, integratable dan maintainable

II. TINJAUAN PUSTAKA

2.1 Web Service

W3C mendefinisikan web service sebagai sebuah sistem perangkat lunak yang dirancang untuk mendukung komunikasi dan interaksi antar mesin ke mesin (Machine to Machine) melalui sebuah network (jaringan). Web Service juga termasuk WebAPIs yang dapat diakses melalui jaringan seperti misalnya internet, dan dieksekusi melalui sebuah sistem jarak jauh sesuai dengan layanan yang diminta. Definisi Web Service menurut W3C juga meliputi banyak sistem berbeda, tetapi pada umumnya lebih menyangkut pada client dan server yang berkomunikasi menggunakan XML yang memenuhi standar SOAP (Simple Object Access Protocol). Asumsi secara umum adalah pada terminologi terdapat deskripsi dari mesin yang layanannya disediakan oleh server, atau sama seperti konsep dari WSDL. WSDL bukan termasuk standard dari SOAP tetapi merupakan syarat mutlak untuk client-side otomatis pada framework Java dan .NET SOAP. Beberapa organisasi industri seperti WS-I mengklaim baik SOAP dan WSDL sebagai definisi sari Web Service.

Selain SOAP dengan XML nya terdapat jenis engine web service lainnya yang banyak diimplementasikan pada aplikasi web, yaitu REST. REST web service atau yang kadang disebut RESTful web service atau RESTful API adalah web service yang mengimplementasikan arsitektur REST. Pada arsitektur REST, setiap service atau layanan dipandang sebagai sebuah Resources yang diidentifikasi melalui URL. (Abdul dkk, 2013:2).

2.2 API (Application Programming Interface)

API merupakan software interface yang terdiri atas kumpulan instruksi yang disimpan dalam bentuk library dan menjelaskan bagaimana agar suatu software dapat berinteraksi dengan software lain. Penjelasan ini dapat dicontohkan dengan analogi apabila akan dibangun suatu rumah. Dengan menyewa kontraktor yang dapat menangani bagian yang berbeda, pemilik rumah dapat memberikan tugas yang perlu dilakukan oleh kontraktor tanpa harus mengetahui bagaimana cara kontraktor menyelesaikan pekerjaan tersebut. Dari analogi tersebut, rumah merupakan software yang akan dibuat, dan kontraktor merupakan API yang mengerjakan bagian tertentu dari software tersebut tanpa harus diketahui bagaimana prosedur dalam melakukan pekerjaan tersebut. (Webber dkk, 2010:3).

2.3 REST (REpresentational State Transfer)

REST (REpresentational State Transfer) adalah suatu arsitektur metode komunikasi yang sering diterapkan dalam pengembangan layanan berbasis web. Arsitektur REST yang umumnya dijalankan via HTTP (Hypertext Transfer Protocol), melibatkan proses pembacaan laman web tertentu yang memuat sebuah file XML atau JSON. File inilah yang menguraikan dan memuat konten yang hendak disajikan. Setelah melalui sebuah proses definisi tertentu, konsumen akan bisa mengakses antarmuka aplikasi yang dimaksudkan.

Kekhasan REST terletak pada interaksi antara klien dan server yang difasilitasi oleh sejumlah tipe operasional (verba) dan Universal Resource Identifiers (URIs) yang unik bagi tiap-tiap sumberdaya. Masing-masing verba GET, POST, PUT dan DELETE memiliki makna operasional khusus untuk menghindari ambiguitas. REST kerap dipergunakan dalam mobile application, situs web jejaring sosial, mashup tools, dan automated business processes.

Arsitektur REST yang decoupled (terpisah) serta beban komunikasi yang ringan antara produsen dan konsumen membuatnya populer di dunia cloud-based API, seperti yang disajikan oleh Amazon, Microsoft, dan Google. Layanan berbasis

web yang menggunakan arsitektur REST semacam itu dinamakan RESTful APIs (Application Programming Interfaces) atau REST APIs (Sumber: ekajogja.com).

2.4 JavaScript

JavaScript merupakan salah satu bahasa script website yang paling banyak digunakan untuk menambah manipulasi script HTML dan CSS pada sisi client/browser. JavaScript mampu memberikan fungsionalitas lebih pada website, seperti validasi form, berkomunikasi dengan server serta membuat website lebih interaktif dan animatif.

