



**PROSIDING SEMINAR NASIONAL
PENERAPAN TEKNOLOGI INFORMASI
DALAM MENDUKUNG
MANAJEMEN RITEL**

**Hari, Tanggal;
Di Aula Lantai 3 Kampus
Universitas Informatika dan Bisnis Indonesia
Jl. Soekarno Hatta No. 643 Bandung**

PROSIDING SEMINAR NASIONAL
PENERAPAN TEKNOLOGI INFORMASI
DALAM Mendukung MANAJEMEN RITEL



Lembaga Penelitian dan Pengabdian Kepada Masyarakat
Universitas Informatika dan Bisnis Indonesia



DAFTAR ISI

1. Pengaruh Persepsi Kebermanfaatan dan Kepuasan Wajib Pajak Terhadap Penggunaan *E-Filing* di Setda Pemkab Bandung Barat
Elan Rusnendar, Nicholas Naibahao
2. Studi Literatur Kepemimpinan Perempuan Budaya Jawa *Asih Asah Asuh* Untuk Meningkatkan Kinerja Organisasi
Sofia Nuryanti
3. Pengaruh Kinerja Keuangan, Kebijakan Dividen, dan Ukuran Perusahaan Terhadap Nilai Perusahaan Non Keuangan
Eko Purwanto
4. Pelaksanaan CSR PT. PLN Persero Meningkatkan Pengetahuan Lingkungan Hidup Masyarakat Kampung Nelayan Hutan Mangrove Surabaya
Rebecca Kizia, Rosita Manurung
5. *Self Disclosure* Homoseksual (Studi Komunikasi Antarpribadi Dalam Hubungan Pertemanan)
Indah Sari
6. Analisis Bahasa Rupa Pada Film Animasi ‘Sita Sings The Blues’
Citra Kemala Putri
7. Analisis Makna Denotasi dan Konotasi Pada Iklan Produk Jeans Levi’s
Citra Kemala Putri
8. Connective Action Melalui Petisi Daring Sebagai Pelengkap Gerakan Sosial Nyata: Petisi Dukong Transportasi Online di Jawa Barat
Rachmawati Windyaningrum, Diwan Setiawan
9. Kemerdekaan Pers! Mengapa dan Untuk Apa?” (Jurnal Dewan Pers, Edisi:12,September 2016 Oleh: Bagir Manan), Sebuah Analisis Wacana Kritis.
Hanafi
10. Pengaruh Kualitas Sistem, Kualitas Informasi dan Kualitas Layanan Terhadap Kepuasan Pengguna Website SMK Negeri 3 Bandung
Vani Maharani Nasution, Ridona U. S.
11. Media Pembelajaran Bahasa Indonesia Untuk Penutur Asing
Reni Nursyanti, Ratih Hardiantini, Nisya Syafiska

12. Aplikasi Pengendalian Produksi di Bagian Pengemasan PT. Bio Farma (Persero)
Rd. Yadi Rakhman Alamsyah, Ade Cahyan
13. Perangkat Lunak Penjualan Online Studi Kasus Pada PT. Sinergi Sukses Mobilindo
Titan Paramayoga, Rd. Yadi Rakhman Alamsyah, Stefanus
14. Pengaruh Sistem Informasi terhadap Efektivitas Kerja Pegawai (Studi Empiris pada Kantor Pusat Yayasan Rumah Zakat Indonesia)
Graha Prakarsa
15. Praanggapan Dalam Interview Ahok Dengan Kompas TV Perihal “Keluarnya Ahok Dari Gerindra”
Dwi Pebrina Sinaga, Eline Rozaliya Winarto
16. Sistem Informasi Pengelolaan Data Penggunaan Sumber Energi Dalam Industri Garmen (Studi Kasus: PT. Kahatex Bandung)
Chairul Habibi, Marwondo
17. Pemodelan Arsitektur Enterprise Sekolah Menggunakan Enterprise Arsitektur Planing (EAP)
Reni Nursyanti, R. Yadi Rahman A, Rohmat
18. Pemanfaatan Teknologi Radio Frequency Identification (RFID) Untuk Mendeteksi dan Mengidentifikasi Kehadiran
Ivan Michael Siregar
19. Implementasi SIG Pada Pengelolaan Sumber Benih Tanaman Hutan
Ivan Michael Siregar
20. Pengaruh Kualitas Pelayanan dan Pengetahuan Perpajakan Terhadap Kepatuhan Wajib Pajak Orang Pribadi Pada KPP Pratama Bandung Bojonagara
Eko Purwanto, Nabilah Rizky Faujiah
21. Pengaruh Jumlah Wisatawan, Jumlah Hotel, Terhadap Penerimaan Pajak Hotel
Eko Purwanto
22. Aplikasi Perencanaan Produksi (Studi Kasus: PT Multi Garmentama)
Marwondo, Vani Maharani Nasution
23. Implementasi Aplikasi Mobile Android Untuk Manajerial Pelawatan Anggota Organisasi
Akbar Pasha, Ivan Michael Siregar
24. Pengembangan Sistem Informasi Penanganan Keluhan (Studi Kasus: Istana Plaza)
Chairul Habibi, Riki Aprian

25. Perancangan Desain User Experience (UX) dan User Interface (UI) Aplikasi 'Ruang Anak'
Nichi Hana Karlina, Annisa Ayuratnasari
26. Perancangan Identitas Visual dan Media Launching Komunitas 'Peduli Jilbab on The Street' Regional Bandung
Nichi Hana Karlina, Titin Mulyati
27. Perancangan Single Page Application menggunakan Flask-Restful dan Mithril.js (Studi Kasus: Website Directory Travel Haji dan Umroh)
Teguh Reinaldo
28. Hubungan Antara Atribusi Bencana Banjir Dengan Dukungan Sosial Pada Warga Kelurahan Andir Kecamatan Baleendah Bandung Selatan
Evi Srinur Hastuti

**PERANCANGAN SINGLE PAGE APPLICATION MENGGUNAKAN
FLASK-RESTFUL DAN MITHRIL.JS
(Studi Kasus: Website Directory Travel Haji dan Umroh)**

Teguh Reinaldo

Fakultas Teknologi dan Informatika, UNIBI
teguh.reinado@unibi.ac.id

Abstrak

Pangsa pasar jamaah umrah yang besar di Indonesia memicu adanya biro perjalanan swasta yang tidak terdaftar melanggar aturan pemerintah dengan ikut menyelenggarakan perjalanan ibadah umrah dan mengakibatkan berbagai masalah, seperti peserta umrah tidak jadi berangkat ke tanah suci, pelantaran maupun penipuan yang mengakibatkan kerugian material dan finansial. Permasalahan ini dapat dihindari jika calon jamaah memiliki informasi yang lengkap mengenai biro perjalanan umrah mana yang aman dan berkualitas. Informasi itu masih tersebar dan belum ada katalog yang memudahkan. Artikel ini memaparkan penelitian tentang perancangan konsep Single Page Application (SPA) pada Website Directory Travel Haji dan Umroh menggunakan Flask-Restful dan Mithril.js. SPA adalah sebuah konsep pengembangan web terbaru yang telah diuji oleh beberapa peneliti dan unggul dalam hal penghematan penggunaan bandwidth data dan loading time dibanding website tradisional.

