

Vol.1 No.1 | Februari 2019 | p-ISSN 2655-867X
e-ISSN 2655-8661

SisInfo

Jurnal Sistem Informasi dan Informatika



Penerbit :

**FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS INFORMATIKA DAN BISNIS INDONESIA**

DAFTAR ISI

Analisis Faktor-Faktor Yang Mempengaruhi Penggunaan <i>E-Marketplace</i> Shopee Graha Prakarsa	1 – 11
Sistem Informasi Tempat Kost Berbasis Webservice Di Sekitar Universitas Islam Nusantara Menggunakan Metode SAW (<i>Simple Additive Weighting</i>) Hendriyana, Ripal Maulana	12 – 18
Pengembangan Aplikasi Lagu Daerah Dan Nasional Berbasis Android Ivan Michael Siregar	19 – 24
Studi Komparasi Algoritma Similaritas Pada Prediksi Rating Berbasis Item Pada <i>Collaborative Filtering</i> Studi Kasus Pada <i>Data Review</i> Restoran Mochamad Iqbal Ardimansyah	25 – 29
Pengembangan Media Pembelajaran Limit Fungsi Berbasis Multimedia Untuk Sekolah Menengah Atas Kelas XI Marwondo1, R. Yadi Rakhman A, Wilner Saut Lamhot	30 – 40
Aplikasi Pengolahan Data Perusahaan Berbasis Web Dengan Menggunakan <i>Framework Codeigniter</i> (Studi Kasus Cv. Prima Nusa) Tarsinah Sumarni	41 – 48
Sistem Informasi Manajemen Bantuan Untuk Korban Bencana Alam Berbasis Web Titan Parama Yoga, Iis Ismail	49 – 58
Model Sistem Pendukung Keputusan Transportasi melalui Metode Saving Matrix Pada CV XYZ Tombak Gapura Bhagya	59 – 68
Rancang Bangun Single Page Application Berbasis Framework Laravel Dan Elm (Studi Kasus E-Job XYZ) Trisna Gelar Abdullah, Budiman	69 – 78

RANCANG BANGUN *SINGLE PAGE APPLICATION* BERBASIS FRAMEWORK LARAVEL DAN ELM (STUDI KASUS E-JOB XYZ)

Trisna Gelar Abdillah¹, Budiman²

Fakultas Teknologi dan Informatika, Universitas Informatika dan Bisnis Indonesia

trisnagelar@unibi.ac.id, budiman@unibi.ac.id

ABSTRAK

E-Job merupakan aplikasi lowongan kerja online yang memberikan layanan informasi lowongan pekerjaan kepada para pencari kerja. Aplikasi ini bertujuan untuk membantu pencari kerja berdasarkan kualifikasi yang diberikan oleh pembuka lowongan kerja atau perusahaan. Artikel ini memaparkan rancang bangun *single page application* berbasis framework laravel dan elm (studi kasus e-job xyz). Single App Application adalah pengembangan web yang hanya menggunakan satu halaman saja. Artinya, meskipun pengguna beralih ke menu lain, pada URL tidak menunjukkan perubahan halaman. Hal ini dapat diimplementasikan dengan sebuah sistem backend berbasis RestFulAPI dan sistem frontend yang bersifat reaktif. Pada penelitian ini penulis menggunakan Laravel, sebuah Web Framework berbasis PHP yang memudahkan pengembang aplikasi untuk membangun sebuah arsitektur web berbasis RESTFul API, dilengkapi dengan fitur keamanan akses data menggunakan otorisasi berbasis Oauth2 sebagai sistem backend sedangkan Bahasa pemrograman ELM atau arsitektur ELM digunakan untuk membangun sistem frontend pada SPA E-job.

Kata Kunci : Single Page Application, Laravel, dan ELM

ABSTRACT

E-Job is an online job application that provides job vacancy information services to job seekers. This application aims to help job seekers based on their qualifications given by the opening job or company. This article describes the design of a single page application based on the laravel and elm framework (e-job xyz as case study). Single App Application is a web development that uses only one page. That is, even if the user switches to another menu, the URL does not show page changes. This can be implemented with a RestFulAPI-based backend system and a reactive programming frontend system. In this research the author uses Laravel, a PHP-based Web Framework that makes it easy for application developers to build a RESTFul API-based web architecture, equipped with data access security features using Oauth2-based authorization as a backend system. ELM programming language or ELM architecture is used to build frontend systems on this application.

Keywords: Single Page Application, Laravel, and ELM

1. PENDAHULUAN

1.1. Latar Belakang Masalah

Perkembangan teknologi informasi saat ini semakin berkembang pesat, hal ini menunjukkan tingginya minat masyarakat menggunakan layanan teknologi informasi. Teknologi informasi menjadi kebutuhan hidup masyarakat sehari-hari dari berbagai kalangan yang akan memudahkan masyarakat untuk mencari berbagai informasi yang dibutuhkan dengan cepat.