JavaScript digunakan pada banyak browser seperti Internet Explorer, Firefox, Chrome, Opera, Safari dan lain sebagainya. Hampir seluruh browser mendukung JavaScript sehingga tidak perlu khawatir kode JavaScript yang digunakan pada website tidak berfungsi (Wahana Komputer, 2012:2).

2.5 JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari bahasa pemrograman JavaScript. JSON terdiri dari dua struktur, yaitu:

1. Pasangan nama dengan nilai. Pada beberapa bahasa hal ini dinyatakan sebagai object, record, struct, dictionary, hashtable, keyedlist atau associative array.

2. Daftar nilai terurutkan (anorderedlist of values). Pada kebanyakan bahasa, hal ini dinyatakan sebagai array, vector, list atau sequence (Abdul dkk, 2013:2).

2.6 NodeJS (JavaScript Runtime)

Node.js adalah perangkat lunak yang didesain untuk mengembangkan aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman

JavaScript. Bila selama ini kita mengenal JavaScript sebagai bahasa pemrograman yang berjalan di sisi client / browser saja, maka Node.js ada untuk melengkapi peran JavaScript sehingga bisa juga berlaku sebagai bahasa pemrograman yang berjalan di sisi server, seperti halnya PHP, Ruby, Perl, dan sebagainya. Node.js dapat berjalan di sistem operasi Windows, Mac OS X dan Linux tanpa perlu ada perubahan kode program.

Berbeda dengan bahasa pemrograman sisi server pada umumnya yang bersifat sinkronis atau blocking, Node.js bersifat asinkronis atau non-blocking sebagaimana halnya JavaScript bekerja. Node.js berjalan dengan basis event (event-driven). Maksud dari blocking secara sederhana adalah bahwa suatu kode program akan dijalankan hingga selesai, baru kemudian beralih ke kode program selanjutnya (Sheba, 2017:8).

2.7 SocketIO

Socket.io merupakan sebuah library JavaScript yang membantu dalam pembuatan aplikasi web yang real-time lebih mudah, dengan menggunakan socket.io kita dapat menghubungkan antara client dan server dapat terjadi secara bidirectional (dua arah). Maksudnya yaitu kita dapat menghubungkan client dan server sehingga dapat berperan sebagai pengirim dan sekaligus penerima data, komponen yang terdapat pada socket.io terdiri dari dua bagian yang pertama client-side yaitu yang berjalan pada browser, dan server-side yang dapat digunakan sebagai modul untuk Node.js (Sumber: www.kursuswebsite.org).

2.8 Express (Web Framework)

Express.js adalah satu web framework paling populer di dunia Node.js. Dokumentasinya yang lengkap dan penggunaannya yang cukup mudah, dapat membuat kita mengembangkan berbagai produk seperti aplikasi web ataupun RESTful API. Express.js pun dapat digunakan menjadi pijakan untuk membangun web framework yang lebih kompleks seperti, Sails.js, MEAN (MongoDB, Express.js, Angular.js, Node.js) dan

MERN (MongoDB, Express.js, React.js, Node.js). Express.js dibuat oleh TJ Holowaychuk dan sekarang dikelola oleh komunitas (Sumber: arfianhidayat.com).

2.9 MongoDB

MongoDB merupakan sebuah sistem basis data yang berbasis dokumen (Document Oriented Database) dan termasuk sistem basis data yang menganut paham NoSQL. NoSQL singkatan dari Not Only SQL, artinya sebuah sistem basis data tidak hanya harus menggunakan perintah SQL untuk melakukan proses manipulasi data.

MongoDB tidak memiliki tabel, kolom, dan baris. Pada MongoDB yang ada hanyalah koleksi dan dokumen. Dokumen yang terdapat dalam MongoDB dapat memiliki atribut yang berbeda dengan dokumen lain walaupun berada dalam satu koleksi. Hal ini tidak dapat dilakukan dalam RDBMS, dimana sebuah baris dalam tabel tidak mungkin memiliki kolom yang berbeda dengan baris yang lain jika berada dalam satu tabel.

