

## PEMBANGUNAN APLIKASI PENERIMAAN KARYAWAN BARU BERBASIS *WEBSITE* MENGGUNAKAN *FRAMEWORK LARAVEL* (STUDI KASUS: PT. XYZ)

Gabrielle Beauty Linelejan<sup>a</sup>, Nina Setiyawati<sup>b</sup>

<sup>ab</sup>Fakultas Teknologi Informasi Universitas Kristen Satya Wacana, Salatiga 50711

<sup>a</sup> [672019299@student.uksw.edu](mailto:672019299@student.uksw.edu) , <sup>b</sup> [nina.setiyawati@uksw.edu](mailto:nina.setiyawati@uksw.edu)

### ABSTRAK

Program Aplikasi merupakan salah satu sistem yang sering digunakan oleh banyak instansi atau perusahaan pada zaman yang semakin maju, baik itu instansi pemerintah atau swasta dalam menjalankan operasional kerja setiap hari terkait dengan penyajian data yang cepat dan akurat. Perusahaan PT. XYZ masih menggunakan metode konvensional dalam perekrutan karyawan dan pengelolaan data maka dari itu akan dibangunlah sebuah sistem berbasis *website* untuk membantu dunia kerja dalam menghadapi persaingan dalam era globalisasi saat ini. Metode yang digunakan pada program ini menggunakan metode *waterfall model*. Penelitian ini akan menghasilkan sistem rekrutmen dan pengolahan data karyawan yang memanfaatkan perkembangan teknologi agar membantu perusahaan dan calon karyawan agar semua tahapan berjalan lebih efektif dan efisien.

**Kata kunci:** Rekrutmen Karyawan, Seleksi, PHP, MySQL, Web

### ABSTRACT

*The Application Program is a system that is often used by many agencies or companies in increasingly advanced times, be it government or private agencies, in carrying out daily work operations related to fast and accurate data presentation. Company PT. XYZ still uses conventional methods in recruiting employees and managing data, therefore a website-based system will be built to help the world of work face competition in the current era of globalization. The method used in this program uses the waterfall model method. This research will produce a recruitment and employee data processing system that utilizes technological developments to help companies and prospective employees so that all stages run more effectively and efficiently.*

**Keywords:** Employee Recruitment, Selection, PHP, MySQL, Website

## 1. PENDAHULUAN

Dalam perkembangan teknologi yang semakin pesat, hampir setiap sektor memanfaatkan teknologi, contohnya penggunaan teknologi pada proses perekrutan sumber daya manusia. Para pelamar masih ada yang mencari informasi pekerjaan dengan mengunjungi kantor untuk melihat papan pengumuman yang berisi lowongan pekerjaan, melakukan pencarian melalui media cetak atau mencari informasi dari mulut ke mulut [1]. Dengan adanya

aplikasi ini diharapkan dapat membantu perusahaan dan pelamar agar semuanya dapat berjalan lebih efektif dan efisien [2]. *Internet* akan membantu sistem perekrutan ini berjalan secara cepat dan tepat guna mendapatkan karyawan yang sesuai dengan kebutuhan perusahaan diharapkan pengguna yang mengakses *internet* harus siap selalu terhubung dan terkoneksi dengan jaringan *internet* agar bisa memaksimalkan sumber daya teknologi informasi berbasis *internet* tersebut [3].

PT. XYZ adalah perusahaan yang bergerak dibidang distribusi. Proses penerimaan karyawan baru masih terdapat kekurangan diantaranya pendataan biodata calon karyawan serta penilaian hasil tes yang masih menggunakan cara manual, sehingga terjadilah sistem administrasi yang membutuhkan waktu yang cukup panjang dalam mengelolah data-data yang masuk [4]. Selain itu kurangnya media promosi bisa dikatakan kantor dari perusahaan ini berada di sebuah kabupaten yang kecil sehingga para pelamar kesulitan untuk mendapatkan informasi terbaru mengenai perkembangan apa saja yang ada diperusahaan tersebut mengenai perekrutan karyawan baru. Aplikasi perekrutan karyawan baru berbasis *website* ini diharapkan dapat mempermudah proses penyeleksian calon karyawan yang akan masuk dalam perusahaan, khususnya dalam divisi HRD dalam pengelolaan data para calon karyawan dan juga mempermudah untuk mendapatkan suatu informasi mengenai lowongan yang dibuka dari perusahaan tersebut [5]. Adapun *website* ini dibangun dengan menggunakan bahasa pemrograman PHP dan *framework Laravel* agar tampilan dari *website* tersebut menarik dan *responsive*.

Pada penelitian ini menggunakan beberapa penelitian terdahulu sebagai bahan acuan dan perbandingan serta memiliki keterkaitan masalah dengan penelitian yang dilakukan. Dari penelitian Sonty Lena dan Lintang Sangita memakai metode diskriptif kuantitatif agar dapat mengetahui proses rekrutmen para pelamar, lalu sistem tersebut diimplementasikan dalam *prototype* lalu dilakukan pengujian *prototype* dan mendapatkan hasil Web Server: Apache Tomcat Database Server: MySQL [6].

Pada penelitian selanjutnya dari Usep Herdiana dan Agus Ramdhani Nugraha menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language (UML)*. Jadi pihak

perusahaan dapat mengolah data menjadi akurat dan efisien [7].

Pada penelitian ketiga ini dari Mursalim, Jauhari Maulani, dan Gita Ayu Safarina memakai sistem pemecahan masalah *systems development life cycle* dengan *waterfall development model* yang digambarkan dalam bentuk perancangan UML (*Unified Modeling Language*). Pembangunan sistem memakai *PHP* sebagai bahasa pemrograman dan *MySQL* untuk media penyimpanan *database* [8].