E-Job merupakan aplikasi lowongan kerja online yang memberikan layanan informasi lowongan pekerjaan kepada pencari kerja. Aplikasi ini bertujuan untuk membantu pencari kerja berdasarkan kualifikasi yang diberikan oleh pembuka lowongan kerja atau perusahaan. Berdasarkan data Badan Pusat Statistik yang dirilis pada tanggal 7 Mei 2018 bahwa jumlah angkatan kerja pada Februari 2018 sebanyak 133,94 juta orang, naik 2,39 juta orang dibanding Februari 2017. Sejalan dengan itu, Tingkat Partisipasi Angkatan Kerja (TPAK) sebesar 69,20 persen, meningkat 0,18 persen poin. Dalam setahun terakhir, pengangguran berkurang 140 ribu orang, sejalan dengan Tingkat Pengangguran Terbuka (TPT) yang turun menjadi 5,13 persen pada Februari 2018. Dilihat dari tingkat pendidikan, TPT untuk Sekolah Menengah Kejuruan (SMK) tertinggi diantara tingkat pendidikan lain, yaitu sebesar 8,92 persen.

Berdasarkan permasalahan diatas maka peneliti akan membuat penelitian dengan judul “Rancang Bangun *Single Page Application* berbasis framework Laravel dan Elm (studi kasus e-job xyz)”. Penelitian ini diharapkan dapat membantu menurunkan jumlah angka pengangguran di Indonesia.

1.2. Rumusan Masalah

Berdasarkan latar belakang di atas, maka dapat dirumuskan permasalahan yaitu bagaimana merancang dan membangun aplikasi E-Job berbasis *Single Page Application*.

1.3. Batasan Masalah

Batasan masalah dari sistem yang dibahas sebagai berikut:

1. Sistem ini membahas informasi lowongan kerja
2. Sistem ini membahas pembukaan lowongan kerja oleh perusahaan
3. Sistem ini tidak membahas proses seleksi bagi pencari kerja
4. Sistem ini berbasis Single Page Application



Gambar 1.1 Ketenagakerjaan Indonesia Februari 2018
Sumber: <https://www.bps.go.id/website/images/2018-Feb-Tenaga-Kerja-ind.png>

1.4. Tujuan Penelitian

Tujuan dari penelitian ini adalah merancang dan membangun aplikasi E-Job berbasis Single Page Application.

2. TINJAUAN PUSTAKA

2.1. Rancang Bangun

Menurut Sutabri (2005) perancangan sistem adalah penentuan proses dan data yang diperlukan oleh sistem baru. Jika sistem itu berbasis komputer, rancangan dapat menyertakan spesifikasi jenis peralatan yang akan digunakan. Sedangkan Jogiyanto (2001) menjelaskan bahwa perancangan sistem dapat didefinisikan sebagai gambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisahkan kedalam satu kesatuan yang utuh dan berfungsi. Dari penjelasan diatas dapat ditarik kesimpulan bahwa

rancang bangun merupakan kegiatan hasil analisa yang dapat di implementasikan ke dalam perangkat lunak untuk menggantikan atau memperbaiki sistem yang sudah ada.

2.2. Aplikasi

Menurut Safaat H (2012: 9) Perangkat lunak aplikasi adalah suatu subkelas perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan suatu tugas yang diinginkan pengguna. Biasanya dibandingkan dengan perangkat lunak sistem yang mengintegrasikan berbagai kemampuan komputer, tapi tidak secara langsung menerapkan kemampuan tersebut untuk mengerjakan suatu tugas yang menguntungkan pengguna. Contoh utama perangkat lunak aplikasi adalah pengolah kata, lembar kerja, dan pemutar media. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan Open Office.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan setiap aplikasi. Sering kali, aplikasi ini memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna.

2.3. Single Page Application

Single Page Application merupakan suatu aplikasi berbasis web yang hanya menggunakan satu halaman saja. Artinya, meskipun pengguna beralih ke menu lain, pada URL tidak menunjukkan perubahan halaman. Dalam Single Page Application, semua komponen seperti CSS, gambar, skrip dan resources lain yang diperlukan dimuat pada satu waktu di halaman utama dan kemudian konten yang sesuai atau komponen dimuat secara dinamis tergantung pada interaksi atau permintaan pengguna (Jadhav et al, 2015). Server hanya perlu mengirim dan menyegarkan bagian tertentu sesuai permintaan pengguna sehingga diharapkan dapat menghemat bandwidth.