MongoDB merupakan sistem basis data yang menggunakan konsep key-value, artinya setiap dokumen dalam MongoDB pasti memiliki key. Hal ini berbeda dalam RDBMS yang bisa tidak menggunakan primary key ketika membuat sebuah tabel. Sehingga walaupun kita membuat sebuah dokumen tanpa menggunakan primary key, tapi secara otomatis MongoDB memberinya sebuah key. Penggunaan konsep key-value sangat berperan penting, karena hal ini membuat MongoDB menjadi sistem basis data yang sangat cepat jika dibandingkan dengan non key-value seperti RDBMS (Eko dan Fitry, 2016:69).

III. METODOLOGI PENELITIAN

3.1 Perancangan RESTful Web Service

Pada dasarnya ada dua standard web service yang bisa digunakan yaitu SOAP dan REST. Namun yang umum sekarang digunakan di dunia web adalah REST, REST merupakan standard dalam arsitektur web yang menggunakan protokol HTTP untuk pertukaran data.

Konsep REST pertama kali diperkenalkan oleh Roy Fielding pada tahun 2000. Secara sederhana konsep ini dapat dijelaskan bahwa REST server menyediakan jalur untuk akses resource atau data, sedangkan REST client melakukan akses resource dan kemudian menampilkan atau menggunakannya. Resource yang dihasilkan sebenarnya berupa teks, namun formatnya bisa bermacam-macam tergantung kebutuhan, umumnya adalah JSON dan XML. Web service yang berbasis arsitektur REST kemudian dikenal sebagai RESTful Web Service.

3.2 Method Web Service

Dalam mengakses sebuah resource, REST juga menggunakan konsep uniform resource identifier (URI) di mana ada method yang digunakan, secara default adalah GET. Ada banyak method yang digunakan dalam web service dengan arsitektur REST, Namun yang penulis gunakan adalah sebagai berikut :

1. PUT digunakan untuk membuat (create) suatu resource.
2. GET digunakan untuk membaca (read) suatu resource.
3. POST digunakan untuk memodifikasi (update) suatu resource.
4. DELETE digunakan untuk menghapus (delete) suatu resource.

3.3 Cara Kerja RESTful Web Service

Cara kerja web service berbasis REST cukup sederhana yaitu mula-mula suatu client mengirimkan permintaan data melalui protokol HTTP (HTTP request) pada endpoint (alamat URI dari suatu resource) tertentu dan kemudian server merespon permintaan tersebut (HTTP response). Adapun komponen dari HTTP request adalah:

1. Verb, HTTP method yang digunakan misalnya GET, POST, DELETE, PUT.
2. URI, endpoint untuk mengidentifikasi lokasi resource pada server
3. Request header, berisi meta data untuk HTTP request. Contoh, jenis client/browser, format yang didukung oleh client, format dari body pesan, pengaturan cache dan lain-lain.
4. Request body, konten dari resource.

Sedangkan komponen dari HTTP response adalah:

1. Status/response code, mengindikasikan status server terhadap resource yang di-request. misal: 404, artinya resource tidak ditemukan dan 200 artinya OK.
2. HTTP version, menunjukkan versi dari HTTP yang digunakan, contoh HTTP v1.1.
3. Response header, berisi meta data untuk HTTP response. Contoh: jenis server, panjang konten, jenis konten, waktu response dan lainnya.
4. Response body, konten dari resource yang diberikan.

3.4 HTTP Response Code

Setiap terjadi request dari client ke server melalui protokol HTTP maka akan mengembalikan respon yang salah satunya adalah kode respon. Kode ini mengindikasikan status server terhadap resource yang di-request tersebut. Ada banyak kode respon HTTP dan kode-kode tersebut dikelompokkan menjadi 5 kelompok. Tabel 1. adalah kelompok kode dari respon HTTP.

Kelompok Kode	Keterangan
1xx	<i>Informational response</i>
2xx	<i>Success</i>
3xx	<i>Redirection</i>
4xx	<i>Client Errors</i>
5xx	<i>Server Errors</i>

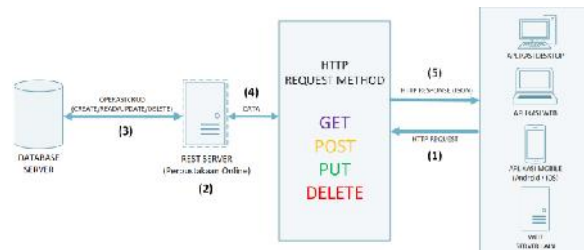
Dari 5 kelompok kode respon HTTP tersebut, penulis menggunakan beberapa kode respon yang paling umum digunakan pada RESTful Web Service, yaitu :