Kata Kunci: *Web Direktori, Single Page Application, Website*

Abstract

The large market shares of umrah pilgrims in Indonesia triggered unregistered private travel agents violating the government's rules by participating in the Umrah pilgrimage and causing problems, such as Umrah participants not to go to holy land, abandonment or deception resulting in material and financial losses. This problem could be avoided if prospective pilgrims have complete information about which umrah travel agency is safe and have excellent quality. The information is still scattered and there is no catalog that makes it easy thus Directory Website become solution. This article describes research on Single Page Application (SPA) concept design on Hajj and Umroh Website Directory Travel using Flask-Restful and Mithril.js. SPA is a new web development concept that has been tested by several researchers and excels in terms of savings in bandwidth usage and loading time compared to traditional websites.

Keywords: *Web Directory, Single Page Application, Website*

1. PENDAHULUAN

1.1 Latar Belakang

Sejak disahkannya peraturan Menteri Agama Republik Indonesia nomor 18 Tahun 2015 tentang penyelenggaraan perjalanan ibadah umrah (PPIU), biro perjalanan wisata kini dapat menjadi provider perjalanan ibadah umrah jika telah mendapatkan izin dari kementerian Agama. Tercatat berdasarkan data (Daftar Penyelenggara PPIU, 2017) terdapat sekitar 900 biro perjalanan umrah yang telah resmi terdaftar di seluruh Indonesia pada tahun 2017. Hal ini berbanding lurus dengan minat masyarakat muslim Indonesia untuk beribadah umrah, dilansir dari kata data (Berapa Jemaah Umrah dari Indonesia, 2017) Negara Indonesia masuk pada urutan ke 4 dari 10 besar negara dengan jumlah jemaah umrah terbesar menurut Kementerian Agama dan Umrah Arab Saudi 2016, yaitu sekitar 699.6 ribu Jemaah.

Pangsa pasar yang besar ini memicu biro perjalanan swasta yang tidak terdaftar melanggar aturan pemerintah dengan ikut menyelenggarakan perjalanan ibadah umrah dan mengakibatkan berbagai masalah, seperti peserta umrah tidak jadi berangkat ke tanah suci, pelantaran maupun penipuan yang mengakibatkan kerugian material dan finansial. Permasalahan berikut dapat diatasi jika calon jemaah dapat memilah dan memilih biro perjalanan umrah mana yang aman dan berkualitas. Namun pada kenyataannya informasi mengenai biro perjalanan yang diberikan oleh departemen agama hanya berupa tabel yang bersifat statis. Informasi lengkap mengenai biaya, hak dan kewajiban biro perjalanan masih terpisah pada website biro perjalanan itu masing-masing. Maka diperlukan sebuah website yang bersifat seperti sebuah katalog atau disebut website direktori.

Dalam pengembangannya website direktori yang ada terdapat berbagai permasalahan khususnya lamanya proses page loading time dan response time pada saat pengguna mencari informasi yang diinginkan. Hal ini disebabkan oleh struktur

website yang masih tradisional, yakni setiap pengguna berinteraksi dengan tautan informasi, maka halaman baru akan terbuka, konsekuensinya semua asset seperti CSS, Javascript, HTML dan Images harus dimuat ulang, sehingga memperlambat respon bagi pengguna.

Kini, perkembangan website modern mengenal sebuah konsep penyegaran pada sebagian halaman atau komponen website yang independen atau disebut sebagai Single Page Application (SPA). Menurut M.S. Mikowski (Single Page Web Application, 2014) SPA, meningkatkan responsifitas bagi pengguna website dan membuat kesan pada pengguna sedang menggunakan aplikasi desktop karena semua proses dilakukan pada satu halaman yang sama tanpa perlu memuat ulang semua asset. Sehingga menghemat bandwidth dan response time sebuah website.

Berdasarkan pemaparan masalah di atas penulis tertarik untuk merancang sebuah website direktori berbasis SPA namun berbeda dengan penelitian sebelumnya kami menggunakan Flask-Restful dan Mithril.js sebagai perkakas pengembangan. Selain itu Studi kasus yang diambil adalah perancangan website direktori biro travel Umrah dan Haji yang telah terdaftar oleh Departemen Agama dan memiliki informasi yang dibutuhkan oleh calon jemaah.

1.2 Identifikasi Masalah

Berdasarkan pemaparan latar belakang di atas penulis mengidentifikasi beberapa permasalahan pokok pada penelitian ini antara lain:

1. Informasi mengenai biro perjalanan umrah dan haji yang terdaftar resmi oleh Kementerian Agama berbentuk tabel statis dan sederhana, sehingga kurang membantu jemaah dalam memilih biro perjalanan tersebut. Konten informasi yang diberikan masih sebatas biodata dasar, tidak ada informasi lengkap lain seperti biaya perjalanan, hak dan kewajiban, pilihan paket, dll. Akibatnya jemaah harus mencari satu-persatu pada website biro perjalanan tersebut.

2. Keterbatasan teknologi website tradisional adalah masih memiliki masalah seperti waktu respon yang lambat dan kurang interaktif bagi pengguna.

1.3 Rumusan Masalah

Sesuai dengan pengidentifikasian masalah diatas maka dapat dirumuskan beberapa masalah yang yang mendasari penelitian ini sebagai berikut:

1. Bagaimana pengguna aplikasi dapat mencari informasi yang lengkap mengenai biro perjalanan ibadah umrah dan haji dalam satu tempat?
2. Bagaimana membuat aplikasi web biro perjalanan ibadah umrah dan haji yang memiliki waktu respon yang cepat dan interaktif?