Berdasarkan hasil penelitian dari Suci Aulianisa Sakinah, Achmad Sarwandianto, Halimatus Sa'diah. Perancangan terdiri atas dua aktifitas yaitu perancangan *hardware* dan *software* yang mencakup perancangan *database* yaitu berupa *ERD* dan perancangan sistem berupa *DFD* [9].

Penelitian yang terakhir ini dari Marfuah dan Suryo Widiatoro memakai pemecahan masalah penelitian *System Development Life Cycle*. Rancang bangun aplikasi pegawai berbasis dibuat rancangan sesuai dengan kebutuhan *user*, sehingga rancangan yang dibuat mampu untuk peningkatan layanan [10].

Dari beberapa penelitian sebelumnya dapat dilihat bahwa *output* yang dibuat hanya informasi dasar mengenai pemecahan masalah tentang suatu aplikasi rekrutmen agar lebih akurat dan efisien mengenai data karyawan. Berdasarkan hal tersebut pada penelitian ini akan dibuatkan penambahan sistem seleksi karyawan yaitu tes psikotes yang terdapat didalam aplikasi rekrutmen gunanya untuk mempercepat proses seleksi pegawai yang memenuhi kriteria yang dibutuhkan oleh perusahaan.

Menurut Naheed dan Amir (2012) rekrutmen adalah proses mencari orang yang dinilai tepat untuk suatu pekerjaan. Setiap bidang pekerjaan memiliki bidang bisnis yang membutuhkan orang tertentu

yang punya kemampuan dan kualitas [11].

Aplikasi adalah suatu program komputer yang dibangun dengan tujuan melaksanakan suatu pekerjaan dari *user*, contohnya mengelola data, mengatur *Windows* dan sebagainya [12].

Menurut Simarmata (2010), web adalah kumpulan halaman yang saling terhubung dengan komponen perangkat lunak secara semantis [13].

Laravel adalah *framework* PHP bersifat kode terbuka (*open source*) dengan desain MVC (*Model-View-Controller*) yang berguna untuk suatu pembangunan aplikasi [14].

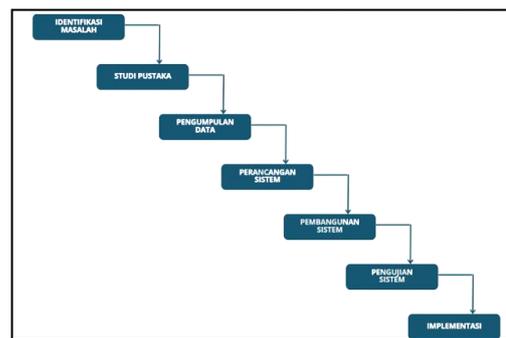
PHP merupakan singkatan dari *Hypertext Preprocessor* yang berguna sebagai bahasa *script server-side* untuk pengembangan web pada dokumen HTML [15].

*Database MySQL* adalah suatu *software database* yang berbentuk *database* relasional yang menggunakan suatu bahasa permintaan yang bernama *SQL* [16].

XAMPP merupakan suatu perangkat lunak bebas, yang banyak mendukung sistem operasi. XAMPP berfungsi sebagai *server* yang berdiri sendiri (*localhost*) [17].

## 2. METODE PENELITIAN

Metode penelitian yang dipakai dalam menerapkan aplikasi penerimaan karyawan baru berbasis *website* menggunakan *framework Laravel*. Proses yang dipakai untuk membangun aplikasi ini yaitu secara *waterfall model*. Pressman (2015,42). Mengatakan bahwa *waterfall* adalah model klasik yang bersifat sistematis yang berurutan dalam membangun suatu *software* [18]. Untuk metodologi penelitian menjadi seperti dibawah ini :



Gambar 1. Metodologi Penelitian

Tahap awal yang dilakukan adalah identifikasi masalah agar menemukan apa saja kebutuhan yang diperlukan untuk membangun perangkat lunak. Studi pustaka menggunakan pengumpulan *literatur review* [19]. Beberapa *litelatur review* yang digunakan adalah bahasa pemrograman *Php*, *MySQL*, *framework Laravel*, dan *XAMPP*. Lalu masuk pada teknik pengumpulan data, disini memakai 2 teknik yaitu:

### a. Observasi

Observasi dilakukan dengan cara terjun langsung pada tempat kerja dengan tujuan mendapatkan data yang akurat berdasarkan permasalahan yang terjadi di tempat kejadian dan hasil yang didapat yaitu belum adanya sistem pendukung untuk penerimaan karyawan baru yang berbasis *online* sebagai media penyebaran lowongan kerja dan pengolahan data, pada perusahaan tersebut saat ini masih menggunakan sistem yang berkerja secara manual sehingga memperlambat pemrosesan data karyawan baru. Metode observasi dalam sejarah perkembangan ilmu pengetahuan bersumber dari dunia empiris, sejak observasi botanis Aristoteles [20].

### b. Wawancara

Wawancara dilakukan dengan cara tanya jawab secara langsung kepada narasumber yang bersangkutan guna mendapatkan informasi yang menjadi kebutuhan untuk pembuatan sistem,

pihak yang diwawancarai adalah orang dari divisi HRD, pertanyaan yang diberikan yaitu, apa yang menjadi kendala dan hambatan dalam proses rekrutmen karyawan baru dan sistem seperti apa yang yang dibutuhkan agar dapat mempermudah perusahaan. Kemudian hasil yang di dapatkan dalam wawancara yaitu perekrutan karyawan pada perusahaan tersebut masih menggunakan cara konvensional sehingga diperlukan pembuatan aplikasi ini dan juga mempermudah pihak HRD dan pihak-pihak terkait dalam pengolahan data perusahaan. Wawancara adalah suatu kaidah

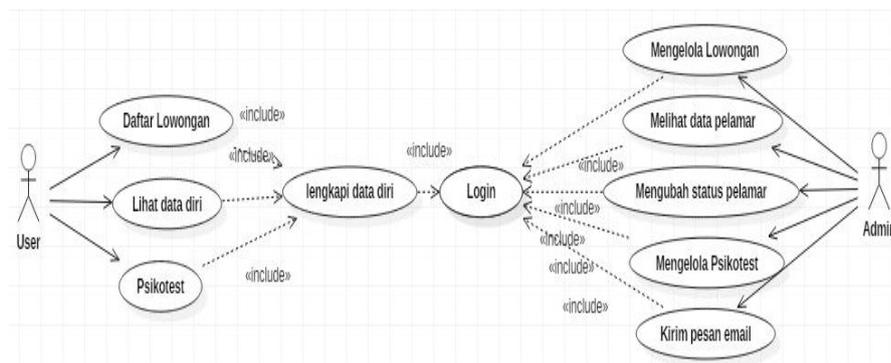
mengumpulkan data yang paling biasa digunakan dalam penelitian sosial [21].