3.5 Arsitektur RESTful Web Service

Arsitektur RESTful Web Service dibuat dengan tujuan merancang sebuah kerangka dan cara kerja sistem yang memiliki Layanan Web atau Web Service. Tabel

Kode	Status	Keterangan
200	<i>OK</i>	<i>Request resource</i> berhasil
400	<i>Bad request</i>	Terjadi kesalahan pada <i>request client</i>
401	<i>Unauthorized</i>	Belum <i>login</i> atau belum terautentikasi
403	<i>Forbidden</i>	Client tidakizinkan untuk mengakses <i>resource</i>
404	<i>Not found</i>	<i>Resource</i> tidak ditemukan
405	<i>Method not allowed</i>	<i>Method</i> salah, atau <i>method</i> tidak diizinkan
500	<i>Internal server error</i>	Terjadi kesalahan atau <i>error</i> pada <i>server</i>

2. Adalah **Arsitektur RESTful Web Service**. Entitas dari rancangan ini terdiri dari client yang mengakses aplikasi Perpustakaan Online dengan perangkat ataupun server lain yang membutuhkan sistem informasi Perpustakaan Online di jurusan Teknik Komputer Polstri.



Gambar 1. Arsitektur RESTful Web Service

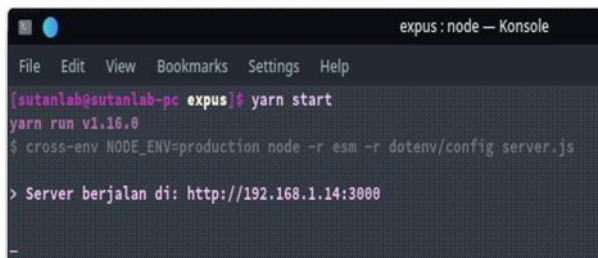
Penjelasan :

1. Pada sistem RESTful API, client akan me-request ke API endpoint di server melalui HTTP request dengan berbagai method yang disediakan, umumnya adalah PUT , GET, POST dan DELETE.
2. REST Server akan menerima request serta parameter dari client, lalu server akan segera memproses dan melakukan operasi database atau juga sering disebut dengan CRUD (CREATE, READ, UPDATE, DELETE).
3. Setelah server melakukan operasi database, server akan menerima resource berupa data dari database maupun hanya berupa respon operasi berhasil.
4. Kemudian server akan meneruskan data tersebut serta http response code untuk sebagai tanda bahwa request berhasil atau gagal ke client. Format data berupa JSON.
5. Client akan menerima JSON tersebut lalu diproses lebih lanjut.

IV. HASIL PENGUJIAN

Dari perancangan Sistem Informasi Perpustakaan menggunakan Web Service pada jurusan Teknik Komputer POLSRI menghasilkan sebuah server yang melayani dalam permintaan resource-resource yang berada di Perpustakaan.

Selain itu, berbeda dengan server pada umumnya yang biasa dikonfigurasi terlebih dahulu sebelum dipakai, server yang dibuat adalah server yang lebih flexible digunakan, dapat dengan mudah untuk berpindah dan berjalan di perangkat lain karena konfigurasinya bersifat tetap dan tersimpan dalam satu folder yang tergabung dengan proyek aplikasi.