1.4 Batasan Masalah

Pada penelitian ini penulis memiliki beberapa batasan antara lain:

1. Informasi lengkap mengenai masing-masing biro perjalanan yang terdaftar meliputi hak, kewajiban, jadwal keberangkatan dan biaya berdasarkan data dummy yang degenerate oleh user.
2. Aplikasi yang dihasilkan berbasis Restful-API dan berbentuk Single Page App Website belum dan bersifat Web Responsif dapat digunakan pada website modern seperti Chrome dan Mozilla.
3. Arsitektur resful dibatasi, tidak menggunakan konsep cache.
4. Pengisian data travel dilakukan oleh administrator. Pengguna dengan otorisasi Jamaah adalah sebuah *dummy* sebanyak tiga.

3. LANDASAN TEORI

3.1 Web Directory

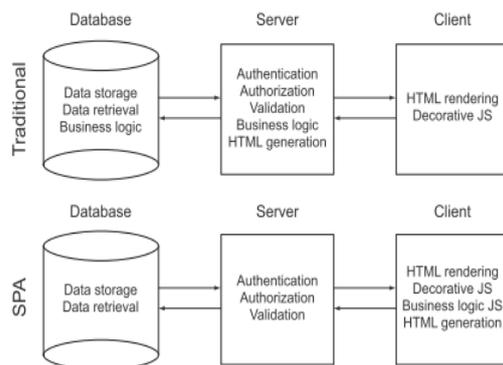
Kata *web direktori* atau katalog website sering disalah artikan dengan istilah *search engine* meskipun memiliki konsep

yang sama keduanya merupakan hal yang berbeda. Menurut artikel yang ditulis oleh Jerri Collins (*What Is a Web Directory?* 2017), web direktori adalah sebuah website yang berfungsi untuk membuat daftar dari kumpulan website yang memiliki tema yang khusus bisa satu atau lebih. Selain itu konten dari web direktorielihara oleh pengguna secara koloberatif. Data pada web direktori dapat ditulis oleh seorang administrator atau sekumpulan editor komunitas yang secara sukarela melakukan proses pembaharuan data agar data yang dihasilkan transparan dan akuntabel. Hal lain yang membedakan web direktori dengan mesin pencari adalah jumlah koleksi tautan yang di simpan web direktori lebih kecil karena tidak menggunakan spiders.

Penulis memilih untuk membuat website direktori travel haji dan umrah terdaftar karena informasi resmi mengenai biro perjalanan resmi yang diberikan oleh Departemen Agama berupa tabel sederhana yang bersifat statis. Sehingga informasi-informasi lain yang dibutuhkan jamaah seperti biaya, hak dan kewajiban biro perjalanan masih terpisah pada website biro perjalanan itu masing-masing. Selain itu karena sifatnya koloberatif pengguna / jamaah yang telah menggunakan jasa sebuah travel dapat memberikan review sebagai tambahan informasi untuk jamaah lain.

3.2 Single Page Application

Kini, perkembangan website modern mengenal sebuah konsep penyegaran pada sebagian halaman atau komponen website yang independen atau disebut sebagai Single Page Application (SPA). Menurut M.S. Mikowski (Single Page Web Application, 2014) SPA, meningkatkan responsifitas bagi pengguna website dan membuat kesan pada pengguna sedang menggunakan aplikasi desktop karena semua proses dilakukan pada satu halaman yang sama tanpa perlu memuat ulang semua asset.

**Gambar 2.1**

Single Page Application (Mikowski, 2014)

Gambar diatas menunjukkan perbedaan proses antara konsep web tradisional dan Single Page App (Mikowski, 2014), proses logika bisnis dan proses membangkitkan halaman html dilakukan di sisi client, berbeda dengan pengembangan tradisional dimana proses logika bisnis dan proses lain terletak di sisi basis data dan/atau pada server. Oleh karena itu SPA juga disebut dengan Aplikasi *Fat Client*.

Beberapa penelitian menunjukkan keunggulan dari konsep SPA dibandingkan dengan web tradisional, seperti penelitian yang dilakukan Aryani Ristyabudi dan Husni Tamhrin (Penerapan Single Page Application Pada Proses Pengisian Online Data Rencana Studi Mahasiswa, 2016) menerapkan SPA pada untuk menggantikan Website Pengisian KRS sebelumnya, menghemat transfer time sebesar tiga kali lipat dari 178,70 detik ke 6.54 detik. Penelitian ini menggunakan teknologi PHP dan AngularJS. Namun terdapat celah keamanan pada program yang dibuat yaitu tidak ada otorisasi tambahan pada request dan response mengakibatkan pesan dapat dilihat.

Peneliti lain yaitu Adib Luqman Azhari dan Radius Tanone (Analisis Penerapan Single Page Application Menggunakan Teknologi AJAX dan REST API, 2017), menerapkan SPA pada aplikasi Reservasi Wisma di UKSW, penelitian tersebut menunjukkan penggunaan bandwidth dan loading time dari proses Create, Read, Update dan Delete (CRUD) pada tiap modul.

Hasil rata-rata dari pengujian membuktikan penggunaan konsep SPA menghemat bandwidth data kurang dari sepersepuluh dan menghemat loading time yang dibutuhkan menjadi sepertiga dari proses sebelumnya.

3.3 Frontend SPA Framework

Pengembangan sebuah Single Page Application terbagi menjadi dua yaitu Client dan Server. Pada sisi Frontend, sudah terdapat beberapa library yang dapat membantu pengembang membuat single page application secara cepat, seperti; React, AngularJS, VueJS dan MithrilJS. Framework tersebut memiliki fungsi-fungsi dasar pembuat SPA (Mikowski, 2014) sebagai berikut:

1. AJAX (Asynchronous JavaScript + XML) teknik pemrograman web yang memungkinkan aplikasi mengirim dan menerima data kepada server dibelakang layar (background) konten halaman web dapat berubah tanpa mereload aplikasi.
2. Melakukan transport data yang berupa (XML, JSON).
3. Melakukan proses pembangkitan html pada sebuah elemen HTML seperti DOM.
4. Memiliki konsep Model-View-Controller yang memisahkan logika bisnis, data dan tampilan disisi Client sehingga memudahkan proses pengembangan aplikasi kompleks.

Pada penelitian ini penulis memilih menggunakan MithrilJS karena selain memiliki ukuran file yang kecil dan performansi yang baik.