Selain itu menurut Mikowski dan Powell (2014), Single Page Application dapat memberikan responsivitas yang tinggi dan kesenangan bagi pengguna, tidak membingungkan dan mengganggu pengguna.

2.4. Laravel sebagai Backend(RestFul API)

Laravel adalah sebuah framework berbasis PHP yang dibangun oleh Taylor Ottwell pada tahun 2010. Menurut Stauffer Matt(2016: 5), seorang developer akan merasa bahagia jika membangun sebuah aplikasi menggunakan laravel karena dapat mempermudah pekerjaannya. Terdapat empat kemudahan yang didapatkan seorang pengembang aplikasi jika menggunakan laravel yakni, mudah dipelajari, simpel, memiliki API yang konsisten, dan arsitektur/struktur yang dapat diprediksi untuk seluruh modul dalam framework.

Dalam membangun single app application adalah diperlukan sebuah backend yang berarsitektur RESTful API, yang terdiri dari empat metode yaitu GET, POST, PUT, dan Delete dan memungkinkan pengimplementasian CRUD (Create, Update dan Delete). Untungnya, arsitektur ini dapat dibangun secara mudah menggunakan laravel. Implementasi CRUD, dapat dengan mudah dibangun karena laravel mengadopsi design pattern Model View Controller(MVC) yang memisahkan logika, data dan persentasi. Dalam hal pembuatan API hanya Model dan Controller dari sebuah kelas saja yang dibangun, ditambah dengan koneksitas API tersebut dihubungkan dengan “*api routes*”, dan selain itu keamanan mengakses elemen API dapat ditingkatkan dengan penggunaan passport sebuah library yang berbasis Oauth2, digunakan untuk autentikasi API yang dibangun. Dalam penelitian ini kami akan membangun sistem backend E-Job menggunakan framework Laravel beserta library-library pendukung.

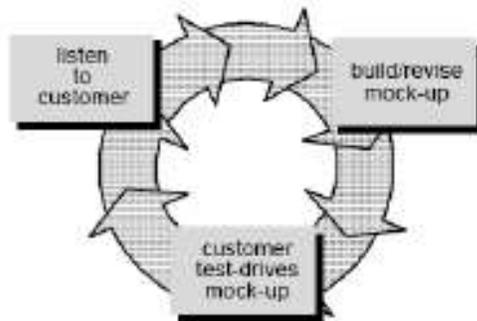
2.5. ELM sebagai Frontend

Menurut Czaplicki, Evan (2012: 3), Elm dibangun sebagai bahasa untuk *Functional Reactive Programming* (FRP) pada pembuatan sistem GUI. FRP adalah sebuah cara deklaratif untuk membuat

sebuah sistem yang reaktif. Bahasa deklaratif memungkinkan pengembang program menyatakan “apa” yang akan dibuat tanpa mengetahui “bagaimana” cara komputer menghasilkan hal tersebut. Dengan cara ini, hal-hal detail yang tidak relevan diserahkan kepada compiler, memudahkan pengembang memikirkan hal yang levelnya lebih tinggi. Namun seiring berjalannya pengembangan bahasa ELM, istilah FRP mulai ditinggalkan, mulai versi 0.17 diubah dengan Arsitektur ELM, yang merupakan bahasa fungsional yang berbasis concurrency. Dalam penelitian ini kami akan membangun sistem frontend E-Job menggunakan bahasa ELM yang mengambil API dari Laravel.

2.6. Metode Pengembangan Sistem

Menurut Pressman (2002) mengemukakan bahwa Prototyping Paradigma dimulai dengan mengumpulkan kebutuhan. Pengembang dan pelanggan bertemu dan mendefinisikan obyektif keseluruhan sistem atau perangkat lunak yang akan dibuat, mengidentifikasi segala kebutuhan yang diketahui, dan area garis besar dimana definisi lebih jauh merupakan keharusan kemudian dilakukan “perancangan kilat”. Perancangan kilat berfokus pada penyajian dari aspek-aspek perangkat lunak tersebut yang akan nampak bagi pelanggan atau pemakai.