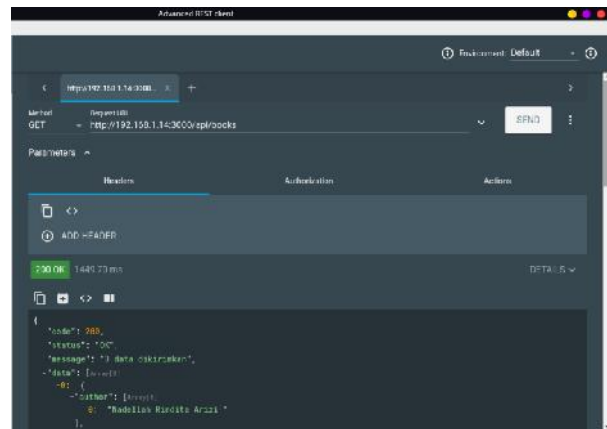


```

expus: node — Konsole
File Edit View Bookmarks Settings Help
[sutanlab@sutanlab-pc expus]$ yarn start
yarn run v1.16.0
$ cross-env NODE_ENV=production node -r esm -r dotenv/config server.js
> Server berjalan di: http://192.168.1.14:3000

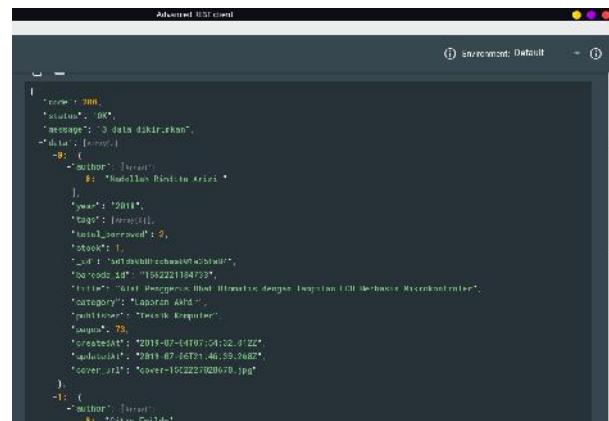
```

Setelah itu, perlu dilakukan pengujian Web Service agar dapat memastikan bahwa Web Service yang dibangun dapat bekerja dengan baik. Untuk pengujian RESTful Web Service bisa menggunakan tools Advanced REST Client atau POSTMAN. Namun kali ini penulis menggunakan tools Advanced REST Client. Adapun cara menguji Web Service menggunakan aplikasi Advanced REST Client adalah dengan cara memilih method yang digunakan dan memasukkan api endpoint pada form request URL seperti berikut :



Gambar 2. Form Request URL Perpustakaan

Pada Gambar 2. tersebut, telah ditampilkan bahwa REST server memberi kode respon 200 yang berarti request resource berhasil dengan menyertakan data yang diperoleh dari database server. Berikut adalah detail resource yang telah dikirim dari REST server berupa format data JSON.



Gambar 3. Respon Server Web Service

Pengujian *RESTful Web Service* dapat diuji dengan menggunakan metode pengujian *blackbox testing*. Sistem pendukung keputusan berikut menggunakan data uji berdasarkan data yang telah didapat dari *resource*. Sebelum melakukan pengujian, lakukan identifikasi dan perencanaan terlebih dahulu untuk mengetahui data uji yang akan dilakukan pengujian.

V. KESIMPULAN

Berdasarkan pembahasan dan pengujian, maka dapat diperoleh beberapa kesimpulan yaitu :

1. Web Service ini membuat sistem informasi menjadi flexible, reusable, dan fungsional yang bisa terintegrasi dengan Aplikasi Perpustakaan Online Jurusan Teknik Komputer serta aplikasi lainnya yang membutuhkan resource dari Perpustakaan di jurusan Teknik Komputer.
2. Dengan adanya Web Service ini, semua platform aplikasi bisa dengan mudah mengakses resource perpustakaan Jurusan Teknik Komputer hanya dengan melalui HTTP request tanpa harus berinteraksi dengan query-query pada database server.

REFERENSI

- [1] Abdul Aziz, Wiharto, dan Bayu Wicaksono. 2013. *Pemanfaatan Web Service Moodle berbasis REST-JSON untuk membangun Moodle Online Learning Extension berbasis Android*
- [2] J. Webber, S. Parastatidis, and I. Robinson. 2010. *REST in Practice*. United States: O'Reilly Media, Inc.
- [3] Hidayat, Arfian. 2018. Pengenalan Express JS. <https://arfianhidayat.com/pengenalan-express-js>
- [4] Putra, Eko Kurniawanto dan Fitry Rahmayeni. 2016. Implementasi MongoDB untuk sistem informasi Bimbingan Konseling berbasis Web.
- [5] Saputra, Eka. 2015. REST (Representational State Transfer). <https://ekajogja.com/arsip/definisi/rest-representational-state-transfer>.
- [6] Sheba, Lusfekar. 2017. *Panduan Node.js 8.X dalam 14 BAB*.
- [7] Wahana Komputer. 2012. Paling Dicari! JavaScript Source Code. Yogyakarta: Andi Offset
- [8] Deswitansyah, Irwan. 2017. Mengenal apa itu Socket.io. <http://kursuswebsite.org/mengenal-apa-itu-socket-io/>.