3.4 RESTFUL API

RestFul API adalah sebuah arsitektur pengembangan Application Programable Interface (API) yang menghasilkan solusi optimal. Pada artikel lazlojuly (What is a RESTful API? 2016) yang mengutip disertasi dari Roy Fielding yang berjudul Architectural Styles and the Design of Network-based Software Architectures,

terdapat 6 pakem Arsitektur REST yaitu lima diantaranya adalah:

1. Pemisahan antara arsitektur Client dan Server, hal ini memungkinkan tiap sisi untuk berkembang secara independen, sehingga API yang dibuat dapat digunakan oleh berbagai client (Web, Desktop, IOT, dll).
2. Komunikasi antara client dan server harus tanpa *state*. Server tidak usah mengingat *state* yang dimiliki client untuk setiap request, sehingga membebaskan sumber daya server untuk melayani permintaan client yang lain.
3. Memiliki sistem cache, client dapat menyimpan hasil permintaan pada jaringan di server, hal ini dapat menghemat waktu komunikasi jika client meminta permintaan yang sama atau telah dilakukan sebelumnya.
4. Agar sistem cache ini dapat digunakan secara konsisten, maka diperlukan antarmuka yang konsisten sehingga data yang dikirim sesuai dengan format yang standar. Meliputi; Mengidentifikasi resource, Memanipulasi resource melalui abstraksi, pesan yang dapat dimengerti dengan mudah, hypermedia adalah mesin dalam perubahan *state*.
5. Dapat digunakan pada jaringan yang berlapis-lapis, contoh proxy untuk kepentingan membuat cache.

Pada penelitian ini arsitektur berbasis Restful digunakan untuk pengembangan Single Page Application. Terdapat beberapa backend framework yang telah mengikuti konsep ini salah satunya adalah Flask-Restful. Library yang dimiliki oleh Flask, sebuah python framework yang telah memenuhi prinsip-prinsip arsitektur Restful.

4. METODE PENELITIAN

Metode penelitian yang digunakan adalah melakukan perancangan software meliputi pembuatan model uml dan

perancangan user interface dan pembuatan restful API untuk Single Page Application Website Directory Travel Haji dan Umroh.

4.1 Jenis dan Metode Pengumpulan Data

Pengumpulan data dilakukan dengan cara mengumpulkan data (Dokumen Daftar Penyelenggara Ibadah Umrah, 2017) yang berbentuk travel dan telah resmi terdaftar di Departemen Agama. Serta mempelajari aturan-aturan pemerintah tentang penyelenggara ibadah umrah (PPIU) sebagai panduan penambahan informasi yang penting dipahami bagi calon jamaah umrah.

4.2 Alat dan Bahan

Peralatan yang digunakan selama penelitian terdiri dari software dan hardware. Perangkat lunak yang digunakan antara lain Database SQLite, Web Browser: Google Chrome dan Mozilla Firefox, Untuk mengembangkan SPA: Python 3.6, Flask-Restful dan Mithril.js, Sebagai tambahan untuk user interface digunakan Tabler dashboard template. Untuk membuat Diagram UML digunakan Visual Editor Community Edition. Sedangkan hardware yang digunakan adalah laptop dengan spesifikasi prosesor Intel Core i3, RAM 6 GB, harddisk 500 GB dan sistem operasi Windows 10.

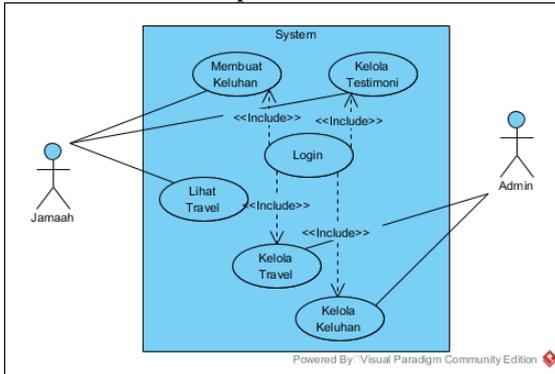
4.3 Perancangan SPA

Perancangan Single Page Application Website Directory Travel Haji dan Umroh menggunakan diagram Unified Modelling Language (UML) meliputi use case diagram, class diagram dan sequence diagram. Berikut adalah model UML dari aplikasi SPA yang dibuat untuk penelitian ini.

4.3.1 Use Case Diagram

Diagram Use Case digunakan untuk mengilustrasikan fungsionalitas utama dari sistem. Didalam diagram ini hubungan antar actor dan sistem diilustrasikan, seperti

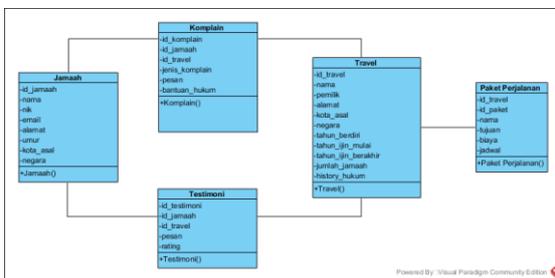
terlihat dari Gambar 3.1, Terdapat dua actor yaitu jamaah dan administrator, masing-masing memiliki hubungan dengan sistem, seluruh pengguna termasuk guest, dapat melihat daftar travel, bagi jamaah yang sudah login, mereka dapat memasukan testimoni atau keluhan terhadap travel yang mereka pakai. Untuk user administrator, dapat mengelola travel, mengelola keluhan untuk diteruskan kepada stake holder.



Gambar 3.1
Diagram Usecase

4.3.2 Class Diagram

Diagram Kelas digunakan untuk menggambarkan hubungan antar kelas/objek dan atribut-nya, dalam ruang lingkup sebuah sistem. Selain itu pada diagram ini dapat terlihat layanan yang berupa fungsi dan prosedur yang tersedia untuk melakukan perubahan nilai atribut dari kelas tersebut. Diagram kelas untuk Single Page Application ini diilustrasikan pada Gambar 3.2.



Gambar 3.2
Diagram Class

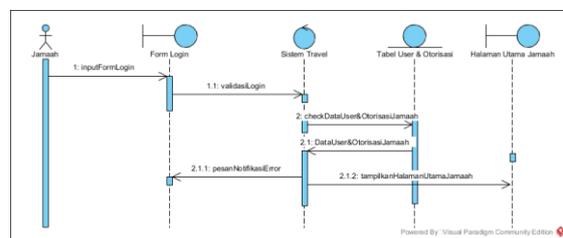
Terdapat lima kelas utama dalam single page application ini yakni kelas Jamaah, kelas Travel, kelas Paket Perjalanan, kelas Komplain, dan Kelas Testimoni. Masing-masing memiliki hubungan yang terikat, Jamaah dapat menulis komplain atau testimony terhadap travel yang pernah digunakan. Travel memiliki beberapa pake perjalanan.