Gambar 2.1 Metode Pengembangan Sistem Prototype Tahap-tahap pengembangan Prototype model menurut Pressman (2002), adalah

1. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari system dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu system yang sesuai kebutuhan, maka harus diketahui terlebih

dahulu bagaimana system yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

2. Merancang dan membuat Prototype

Pada tahap ini, dilakukan perancangan dan pembuatan prototype system. Prototype yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna

3. Uji coba

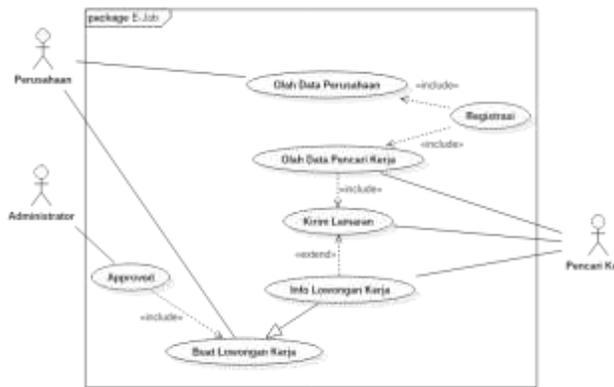
Pada tahap ini, Prototype dari system di uji coba oleh pelanggan atau pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki Prototype yang ada.

3. HASIL DAN PEMBAHASAN

3.1. Perancangan Sistem



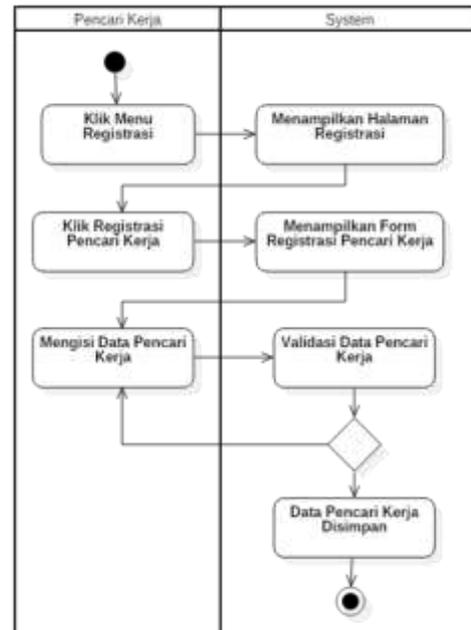
Gambar 3.1 arsitektur Sistem Aplikasi E-Job Gambar Diatas merupakan gambar arsitektur purwarupa Sistem Aplikasi E-Job dengan tiga pengguna yaitu pencari kerja, perusahaan dan administrator.



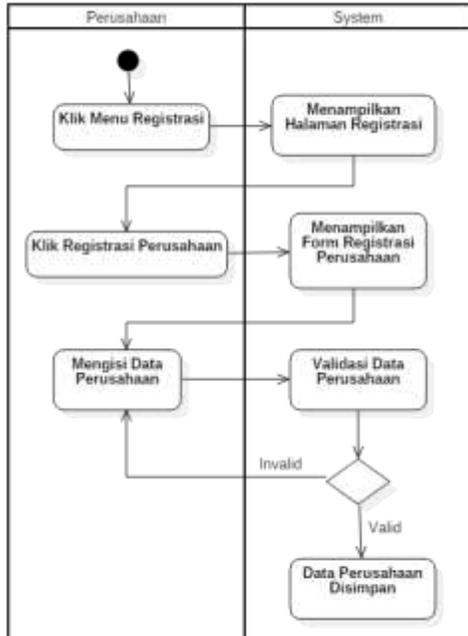
Gambar 3.2 Use Case Diagram E-Job

Untuk pengguna aplikasi E-Job, pencari kerja dan perusahaan harus melakukan proses Registrasi. Perusahaan dapat mengajukan pembukaan lowongan kerja dengan kualifikasi yang dibutuhkan, kemudian administrator akan melakukan evaluasi dan approved untuk dipublikasi sehingga pencari kerja dapat melihat informasi lowongan kerja.

registrasi dengan mengisi data registrasi yang telah disediakan, kemudian sistem akan melakukan validasi dan menyimpan data perusahaan pembuka lowongan kerja.



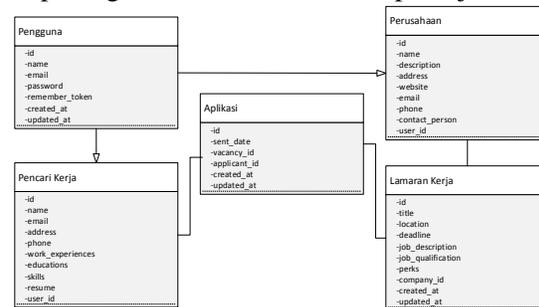
Gambar Activity 3.4 Diagram Registrasi Pencari Kerja



Gambar Activity 3.3 Diagram Registrasi Perusahaan

Pengguna sistem dari sisi perusahaan yang akan membuka lowongan kerja harus melakukan

Pengguna sistem dari sisi pencari kerja yang akan melamar harus melakukan proses registrasi dengan mengisi data pencari kerja atau Curriculum Vitae yang disediakan, kemudian sistem akan melakukan validasi dan menyimpan data pencari kerja yang dapat digunakan untuk melamar pekerjaan.