### 3. HASIL DAN PEMBAHASAN

Dalam tahap perancangan sistem dibangun dengan menentukan apa saja yang akan menjadi isi dalam web tersebut seperti halnya tampilan dalam *website*, beberapa halaman yang berisi *menu-menu* dan isi dari setiap *menu-menu* tersebut (konten).

#### 3.1 Use Case

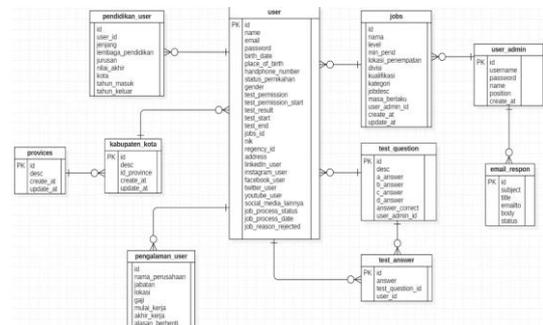
*Use case* merupakan deskripsi fungsi sistem yang dilihat dari sudut pandang para pengguna. *Use case* bekerja menggunakan skenario yang menjelaskan apa yang dilakukan oleh *user* terhadap sistem ataupun sebaliknya [22]. Berikut ini *use case* admin dan *user*:



Gambar 2. Use Case.

#### 3.2 ERD

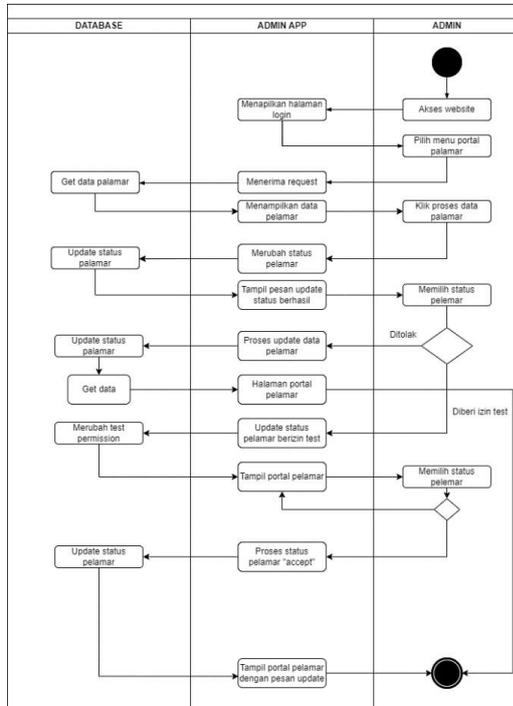
*Entity Relationship Diagram* (ERD) merupakan diagram berbentuk notasi grafis yang ada dalam pembentukan *database* dengan cara menghubungkan antara data satu dengan yang lain yang berfungsi sebagai alat bantu [23]. Berikut ini ERD diagramnya:



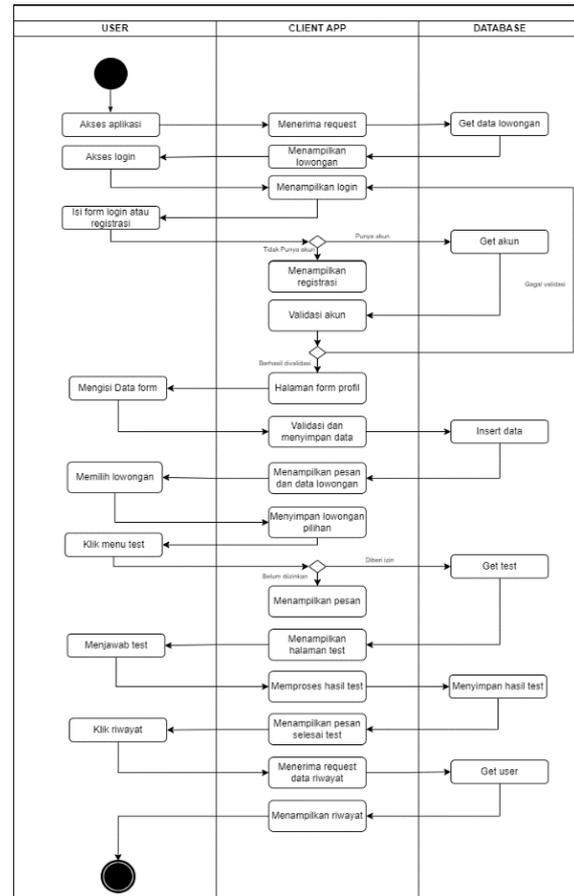
Gambar 3. Entity Relationship Diagram.

### 3.3 Activity Diagram

Activity diagram merupakan gambaran dari berbagai aliran aktivitas untuk suatu sistem yang sedang di rancang [24]. Berikut ini activity diagram yang ada dalam aplikasi:



Gambar 4. Activity Diagram Admin.



Gambar 5. Activity Diagram User.