4.3.3 Sequence Diagram

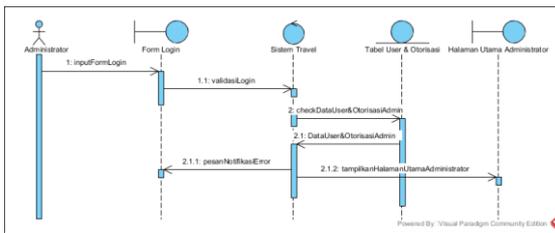
Diagram Sequence merepresentasikan alur komunikasi dari objek-objek pada sebuah sistem dilengkapi dengan ilustrasi waktu, ketika pesan dikirim atau diterima yang digambarkan secara vertical. Pada penelitian ini terdapat 5 sequence diagram, yaitu:

4.3.3.1 Sequence Diagram Login

Diagram sequence untuk proses login terbagi menjadi dua, untuk jamaah dan administrator diilustrasikan pada Gambar 3.3 dan Gambar 3.4 Pertama-tama jamaah atau administrator menginputkan data pada form login seperti username dan password, kemudian sistem akan memvalidasi apakah pengguna itu ada dan password yang dimasukan sesuai pada tabel User, dan pengguna itu memiliki otorisasi yang sesuai, jamaah atau adminstrasi, jika berhasil maka sistem akan melanjutkan pengguna untuk mengakses halaman utama, sesuai dengan otorisasinya.



Gambar 3.3
Diagram Sequence Login Jamaah

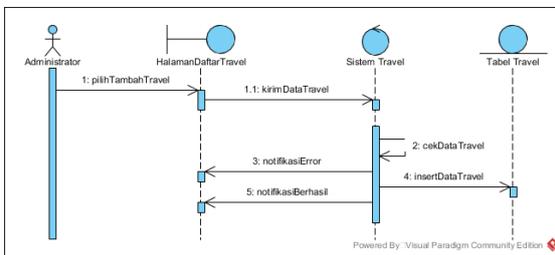


Gambar 3.4

Diagram Sequence Login Adminstrator

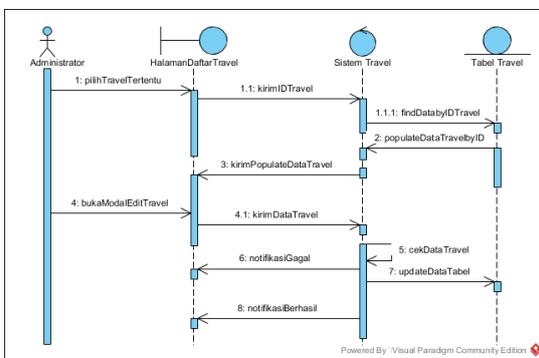
4.3.3.2 Sequence Diagram Kelola Travel

Diagram sequence untuk proses kelola travel diilustrasikan pada Gambar 3.5, 3.6 Pertama-tama pengguna dengan otorisasi admin masuk pada menu travel, dihalaman travel terdapat tabel dimana terdapat tombol add, edit dan delete, ketika tombol tersebut di tekan pengguna akan masuk ke sebuah modal dialog sesuai dengan aksinya REST-Ful API akan menyesuaikan. Contoh, post untuk add, get untuk menampilkan tabel/merefresh tabel, put untuk mengupdate dan delete untuk menghapus data.



Gambar 3.5

Diagram Sequence Tambah Travel

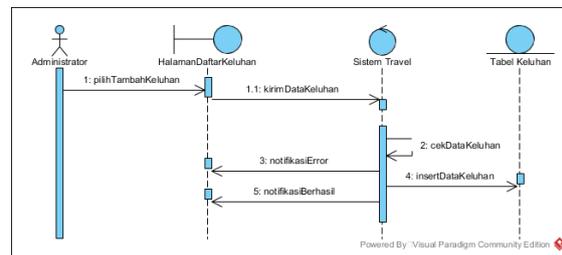


Gambar 3.6

Diagram Sequence Edit Travel

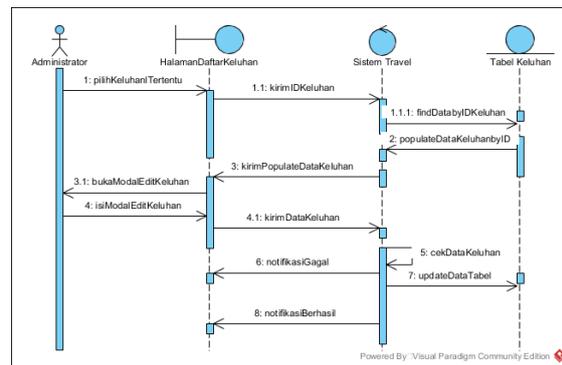
4.3.3.3 Sequence Diagram Kelola Keluhan

Diagram sequence untuk proses kelola travel diilustrasikan pada Gambar 3.7 dan 3.8 Pertama-tama pengguna dengan otorisasi admin masuk pada menu keluhan, dihalaman travel terdapat tabel dimana terdapat tombol edit dan delete, ketika tombol tersebut di tekan pengguna akan masuk ke sebuah modal dialog sesuai dengan aksinya REST-Ful API.



Gambar 3.7

Diagram Sequence Tambah Keluhan

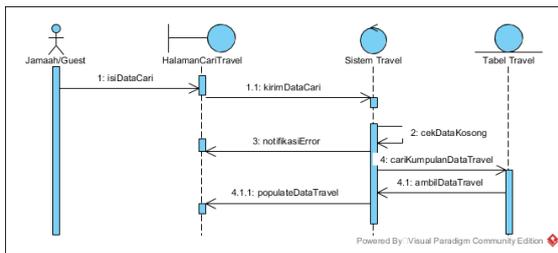


Gambar 3.8

Diagram Sequence Edit Keluhan

4.3.3.4 Sequence Diagram Lihat Travel

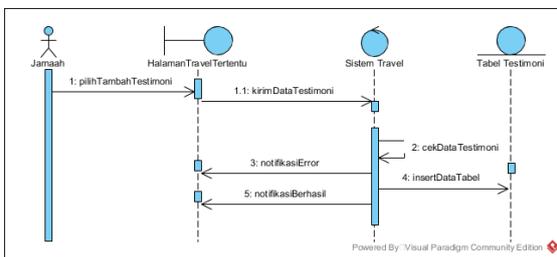
Diagram sequence untuk proses kelola travel diilustrasikan pada Gambar 3.9 Pertama-tama pengguna baik yang memiliki otorisasi ataupun tidak masuk pada menu lihat travel, terletak dihalaman depan, pengguna dapat mencari travel yang diinginkan sesuai dengan parameter searching seperti kota, dan nama travel itu sendiri. RESTFul API yang digunakan adalah fungsi get untuk mendapatkan data.