Gambar 3.5 Class Diagram E-Job

Gambar Diatas merupakan gambar Class diagram dari purwarupa Sistem Aplikasi E-Job, dengan lima kelas utama, yaitu pengguna, pencari kerja, perusahaan, lamaran kerja dan aplikasi. Kelas

Pencari kerja dan Perusahaan adalah turunan dari Kelas Pengguna. Kelas pengguna digunakan untuk proses autentikasi sistem backend dan frontend, memiliki informasi dasar seperti username dan password. Sedangkan Perusahaan dan Pencari Kerja, memiliki atribut tambahan yang khusus untuk kelas-kelas tersebut. Kelas Lamaran kerja berasosiasi dengan Kelas Perusahaan, yakni satu kebanyak, satu perusahaan dapat memiliki banyak lowongan kerja. Sedangkan kelas pencari kerja dan lowongan kerja berasosiasi banyak ke banyak di mediasi oleh kelas aplikasi.

3.2. Perancangan Backend Menggunakan Laravel REST API

3.2.1. CRUD Perusahaan, Pelamar, Lowongan dan Aplikasi

Pada pembuatan CRUD untuk Kelas Perusahaan, Pelamar, Lowongan dan Aplikasi. Penulis menggunakan migration dan seeder menggunakan faker untuk meng-generate data dummy, untuk membangun database, model untuk pengolahan data, dan controller untuk tata-kelola pertukaran API, disini seluruh data dikeluarkan dalam bentuk JSON(*Javascript Object Notation*) yang akan diakses oleh ELM Frontend.

Berikut adalah contoh rancangan migration, model dan controller pada kelas companies dan vacancies.

```

1 <?php
2 use Illuminate\Support\Facades\Schema;
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5 class CreateCompaniesTable extends Migration
6 {
7     public function up()
8     {
9         Schema::create('companies', function (Blueprint $table) {
10             $table->increments('id');
11             $table->string('name', 100);
12             $table->longText('description');
13             $table->text('address');
14             $table->string('website', 100);
15             $table->string('email', 100);
16             $table->string('phone', 30)->nullable();
17             $table->string('contact_person', 50);
18             $table->timestamps();
19         });
20     }
21     public function down()
22     {
23         Schema::dropIfExists('companies');
24     }
25 }

```

Gambar 3.7 Companies Migration

```

1 <?php
2 use Faker\Generator as Faker;
3 $factory->define(App\Company::class, function (Faker $faker) {
4     return [
5         'name' => $faker->company,
6         'description' => $faker->text,
7         'address' => $faker->address,
8         'website' => $faker->domainName,
9         'email' => $faker->safeEmail,
10        'phone' => $faker->phoneNumber,
11        'contact_person' => $faker->name,
12    ];
13 });
14

```

Gambar Companies Factory

```

1 <?php
2 namespace App\Http\Controllers;
3 use Illuminate\Http\Request;
4 use App\Company;
5 use Validator;
6 class CompaniesController extends Controller
7 {
8     public function index()
9     {
10        return response()->json(Company::paginate(), 200);
11    }
12    public function store(Request $request)
13    {
14        $rules = [];
15        $validator = Validator::make($request->all(), $rules);
16        if($validator->fails()){
17            return response()->json($validator->errors(), 400);
18        }
19        $company = Company::create($request->all());
20        return response()->json($company, 201);
21    }
22    public function show($id)
23    {
24        $company = Company::find($id);
25        if(is_null($company)){
26            return response()->json(null, 404);
27        }
28        return response()->json($company, 200);
29    }
30    public function update(Request $request, $id)
31    {
32        $company = Company::whereId($id)->update($request->all());
33        return response()->json($company, 201);
34    }
35 }

```

Gambar 3.8 Companies Controller

```

1 <?php
2 use Illuminate\Support\Facades\Schema;
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5 class CreateVacanciesTable extends Migration
6 {
7     public function up()
8     {
9         Schema::create('vacancies', function (Blueprint $table) {
10             $table->increments('id');
11             $table->string('title', 100);
12             $table->string('location', 100);
13             $table->dateTime('deadline');
14             $table->longText('job_description');
15             $table->text('job_qualification');
16             $table->text('perks');
17             $table->timestamps();
18             $table->integer('company_id')->unsigned();
19             $table->foreign('company_id')->references('id')->on('companies')->onDelete('cascade');
20         });
21     }
22     public function down()
23     {
24         Schema::dropIfExists('vacancies');
25     }
26 }