### 3.4 Rancang Bangun Aplikasi

Rancang bangun aplikasi merupakan suatu rancangan yang disesuaikan dengan kebutuhan sistem, sehingga rancangan yang disusun akan menghasilkan pembangunan suatu aplikasi dalam peningkatan pelayanan diperusahaan.

### Kode Program 1. Dashboard Admin

- **Function Index**

```

1 public function index(){
2     $chartData=array();
3     for ($i=1; $i <= 12; $i++) {
4         $startDate=
5             Carbon::now()->setMonth($i)->start
6             OfMonth();
7         $endDate=
8             Carbon::now()->setMonth($i)->end
9             OfMonth();
10        $chartData[]=$i,
11        count(User::where('created_at', '>',
12        $startDate)->where('created_at', '<=',
13        $endDate)->get());
14    }
15    }

```

```
10 return view('admin.index',[
11 'title'=>'Dashboard',
12 'active'=>'Dashboard',
13 'job'=>count(Job::where('masa_berlaku',
14 '>=', Carbon::today()->get()),
15 'pelamar'=>count(User::all()),
16 'pelamarTerima'=>count(User::where
17 ('job_process_status', '=',
18 'Accepted')->get()),
19 'pelamarTolak'=>count(User::where('
20 job_process_status', '=',
21 'Rejected')->get()),
22 'chartData'=>$chartData
23 ]);
24 ]);
25 }
```

Kode diatas merupakan potongan kode dari *backend* yang mengelola data pada *dashboard*. Fungsi *index* untuk menampilkan halaman dashboard. Baris pertama untuk mendeklarasikan fungsi *index* agar dapat dipanggil pada *route*. Pada baris kedua berfungsi untuk mendeklarasikan variabel *\$chartData* dengan tipe data *array()*. Variabel ini nantinya akan menampung data pelamar dalam setiap harinya pada kurun waktu satu bulan. Pada baris 3 sampai 8 berfungsi untuk *looping* data berdasarkan jumlah hari dalam satu bulan, untuk mendapatkan tanggal awal dan akhir dalam 1 bulan, digunakan *package Carbon* yang dapat digunakan untuk mengelola data tanggal secara cepat dan akurat. Untuk mempermudah mengambil data dalam *database* digunakan fitur *Eloquent* milik *Laravel*, penggunaannya dapat dilihat pada baris 6. Untuk mengambil data *user* pelamar cukup menggunakan nama model yang ditambahkan dengan perintah sesuai dengan kebutuhan. Pada baris 10 sampai 19 berfungsi untuk mengembalikan *view index* yang berada pada *folder admin* dengan beberapa data seperti *title*, total pelamar, total pelamar diterima, total pelamar ditolak dan data jumlah pelamar setiap harinya. Untuk mendapatkan total jumlah data yang ada, digunakan fungsi *count()*.

## Kode Program 2. Input Data Lowongan

### • Function Store

```
1 public function store(Request
2 $request)
3 {
4 $validateData =
5 $request->validate([
6 'nama'=>'required|max:255',
7 'slug' => 'required|unique:jobs',
8 'level_jobs_id'=>'required',
9 'category_jobs_id'=>'required',
10 'lokasi_penempatan'=>'required',
11 'division_id'=>'required',
12 'kualifikasi'=>'required',
13 'jobdesc'=>'required',
14 'masa_berlaku'=>'required'
15 ]);
16 $validateData['user_admin_id']=
17 auth()->user()->id;
18 Job::create($validateData);
19 return
20 redirect('/admin/lowongan')->with
21 ('success', 'New Job has been
22 added !');
23 }
```

Kode diatas merupakan potongan *code* fungsi *store* data pada *Controller Data Lowongan*. Fungsi *store* untuk *insert* data lowongan pekerjaan kedalam *database*. Pada baris 1 berfungsi untuk deklarasi fungsi *store* dengan parameter *\$request*. Parameter *\$request* akan menerima data dari *frontend* yang nantinya akan disimpan pada *database*. Dalam fungsi *store* terdapat validasi data seperti yang dapat dilihat pada baris 3 hingga baris 12. Terdapat beberapa validasi yang dilakukan pada setiap data, seperti untuk mengecek apakah terdapat data kosong (lihat baris 1 sampai 12), menetapkan batas panjang karakter (lihat baris 4), hingga mengecek apakah data tersebut unik dan belum ada dalam *database* (baris 5). Pada baris 14

berfungsi untuk menambahkan data *user admin* id kedalam array, *user id* didapatkan melalui fungsi *auth()* pada laravel yang menyimpan data admin yang *login*. *Syntax insert* data ke dalam *database* dapat dilihat pada baris 15 dengan menggunakan salah satu fungsi dari fitur *eloquent* laravel yaitu fungsi *create* dengan parameter data yang akan di *insert*. Setelah semua prosedur terselesaikan maka fungsi akan mengembalikan pada tampilan melalui *route* seperti pada baris 16 dengan pesan sukses *insert* data.

- **Function Update**

```
1 public function update(Request
  $request, Job $lowongan)
2 {
3     $rules =[
4     'nama'=>'required|max:255',
5     'level_jobs_id'=>'required',
6     'category_jobs_id'=>'required',
7     'lokasi_penempatan'=>'required',
8     'division_id'=>'required',
9     'kualifikasi'=>'required',
10    'jobdesc'=>'required',
11    'masa_berlaku'=>'required'
12 ];
13 if($request->slug !=$lowongan->slug)
14 {
15     $rules['slug']=
16     'required|unique:jobs';
17 }
18 $validateData =
19 $request->validate($rules);
20 $validateData['user_admin_id']=
21 auth()->user()->id;
22 Job::where('id',
23 $lowongan->id)->update($validateData
24 );
25 return
26 redirect('/admin/lowongan')->with('succ
27 ess', 'Job has been updated !');
```