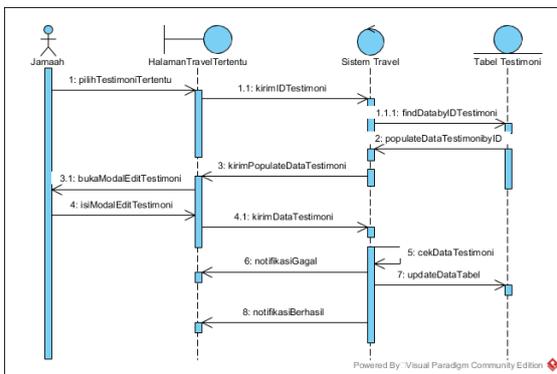
Gambar 3.9
Diagram Sequence Lihat Travel

4.3.3.5 Sequence Diagram Kelola Testimoni

Diagram sequence untuk proses kelola travel diilustrasikan pada Gambar 3.10 dan 3.11 Pertama-tama pengguna dengan otorisasi admin masuk pada menu travel tertentu, dihalaman travel menu untuk menambah, mengedit dan mengdelete testimoni, khusus fungsi edit dan delete dapat dilakukan oleh penulis testimony itu, bukan orang lain. Testimoni disimpan didalam tabel testimoni.



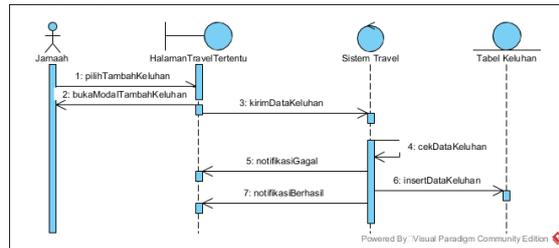
Gambar 3.10
Diagram Sequence Tambah Testimoni



Gambar 3.11
Diagram Sequence Edit Testimoni

4.3.3.6 Sequence Diagram Membuat Keluhan

Diagram sequence untuk proses kelola travel diilustrasikan pada Gambar 3.9 Pertama-tama pengguna yang memiliki otorisasi jamaah, masuk ke menu travel tertentu didalam travel terdapat menu buat keluhan yang nantinya dimasukan kedalam tabel keluhan, sifatnya rahasia jadi hanya pihak administrator yang dapat melihat data keluhan.



Gambar 3.12
Diagram Sequence Tambah Keluhan

4. HASIL DAN PEMBAHASAN

Bagian ini memaparkan tahapan-tahapan implementasi hasil perancangan Single Page Application yang mencakup sisi backend yaitu pembuat Kelas-kelas kedalam python, RestFul API untuk fungsi GET, POST, PUT dan Delete. Serta sisi frontend, penyiapan antarmuka. Tahapan-tahapan implementasi diantaranya adalah:

4.1 Penyiapan Backend

Pada bagian ini, dirancang RESTFUL-API untuk Kelas Utama yaitu kelas Jamaah, Travel, Komplain, dan Testimoni.

4.1.1 Kelas User

Pada kelas ini terdapat tiga field utama yaitu username, password dan otorisasi yang terbagi dua menjadi administrator dan jamaah, kelas ini diisi secara manual oleh penulis dalam pengembangan purwarupa ini. terdapat satu metode yaitu `find_by_username`, untuk menentukan pengguna memiliki hak akses sesuai otorisasinya.

user.py

```

class User(Resource):
    TABLE_NAME = 'users'
    def __init__(self, _id, username, otoritas):
        self.id = _id
        self.username = username
        self.password = password
        self.otoritas = otoritas
    @classmethod
    def find_by_username(cls, username, otoritas):
        connection = sqlite3.connect('data.db')
        cursor = connection.cursor()
        query = "SELECT * FROM {table} WHERE username=? and otoritas=?".format(table=cls.TABLE_NAME)
        result = cursor.execute(query, (username,otoritas))
        row = result.fetchone()
        if row:
            user = cls(*row)
        else:
            user = None
        connection.close()
        return user

```

4.1.2 Kelas Travel

Pada kelas travel terdapat tiga field utama untuk informasi travel yaitu nama, pemilik dan alamat. Pada kelas ini digunakan Flask-Restful yang terdiri dari get, post, put, dan delete.

travel.py

Fungsi find_by_name

```

from flask_restful import Resource, reqparse
import sqlite3
class Travel(Resource):
    TABLE_NAME = 'travel'
    @classmethod
    def find_by_name(cls, name):
        connection = sqlite3.connect('data.db')
        cursor = connection.cursor()

        query = "SELECT * FROM {table} WHERE name=?".format(table=cls
            .TABLE_NAME)
        result = cursor.execute(query, (name,))
        row = result.fetchone()
        connection.close()
        if row:
            return {'travel': {'name': row[0], 'pemilik': row[1], '
                alamat': row[2]}}

```

RESTFUL

```

def get(self, name):
    travel = self.find_by_name(name)
    if travel:
        return travel
    return {'message': 'Data TravelUmrahHaji tidak ditemukan'},
        404
def post(self, name):
    if self.find_by_name(name):
        return {'message': "Data travel dengan nama '{}' sudah
            ada.".format(name)}
    data = Travel.parser.parse_args()
    travel = {'name': name, 'pemilik': pemilik, 'alamat': alamat}
    Travel.insert(travel)
    return travel
def put(self, name):
    data = Travel.parser.parse_args()
    travel = self.find_by_name(name)
    updated_travel = {'name': name, 'pemilik': pemilik, 'alamat':
        alamat}
    if travel is None:
        Travel.insert(updated_travel)
    else:
        Travel.update(updated_travel)
    return updated_travel

```

4.1.3 Kelas Keluhan

Pada kelas keluhan terdapat tiga field utama untuk informasi travel yaitu id_keluhan, id_travel, id_jamaah, jenis_keluhan, dan pesan. Pada kelas ini digunakan Flask-Restful yang terdiri dari get, post, put, dan delete.

keluhan.py

Fungsi find_by_id

```

from flask_restful import Resource, reqparse
import sqlite3
class Keluhan(Resource):
    TABLE_NAME = 'keluhan'
    @classmethod
    def find_by_id(cls, id):
        connection = sqlite3.connect('data.db')
        cursor = connection.cursor()
        query = "SELECT * FROM {table} WHERE id=?".format(table=cls.
            TABLE_NAME)
        result = cursor.execute(query, (id,))
        row = result.fetchone()
        connection.close()
        if row:
            return {'keluhan': {'travel_id': row[0], 'user_id': row[1]
                }, 'jenis_keluhan': row[2], 'pesan': row[3]}