```

Gambar 3.9 Vacancies Migration

```

1 <?php
2 use Faker\Generator as Faker;
3 use App\Company;
4 $factory->define(App\Vacancy::class, function (Faker $faker) {
5     $companies = Company::all()->pluck('id')->toArray();
6     return [
7         'title' => $faker->sentence($nbWords = 3, $variableNbWords = true),
8         'location' => $faker->city,
9         'deadline' => $faker->dateTime($max = 'now', $timezone = null),
10        'job_description' => $faker->paragraph($nbSentences = 3, $variableNbSentences = true),
11        'job_qualification' => $faker->paragraph($nbSentences = 3, $variableNbSentences = true),
12        'perks' => $faker->paragraph($nbSentences = 3, $variableNbSentences = true),
13        'company_id' => $faker->randomElement($companies),
14    ];
15 });

```

Gambar 3.10 Vacancies Factory

```

namespace App\Http\Controllers;
use Illuminate\Http\Request;
use App\Vacancy;
class VacanciesController extends Controller
{
    public function index()
    {
        return response()->json(Vacancy::paginate(),200);
    }
    public function store(Request $request)
    {
        $rules = [];
        $validator = Validator::make($request->all(), $rules);
        if($validator->fails()){
            return response()->json($validator->errors(), 400);
        }
        $vacancy = Vacancy::create($request->all());
        return response()->json($vacancy, 201);
    }
    public function show($id)
    {
        $vacancy = Vacancy::find($id);
        if(is_null($vacancy)){
            return response()->json(null, 404);
        }
        return response()->json($vacancy, 200);
    }
    public function update(Request $request, $id)
    {
        $vacancy = Vacancy::whereId($id)->update($request->all());
        return response()->json($vacancy, 201);
    }
    public function destroy(Request $request, Vacancy $vacancy)
    {
        $vacancy->delete();
        return response()->json(null, 204);
    }
}

```

Gambar 3.11 Vacancies Controller

3.2.2. REST-API

Laravel secara otomatis dapat membuat arsitektur RESTFul-API dengan mendaftarkan controller yang diinginkan dengan memanggil `Route::apiResource('nama', 'Controller')` kedalam file `"route/api.php"`. Penulis mendaftarkan kelas `Companies`, `Vacancies`, `Applicants`, dan `Applications` sebagai berikut:

```

<?php
use Illuminate\Http\Request;
Route::middleware('auth:api')->get('/user', function (Request $request) {
    return $request->user();
});
Route::apiResource('companies', 'CompaniesController');
Route::apiResource('vacancies', 'VacanciesController');
Route::apiResource('applicants', 'ApplicantsController');
Route::apiResource('applications', 'ApplicationsController');
Route::get('companies/{company}/vacancies', 'CompaniesController@vacancies');

```

Gambar 3.12 Route API

Untuk mengamankan pemanggilan route, digunakan sebuah library `passport` yang dapat mengautentikasi konsumsi sebuah API menggunakan skema `Oauth2`.

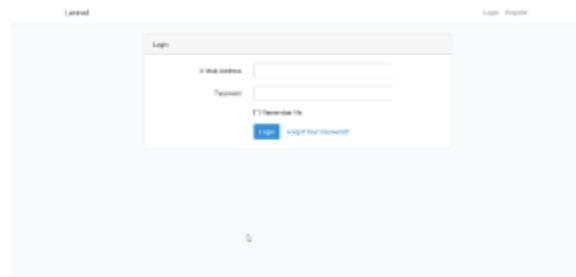
3.3. Perancangan Frontend Menggunakan Arsitektur ELM

Pada penelitian ini frontend yang dikembangkan tidak murni elm, untuk mengaktifkan `Outh2` client agar dapat berkomunikasi dengan server, dalam hal ini digunakan sebuah website baru berbasis laravel hanya untuk keperluan frontend saja dibantu oleh sebuah template blade yang didalamnya terdapat sebuah tag `"id"` yang bernama `"elm-content"`.

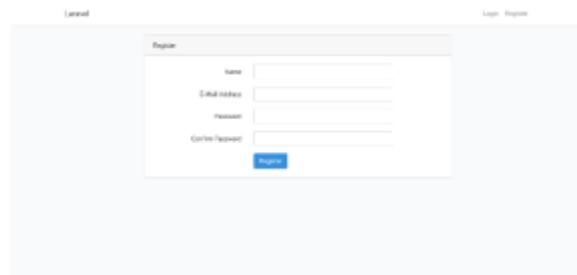
Menurut `Bijan Boustani (2018: 127)` untuk membuat sebuah elm arsitektur diperlukan lima struktur file yaitu `Main`, `Model`, `Update`, `Subscriptions` dan `View`. File `Model` digunakan untuk membuat struktur dari data pada masing-masing kelas (`Pelamar`, `Lowongan Kerja` dan `Perusahaan`), dalam hal ini adalah struktur yang digunakan untuk membuat format data `JSON`. `Update` digunakan untuk melakukan perubahan data dari `Model`, dan menampilkan pesan. `Subscriptions` digunakan untuk menangani penggunaan event dari inputan user seperti `click`, `hover`, dll. `View` digunakan untuk menyimpan `DOM`, `HTML` dan `CSS` untuk menampilkan data ke user.