Kode diatas merupakan potongan *code* fungsi *update* data pada *Controller Data Lowongan*. Fungsi *update* untuk *insert* data lowongan pekerjaan kedalam *database*. Pada baris 1 berfungsi untuk deklarasi fungsi *update* dengan parameter *\$request* dan *\$lowongan*. Parameter *\$request* akan menerima data dari *frontend* yang nantinya akan disimpan pada *database* sedangkan *\$lowongan* akan mengembalikan data lowongan pekerjaan sesuai *id* yang dikirimkan oleh *frontend*. Dalam fungsi *update* terdapat validasi data seperti yang dapat dilihat pada baris 4 hingga baris 11. Pada baris 13 sampai 16 terdapat pengkondisian untuk mengecek apakah *slug* yang baru berbeda dengan *slug* data sebelumnya, jika berbeda maka akan dicek kembali untuk memastikan *slug* yang baru itu unik. Pada baris 17 berfungsi untuk menambahkan data *user admin* id kedalam *array*, *user id* didapatkan melalui fungsi *auth()* pada laravel yang menyimpan data admin yang *login*. *Syntax update* data ke dalam *database* dapat dilihat pada baris 18 dengan menggunakan salah satu fungsi dari fitur *eloquent* laravel yaitu fungsi *update* dengan parameter data yang akan di *update*. Setelah semua prosedur terselesaikan maka fungsi akan mengembalikan pada tampilan melalui *route* seperti pada baris 19 dengan pesan sukses *update* data.

- **Function Delete**

```
1 public function destroy(Job
  $lowongan)
2 {
3     Job::destroy($lowongan->id);
4     return
5     redirect('/admin/lowongan')->with('s
6     uccess', 'Lowongan has been
7     Deleted !');
```

Kode diatas merupakan potongan *code* fungsi *delete* data pada *Controller*

Data Lowongan. Fungsi *delete* berfungsi untuk *delete* data lowongan pekerjaan. Pada baris 1 berfungsi untuk deklarasi fungsi *delete* dengan parameter \$lowongan. Parameter \$lowongan akan mengembalikan data lowongan pekerjaan sesuai *id* yang dikirimkan oleh *frontend*. Untuk mempermudah proses hapus data, maka digunakan salah satu fungsi dari *eloquent* Laravel yaitu *destroy* dengan parameter *id* data yang akan dihapus seperti pada baris 3. Baris 4 untuk mengembalikan tampilan melalui *route* seperti pada baris 4 dengan pesan sukses *delete* data.

### Kode Program 3. Management Pelamar

- **Function Proses Pelamar**

```
1 public function prosesPelamar($id){
2   $user = User::find($id);
3   $user->job_process_status = 'Process';
4   $user->save();
5   $service = new MailController();
6
7   $service->sendNotification('process',
   $user->email);
8   return redirect('/admin/management-
   pelamar/portal-
   pelamar')->with('success', 'Data
   Pelamar berhasil diupdate!');
9 }
```

Kode diatas merupakan potongan *code* fungsi proses pelamar data pada *Controller* Manajemen Pelamar. Fungsi *prosesPelamar* untuk *update* status lamaran menjadi di proses agar pelamar mendapatkan pemberitahuan, dimana berkasnya telah diproses. Fungsi *prosesPelamar* memiliki parameter \$id yang didapatkan dari *frontend*. Pada baris 2 variabel \$user akan diberikan *value array* data pelamar yang dicari berdasarkan *id*. Selanjutnya status pelamar akan diganti dengan 'Process' seperti pada baris 3, lalu data akan disimpan. Agar pelamar mendapatkan notifikasi email, diperlukan fungsi kirim email seperti pada baris 5 dan 6, fungsi *sendNotifikasi* yang ada

pada *class MailController* akan menerima 2 parameter, yaitu status dan email *user*. Pada baris 7 fungsi akan mengembalikan *view* dengan pesan sukses di *update*.

### Kode Program 4. Client Controller (Psikotest)

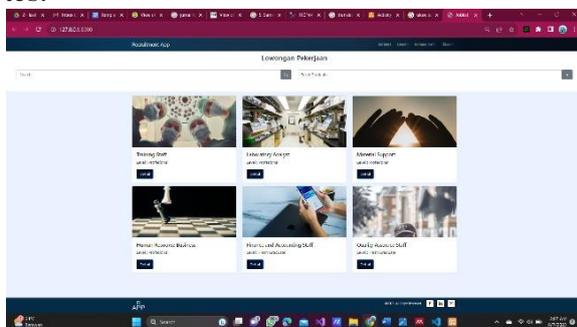
- **Function testSubmit**

```
1 public function testSubmit(Request
   $request){
2   if (Carbon::now()->addHours(7) >
   Carbon::parse(auth()->user()->test_
   start)->addMinutes(61)) {
3     $user =
4     User::find(auth()->user()->id);
5     $user->test_result = '0';
6     $user->test_end =
7     Carbon::now()->addHours(7);
8     $user->save();
9     return
10    redirect('/joblist')->with('warning',
   'Anda gagal submit test tepat
   waktu !');
11  }else{
12    $answerCorrect = 0;
13    $count=1;
14    $percentage=0;
15    foreach (TestQuestion::all() as
   $key ) {
16      if($key['answer_correct']==$reque
   st[$count]){
17        $answerCorrect+=1;
18      }
19      $tmp=['answer'=>$request[$count],
20      'user_id'=>auth()->user()->id,
21      'test_question_id'=> $key['id'],
22      ];
23      TestAnswer::create($tmp);
24      $count++;
25    }
26    $percentage =
27    number_format(($answerCorrect/($
   count-1))*100);
28    $user =
29    User::find(auth()->user()->id);
30    $user->test_result = $percentage;
```

```

26 $user->test_end =
    Carbon::now()->addHours(7);
27 $user->test_permission = false;
28 $user->save();
29 return redirect('/joblist')->with('success',
    'Selamat anda telah menyelesaikan test,
    silahkan
    menunggu admin akan menghubungi
    anda !');
    }
    }
    
```