```

RESTFUL

```

def get(self, id):
    keluhan = self.find_by_id(id)
    if keluhan:
        return keluhan
    return {'message': 'Data KeluhanUmrahHaji tidak ditemukan'},
        404
def post(self, id):
    if self.find_by_id(id):
        return {'message': "Data keluhan dengan id '{}' sudah
            ada.".format(id)}
    data = Keluhan.parser.parse_args()
    keluhan = {'keluhan': {'travel_id': row[0], 'user_id': row[1]
        , 'jenis_keluhan': row[2], 'pesan': row[3]}}
    Keluhan.insert(keluhan)
    return keluhan
def put(self, id):
    data = Keluhan.parser.parse_args()
    keluhan = self.find_by_id(id)
    updated_keluhan = {'keluhan': {'travel_id': row[0], 'user_id':
        row[1], 'jenis_keluhan': row[2], 'pesan': row[3]}
    if keluhan is None:
        Keluhan.insert(updated_keluhan)
    else:
        Keluhan.update(updated_keluhan)
    return updated_keluhan

```

4.1.4 Kelas Testimoni

Pada kelas testimoni terdapat tiga field utama untuk informasi travel yaitu id_keluhan, id_travel, id_jamaah, jenis_keluhan, dan pesan. Pada kelas ini digunakan Flask-Restful yang terdiri dari get, post, put, dan delete.

```

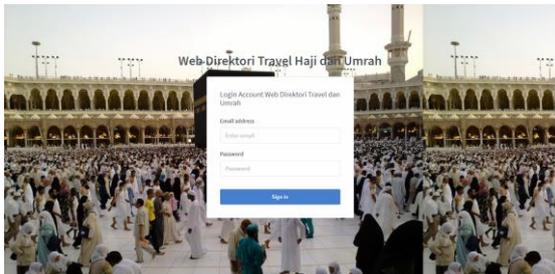
testimoni.py
Fungsi find_by_id
from flask_restful import Resource, reqparse
import sqlite3
class Testimoni(Resource):
    TABLE_NAME = 'testimoni'
    @classmethod
    def find_by_id(cls, id):
        connection = sqlite3.connect('data.db')
        cursor = connection.cursor()
        query = "SELECT * FROM {table} WHERE id=?".format(table=cls.TABLE_NAME, id=id)
        result = cursor.execute(query, (id,))
        row = result.fetchone()
        connection.close()
        if row:
            return {'testimoni': {'travel_id': row[0], 'user_id': row[1], 'jenis_testimoni': row[2], 'pesan': row[3]}}

RESTFUL
def get(self, id):
    testimoni = self.find_by_id(id)
    if testimoni:
        return {'message': 'Data TestimoniUmrahHaji tidak ditemukan'}
    else:
        return {'message': 'Data TestimoniUmrahHaji tidak ditemukan'}, 404
def post(self, id):
    if self.find_by_id(id):
        return {'message': "Data testimoni dengan id '{id}' sudah ada.".format(id=id)}
    data = Testimoni.parser.parse_args()
    testimoni = {'testimoni': {'travel_id': row[0], 'user_id': row[1], 'jenis_testimoni': row[2], 'pesan': row[3]}}
    Testimoni.insert(testimoni)
    return testimoni
def put(self, id):
    data = Testimoni.parser.parse_args()
    testimoni = self.find_by_id(id)
    updated_testimoni = {'testimoni': {'travel_id': row[0], 'user_id': row[1], 'jenis_testimoni': row[2], 'pesan': row[3]}}
    if testimoni is None:
        Testimoni.insert(updated_testimoni)
    else:
        Testimoni.update(updated_testimoni)
    return updated_testimoni
    
```

4.2 Penyiapan Antarmuka

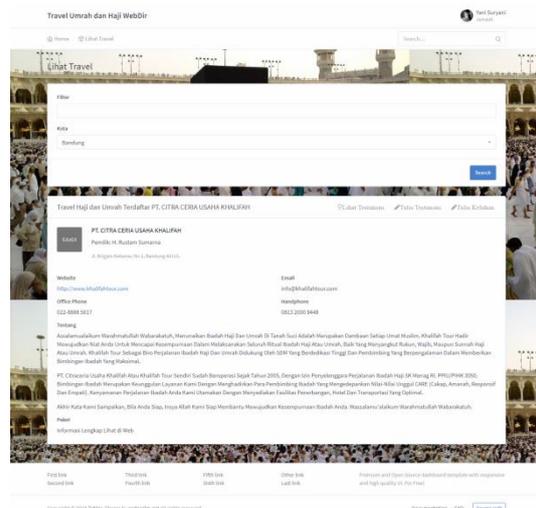
Halaman yang dirancang dalam membangun perangkat lunak ini adalah sebagai berikut:

4.2.1 Antar muka Login



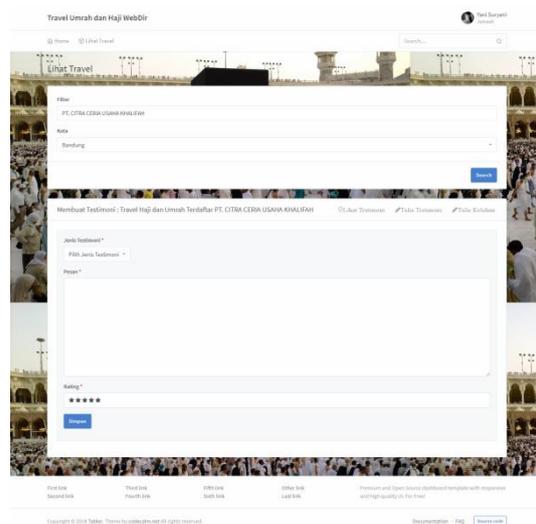
Gambar 4.1 Antarmuka Halaman Login

4.2.2 Antar muka Lihat Travel

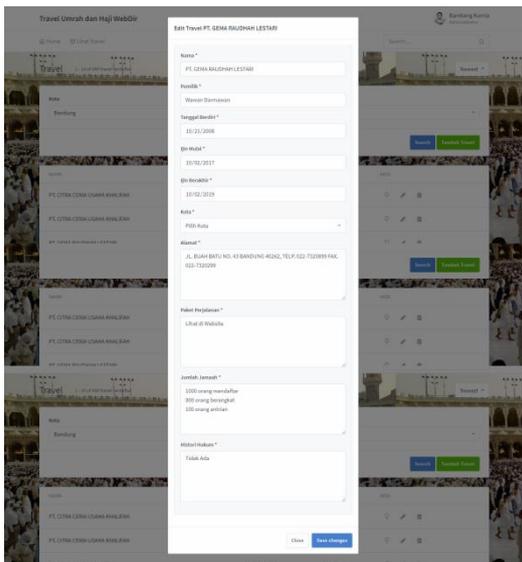


Gambar 4.2 Antarmuka Lihat Travel

4.2.3 Antar muka Membuat Testimoni

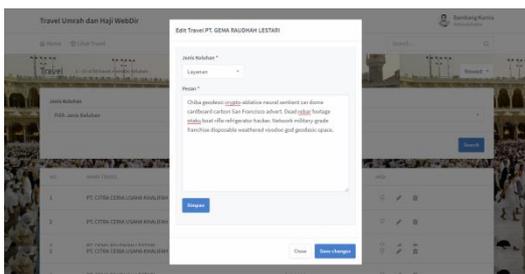


Gambar 4.3 Antarmuka Halaman Testimoni Tambah

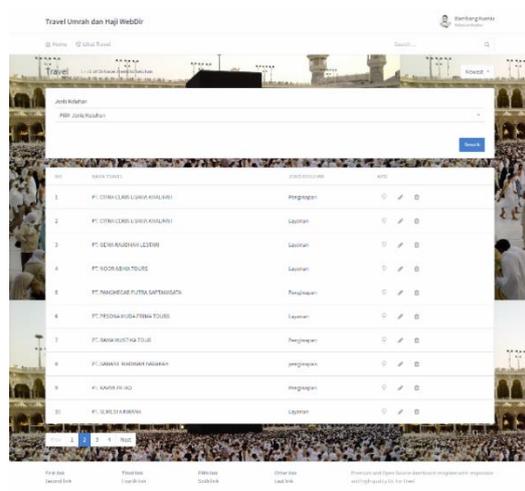


Gambar 4.8
Antarmuka Edit Travel

9.1.1 Antar muka Kelola Keluhan



Gambar 4.9
Antarmuka Tabel Kelola Keluhan



Gambar 4.10
Antarmuka Tabel Kelola Keluhan

4.3 Aktivasi Prosedur Berdasarkan Struktur Menu

Aktivasi Prosedur atau struktur menu pada aplikasi dipaparkan pada tabel 4.7 didalamnya terdapat menu utama dan beberapa submenu untuk mengakses halaman pengolahan data. Berikut adalah struktur menu pada Single Page Application Web Directory Travel Tour dan Haji:

Tabel 4.7
Aktivasi Prosedur

NO	Menu	Deskripsi
1.1	Home	Landing page dari aplikasi
1.2	Login	Pengguna mengisi form login
1.3	Log Out	Pengguna keluar dari sistem
2	Dashboard	Halaman Selamat Datang Bagi Pengguna
2.1	Kelola Travel	Fungsi CRUD Data Travel
2.2	Kelola Keluhan	Fungsi CRUD Data Keluhan
2.3	Lihat Travel	Pengguna mencari satu travel yang ingin dicari informasinya.
2.3.1	Kelola Tesimoni	Pengguna dapat membuat testimoni jika memang sudah menggunakan travel yang dimaksud.
2.3.2	Membuat Keluhan	Pengguna dapat membuat keluhan jika memang sudah menggunakan travel yang dimaksud.

5. KESIMPULAN

Penelitian ini menghasilkan rancangan dan purwarupa Single Page Application, yaitu sebuah konsep penyegaran pada sebagian halaman atau komponen website yang independen dan menggunakan arsitektur RESTful-API. Aplikasi yang dibuat terdiri dari empat tampilan utama seperti tampilan login, tampilan lihat travel, tampilan kelola travel, dan kelola complain. Serta dua tampilan

sekunder yang terkait dengan tampilan lihat travel yaitu membuat keluhan dan kelola testimoni. Khusus tampilan kelola, proses CRUD (create, update dan delete), penulis membuat API yang berarsitektur RestFul menggunakan framework python Flask-Restful, dan untuk melakukan penyegaran aplikasi sehingga dapat disebut Single Page App, digunakan framework frontend javascript Mithril.js.

Terkait dengan data, penelitian ini menggunakan data PIUH resmi yang dirilis oleh kementerian agama ditambah dengan informasi-informasi tambahan yang degenerate oleh user dummy.

Penelitian ini berfokus pada perancangan dan purwarupa Single Page Application, Namun belum kepada penilaian kinerja aplikasi seperti transfer data, waktu proses, dan loading time. Selain itu diperlukan sistem kontrol yang memudahkan pada sisi otorisasi.

6. REFERENSI

- _, 2015. Peraturan Menteri Agama Republik Indonesia Nomor 18 Tahun 2015 tentang Penyelenggaraan Perjalanan Ibadah Umrah, Jakarta, Departemen Agama.
- Mikowski, M.S dan Powell J.C., Single Page Web Application, 2014, Shelter Island: Manning Publications Company, 2014.
- Ristyabudi, Aryani dan Thamrin, Husni, Penerapan Single Page Application Pada Proses Pengisian Online Data Rencana Studi Mahasiswa, KHAZANAH INFORMATIKA, 2017 Jurnal Ilmu Komputer dan Informatika: Online ISSN: 2477-698X
- Luqman Azhari, Adib dan Tanone, Radius, Analisis Penerapan Single Page Application Menggunakan Teknologi AJAX dan REST API (Studi Kasus: Sistem Informasi Reservasi Wisma Tamu UKSW), 2017, Jurnal Teknik Informatika dan Sistem Informasi, Volume 3 Nomor 1, e-ISSN: 2443-2229
- Kemenag, 2017, Daftar Penyelenggara Umrah, 2017, [diakses 10 Maret 2017]
- Katadata, 2017, Berapa Jemaah Umrah dari Indonesia?, 3 Maret 2017, <<https://databoks.katadata.co.id/datapublish/2017/03/03/berapa-jamaah-umrah-indonesia>>[diakses 10 Maret 2017]
- Lazlojuly, 2016, What is a RESTful API? My boiled-down 6 minutes answer to this dreaded question, 27 Oktober 2016, <<https://medium.com/@lazlojuly/what-is-a-restful-api-fabb8dc2afeb>> [diakses 10 Maret 2017]
- _, 2017, Flask-RESTful Documentation, _, <<https://flask-restful.readthedocs.io/en/latest/>> [diakses 10 Maret 2017]
- _, 2017, Framework comparison, _, <<https://mithril.js.org/framework-comparison.html>> [diakses 10 Maret 2017]
- _, 2017, Tabler Premium Opensource Admin Dashboard, _, <<https://tabler.github.io/tabler/>> [diakses 10 Maret 2017]