3.4. Rancangan Antarmuka

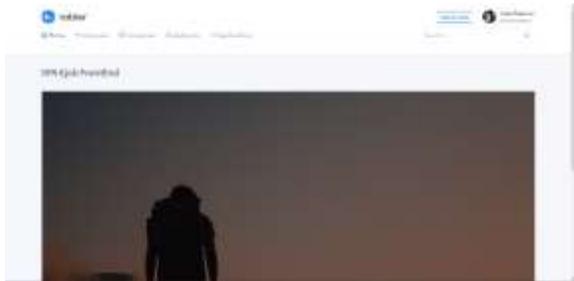
Rancangan antarmuka dari perangkat lunak ini adalah sebagai berikut:



Gambar 3.13 Antar Muka Login



Gambar 3.14 Antar Muka Register



Gambar 3.15 Halaman Utama



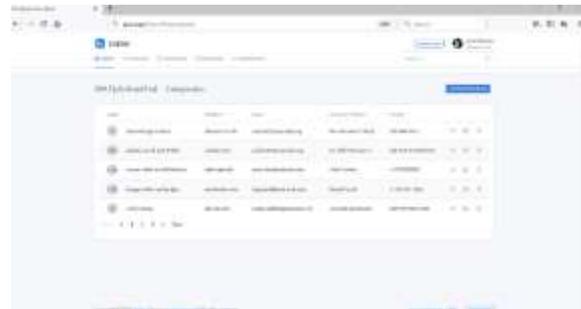
Gambar 3.16 Halaman Lowongan Pekerjaan



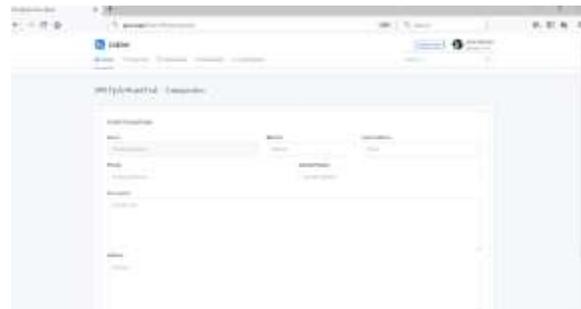
Gambar 3.17 Halaman Tambah Lowongan Pekerjaan



Gambar 3.18 Halaman Ubah Lowongan Pekerjaan



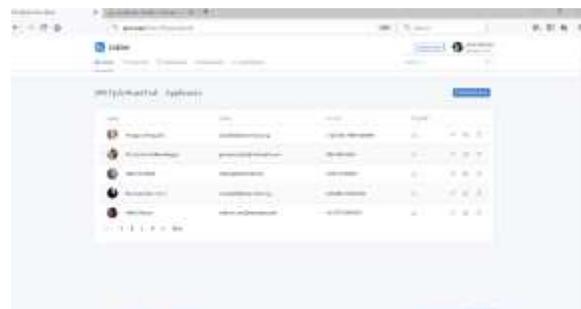
Gambar 3.19 Halaman Perusahaan



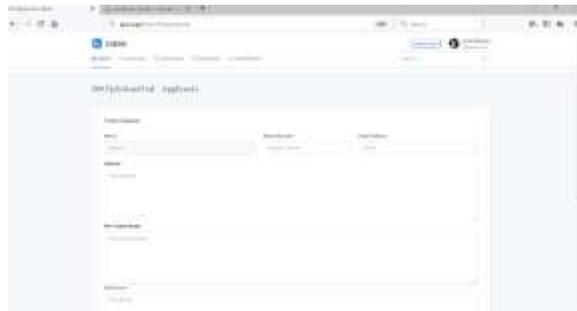
Gambar 3.20 Halaman Tambah Perusahaan



Gambar 3.21 Halaman Ubah Perusahaan



Gambar 3.22 Halaman Pelamar



Gambar 3.23 Halaman Tambah Pelamar



Gambar 3.23 Halaman Ubah Pelamar

3.5. Aktivasi Prosedur Berdasarkan Struktur Menu

Aktivasi Prosedur atau struktur menu pada aplikasi dipaparkan pada tabel didalamnya terdapat menu utama dan beberapa submenu untuk mengakses halaman pengolahan data.