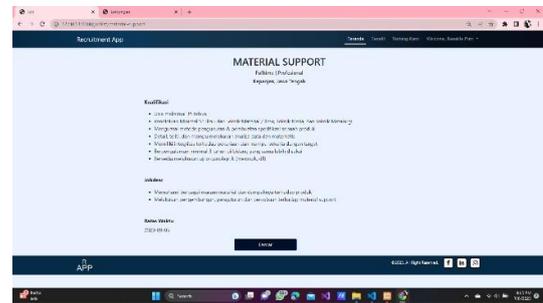
Kode diatas merupakan potongan *code* fungsi *testSubmit* pada *Controller ClientController*. Fungsi *testSubmit* untuk *insert* hasil tes yang telah dikerjakan oleh pelamar. Fungsi ini memiliki parameter *\$request* yang akan menampung data yang dikirimkan dari *frontend*. Pada baris 2 terdapat pengkondisian untuk mengecek apakah *user* telah mengirimkan *test* sesuai dengan waktu yang telah ditentukan, pengecekan ini menggunakan *package Carbon* untuk mempermudah. Jika pelamar mengirim tes tidak tepat waktu, maka hasil tes akan diberi nilai nol dan pelamar akan mendapatkan *alert*. Namun jika *submit* tes tepat waktu, maka setiap jawaban akan di bandingkan dengan jawaban yang benar seperti pada baris 10 sampai 23. Persentase hasil tes yang benar juga akan langsung dihitung seperti pada baris 25 dan akan langsung disimpan pada *database*. Setelah data disimpan, fungsi akan mengembalikan *view* beserta pesan berhasil menyelesaikan tes.



Gambar 6. Halaman Beranda.

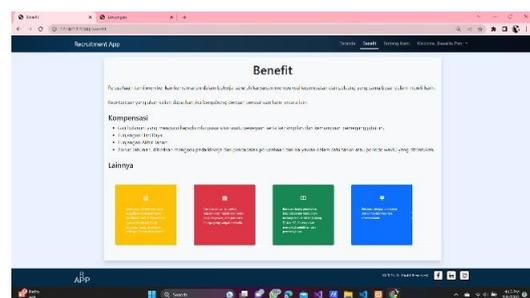
Tampilan pertama pada web ini adalah halaman beranda, disini para calon pelamar

dapat melihat lowongan pekerjaan apa saja yang dibuka oleh perusahaan.



Gambar 7. Halaman Detail Lowongan.

Selanjutnya halaman *detail* lowongan, dimana para calon pelamar bisa langsung melihat *jobdesc* dan kualifikasi apa yang ada dalam persyaratan lowongan yang dibutuhkan. Tetapi sebelum menekan *button* lamar pada bagian bawah, para pelamar harus membuat akun terlebih dahulu karena jika belum mempunyai akun dan langsung menekan *button* lamar, para pelamar akan diarahkan untuk membuat akun terlebih dahulu dan mengisi data-data yang diperlukan, baru setelah itu bisa mengakses semua fitur dalam *website* tersebut.



Gambar 8. Halaman Benefit.

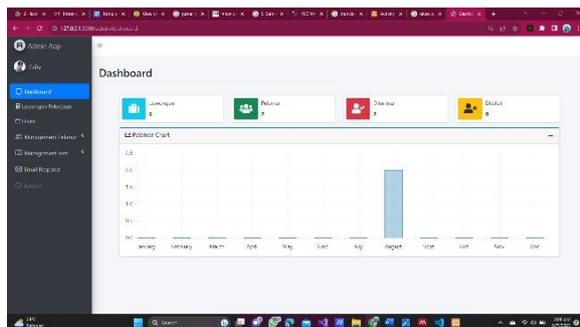
Pada halaman *benefit* akan di jelaskan apa saja yang akan didapatkan ketika para karyawan bekerja dalam perusahaan tersebut, admin perusahaan juga dibantu dengan fitur ini karena

memperudah untuk mengenalkan hal-hal yang ada pada perusahaan.



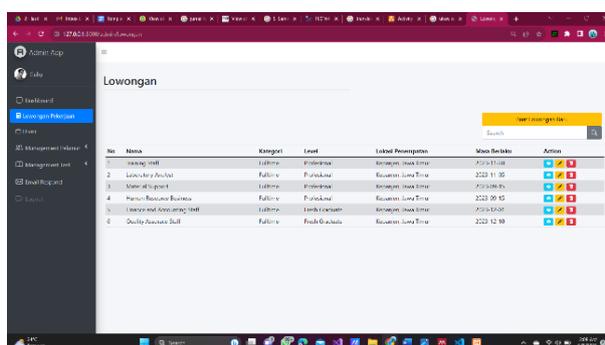
Gambar 9. Halaman Profile Company.

Pada halaman *profile company*, akan di jelaskan sekilas tentang perusahaan tersebut, ini membantu juga bagi calon pelamar untuk mengetahui informasi tentang perusahaan.



Gambar 10. Halaman Dashboard Admin.

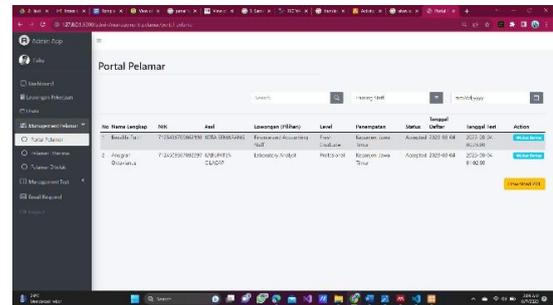
Selanjutnya halaman *dashboard* admin, dimana admin dapat mengontrol para pelamar yang mendaftar dan membantu admin dalam pengecekan data para pelamar.



Gambar 11. Halaman Admin Input Lowongan.

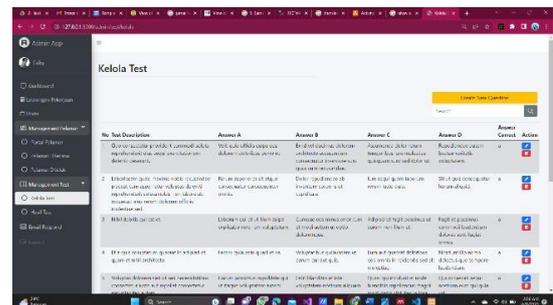
Pada halaman inilah admin membuat lowongan-lowongan pekerjaan yang dibutuhkan oleh perusahaan, fitur ini akan

memperudah admin dalam penginputan lowongan kerja.



Gambar 12. Halaman Portal Pelamar.

Pada halaman ini admin dapat melihat semua berkas dari pelamar di setiap lowongan yang telah masuk data-datanya dan dihalaman ini terdapat fitur *report* yang bisa *download* data-data pelamar menjadi laporan PDF jadi fitur ini juga membantu admin membuat laporan kepada atasan perusahaan.



Gambar 13. Halaman Soal Psikotes.

Halaman ini berfungsi untuk membantu admin dalam mengelola setiap soal dan jawaban psikotes, ini merupakan keunggulan dalam aplikasi ini, karena sudah dibangun sistem tes.

Pengujian perangkat lunak yang digunakan untuk menguji kode program yaitu metode *BlackboxTesting* yang menjadi salah satu cara untuk dipakai dalam proses menguji data yang dihitung melalui banyaknya *field* data entri, ada aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang harus memenuhi syarat untuk mengetahui apakah fungsi, masukan dan keluaran dari sistem berjalan sesuai dengan

spesifikasi yang dibutuhkan. Pengujian ini dilakukan untuk memvalidasi fungsi apakah dapat berfungsi dengan baik. Hasil pengujian dapat dilihat pada **Tabel 1. Blackbox Testing**

Pengujian	Yang diujikan	Hasil Pengujian	Kesimpulan
<i>Form Login</i>	Memasukkan <i>email</i> dan <i>password</i> yang sesuai, memasukkan <i>email</i> dan <i>password</i> yang tidak sesuai.	Berhasil masuk ke halaman <i>admin/user</i> jika sudah sesuai, apabila <i>email</i> dan <i>password</i> tidak sesuai maka akan kembali pada halaman <i>login</i> .	<i>Valid</i>
Tombol <i>Logout</i>	Keluar dari halaman <i>admin/user</i> .	Jika berhasil keluar maka sistem akan masuk ke halaman awal.	<i>Valid</i>
<i>Form Lowongan Perkerjaan</i>	<i>User</i> memilih lowongan pekerjaan .	Berhasil masuk jika telah memiliki akun dan akan masuk inputan lamaran kepada admin, jika belum memiliki akun akan diarahkan untuk membuat akun	<i>Valid</i>

<i>Form Biodata Pelamar</i>	<i>User</i> memasukkan identitas diri dan riwayat pendidikan dari pelamar.	Jika <i>user</i> sudah memasukkan data diri maka akan otomatis masuk pada <i>dashboard admin</i> . Jika admin telah menyetujui berkas pelamar, maka <i>form</i> psikotes akan terbuka.	<i>Valid</i>
<i>Form Psikotes</i>	<i>User</i> mengikuti tes psikotes yang diberikan oleh perusahaan.	Setelah <i>user</i> berhasil mengisi tes psikotes, selanjutnya akan ada pemberitahuan dari admin untuk lanjut ke tahap selanjutnya.	<i>Valid</i>
<i>Form Jadwal Wawancara</i>	<i>Admin</i> mengisi jadwal dan <i>link</i> wawancara untuk <i>user/pelamar</i> yang akan diberitahukan melalui <i>email</i> .	Pesan pemberitahuan berhasil terkirim kepada <i>email user</i> .	<i>Valid</i>

<i>Form Jadwal Masuk Kantor</i>	<i>Admin mengisi jadwal masuk kantor untuk user/pelamar yang akan diberitahukan melalui email.</i>	<i>Pesan pemberitahuan berhasil terkirim kepada email user.</i>	<i>Valid</i>
---------------------------------	--	---	--------------

#### 4. KESIMPULAN

Berdasarkan hasil penelitian yang telah dilakukan pada perusahaan XYZ maka hasil yang didapatkan adalah terbangunlah sebuah Aplikasi Penerimaan Karyawan Baru Berbasis Web menggunakan *Framework Laravel* dengan dibangunnya aplikasi ini diharapkan membantu para pelamar dalam mendapatkan informasi tentang lowongan pekerjaan dan memasukkan lamaran pekerjaan pada PT. XYZ secara *online*. Aplikasi ini juga dapat membantu *admin* perusahaan dalam menginput informasi perusahaan, menyebarkan informasi lowongan pekerjaan, mendapatkan data para pelamar, dan membantu *admin* dalam tes awal karena sudah dibangun juga sistem psikotes maka dari itu ini akan sangat membantu *admin* untuk memproses setiap tahap perekrutan dari awal sampai akhir perekrutan karyawan baru.