Tabel 3.1 Aktivasi Prosedur

NO	Menu	Deskripsi
1.1	Home	Landing page dari aplikasi
1.2	Login	Pengguna mengisi form login
1.3	Register	
1.4	Logout	Pengguna keluar dari sistem
2.1	Companies Table	Pengguna melihat perusahaan yang ingin dicari informasinya.
2.2	Add Company	Fungsi Tambah Data Perusahaan
2.3	Edit Company	Fungsi Edit Data Perusahaan
2.4	Delete Company	Fungsi Delete Data Perusahaan
3.1	Vacancies Table	Pengguna melihat lowongan pekerjaan yang ingin dicari informasinya.
3.2	Add Vacancy	Fungsi Tambah Data Lowongan Pekerjaan
3.3	Edit Vacancy	Fungsi Edit Data Lowongan Pekerjaan
3.4	Delete Vacancy	Fungsi Delete Data Lowongan Pekerjaan

NO	Menu	Deskripsi
4.1	Applicants Table	Pengguna melihat pelamar yang ingin dicari informasinya.
4.2	Add Applicant	Fungsi Tambah Data Pelamar Kerja
4.3	Edit Applicant	Fungsi Edit Data Pelamar Kerja
4.4	Delete Applicant	Fungsi Delete Data Pelamar Kerja

4. KESIMPULAN DAN SARAN

4.1. Kesimpulan

Penelitian ini menghasilkan sebuah aplikasi E-job yang menggunakan Laravel Framework sebagai sistem RESTFul API, dan Kombinasi dari View Laravel dan Arsitektur Elm sebagai frontend yang mengkonsumsi API tersebut dalam kerangka Single Page Application. Dalam aplikasi tersebut terdiri dari empat tiga menu yakni Perusahaan, Lowongan Kerja, dan Pencari Kerja, proses CRUD(create, update dan delete) dari ketiga menu tersebut dapat diakses dalam satu halaman.

4.2. Saran

Pada pengembangan aplikasi E-Job berbasis SPA ini terdapat dua hal teknis yang dapat ditambahkan untuk pengembangan aplikasi lebih lanjut. Pertama adalah penggunaan JSON Web Token sebagai pengganti Oauth2 untuk autentikasi RestFul API dan Kedua adalah penggunaan ExpressJS sebagai server frontend

digunakan untuk mengkonsumsi Restful API dan dikombinasikan dengan arsitektur ELM.

6. REFERENSI

Jadhav, M. A., Sawant, B. R. & Deshmukh, A. (2015). Single Page Application using AngularJS. *International Journal of Computer Science and Information Technologies*, 6 (3): 2876.

Jogiyanto. 2001. *Analisis & Desain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Andi. Yogyakarta.

Mikowski M. S. & Powell, J. C. (2014). *Single Page Web Application*. Shelter Island: Manning Publications Company.

Pressman, Roger S. 2002. *Rekayasa Perangkat Lunak Pendekatan Praktisi (BUKU SATU)*. Andi. Yogyakarta

Safaat H. Nazruddin. 2012 (Edisi Revisi). *Pemrograman Aplikasi Mobile. Smartphone dan Tablet PC Berbasis Android*. Informatika. Bandung.

Sutabri, Tata. 2005. *Analisis Sistem Informasi*. Andi. Yogyakarta.

<https://www.bps.go.id/pressrelease/2018/05/07/1484/februari-2018--tingkat-pengangguran-terbuka--tpt--sebesar-5-13-persen--rata-rata-upah-buruh-per-bulan-sebesar-2-65-juta-rupiah.html> tanggal akses: 3 September 2018. Jam 13.10 WIB.

Stauffer, Matt, 2016, *Laravel: up and running: a framework for building modern PHP apps*. O'Reilly Media, Inc.,

Awaludin, Rahmat, 2016. *Menyelami framework laravel*, Lean Publishing, Victoria, British Columbia, Canada.

Gilmore, Jason, 2018, *Easy Laravel 5*, Lean Publishing, Victoria, British Columbia, Canada.

Czaplicki, Evan. "Elm: Concurrent frp for functional guis." Senior thesis, Harvard University (2012).

Loder, Wolfgang, 2018, *Web Applications with Elm*, Apress, New York, Amerika.

Fairbank, Jeremy, 2018, *Programming Elm Build Safe and Maintainable Front-End Applications Beta Book*, Pragmatic Programmer LLC, PragProg.

<https://elm-lang.org/blog/farewell-to-frp>, tanggal akses: 3 September 2018. Jam 19.10 WIB.

Boustani, Bijan, *Elixir and Elm Tutorial*, 2018, Lean Publishing, Victoria, British Columbia, Canada.