#### DAFTAR PUSTAKA

- [1] B. Y. Faqih, “Perancangan Perancangan Aplikasi Cari Kerja Berbasis Website Menggunakan Laravel,” *JATISI (Jurnal Tek. Inform. dan Sist. Informasi)*, vol. 9, no. 1, pp. 566–581, 2022, doi: 10.35957/jatisi.v9i1.1537.
- [2] N. K. Wahab and F. Eka, “Perancangan Sistem Perekrutan Karyawan Pada PT Graha Multi Sempurna Berbasis Web,” *SATERO (Jurnal Sains, Teknol. dan Robot. )*, vol. 6, no. 2, pp. 1–8, 2022.
- [3] Y. Yusran, L. S. Lesmana, F. Putra, and E. Yandani, “Rancang Bangun Sistem Informasi Lowongan Kerja Berbasis WEB,” *J. Ilm. Teknol. Inf. Asia*, vol. 14, no. 2, p. 119, 2021, doi: 10.32815/jitika.v14i2.454.
- [4] H. Purwantih, Z. Fuji, W. Amelia, R. Dwi, and H. M. Bilqis, “Rancang Bangun Sistem Seleksi Rekrutmen Karyawan Dan Guru Berbasis Website Pada Sekolah Citra Bangsa Tangerang,” *ADI Bisnis Digit. Interdisiplin J.*, vol. 1, no. 2 Desember, pp. 60–70, 2020, doi: 10.34306/abdi.v1i2.219.
- [5] N. Hikmah, R. Bagus Pratama, and Suryanto, “Rancang Bangun Sistem Informasi Akademik Berbasis Web Menggunakan Metode Extreme Programming,” *Semin. Nas. Inov. dan Tren*, vol. 5, no. ISBN: 978-602-61268-5-6, pp. 134–139, 2018.
- [6] S. Lena and L. Sangita, “Aplikasi Rekrutmen Karyawan Berbasis Web Menggunakan Framework Laravel,” *J. Teknol. Inf.*, 2018.
- [7] U. Herdiana and A. R. Nugraha, “Rancang Bangun Aplikasi Penerimaan Pegawai Di Bank Btn Kota Tasikmalaya,” *Jumantaka*, vol. 02, no. 01, pp. 191–200, 2018.
- [8] . M., J. Maulani, and G. A. Safarina, “Rancang Bangun Sistem Informasi Rekrutmen

- Seleksi Dan Kontak Kerja Karyawan Berbasis Web,” *Technol. J. Ilm.*, vol. 13, no. 2, p. 145, 2022, doi: 10.31602/tji.v13i2.6701.
- [9] S. A. Sakinah, A. Sarwandianto, and H. Sa’diah, “Perancangan Aplikasi Sistem Penerimaan Karyawan Baru pada PT Central Mega Kencana,” *J. Ris. dan Apl. Mhs. Inform.*, vol. 1, no. 04, pp. 452–459, 2020, doi: 10.30998/jrami.v1i04.451.
- [10] M. Marfuah and S. Widiatoro, “Rancang Bangun Aplikasi Perekrutan Pegawai Berbasis Web (Studi Kasus Universitas Universal),” *J. SAINTEKOM*, vol. 8, no. 1, p. 78, 2018, doi: 10.33020/saintekom.v8i1.50.
- [11] N. Aisyah and A. Giovanni, “Pengaruh Proses Rekrutmen (Porek) dan Seleksi terhadap Kinerja Karyawan PT. BANK MAYA PADA INTERNASIONAL, Tbk,” *Jesya (Jurnal Ekon. Ekon. Syariah)*, vol. 1, no. 2, pp. 8–18, 2018, doi: 10.36778/jesya.v1i2.17.
- [12] S. Sukatmi, “Aplikasi Absensi Siswa Berbasis Web Dengan Dukungan Sms Gateway Pada Smk Kridawisata Bandar Lampung,” *J. Inf. dan Komput.*, vol. 6, no. 1, pp. 20–29, 2018, doi: 10.35959/jik.v6i1.58.
- [13] Tulisan, “Bab Ii. Landasan Teori,” *J. Chem. Inf. Model.*, vol. 53, no. 9, p. 14, 2019.
- [14] Ismai, “Pemanfaatan Framework laravel Untuk Pengembangan Sistem Informasi Toko Online Di Toko New Trend Baturetno,” *J. Scr.*, vol. 7, no. 2, pp. 232–238, 2019.
- [15] A. Sahi, “Aplikasi Test Potensi Akademik Seleksi Saringan Masuk Lp3I Berbasis Web Online Menggunakan Framework Codeigniter,” *Tematik*, vol. 7, no. 1, pp. 120–129, 2020, doi: 10.38204/tematik.v7i1.386.
- [16] C. Shah, “MySQL,” *A Hands-On Introd. to Data Sci.*, pp. 187–206, 2020, doi: 10.1017/9781108560412.008.
- [17] H. Riyadli, A. Arliyana, and F. E. Saputra, “Rancang Bangun Sistem Informasi Keuangan Berbasis WEB,” *J. Sains Komput. dan Teknol. Inf.*, vol. 3, no. 1, pp. 98–103, 2020, doi: 10.33084/jsakti.v3i1.1770.
- [18] M. Harun, “Rancang Bangun Sistem Informasi Rekrutmen pada PT. Asia Makmur Sejahtera dengan Metode Fishbone,” *J. Akrab Juara*, vol. 4, no. 3, pp. 193–204, 2019.
- [19] W. Andriani, “Penggunaan Metode Sistematis Literatur Review dalam Penelitian Ilmu Sosiologi,” *J. PTK dan Pendidik.*, vol. 7, no. 2, 2022, doi: 10.18592/ptk.v7i2.5632.
- [20] H. Hasanah, “TEKNIK-TEKNIK OBSERVASI (Sebuah Alternatif Metode Pengumpulan Data Kualitatif Ilmu-ilmu Sosial),” *At-Taqaddum*, vol. 8, no. 1, p. 21, 2017, doi: 10.21580/at.v8i1.1163.
- [21] R. Mita, “Wawancara Sebuah Interaksi Komunikasi Dalam Penelitian Kualitatif,” *Jurnal Ilmu Budaya*, vol. 11, no. 2, pp. 71–79, 2015.
- [22] L. Setiyani, “Desain Sistem : Use Case Diagram Pendahuluan,” *Pros. Semin. Nas. Inov. Adopsi*

- 
- TeknoL*. 2021, no. September, pp. 246–260, 2021, [Online]. Available: <https://journal.uui.ac.id/AUTOMATA/article/view/19517>
- [23] K. 'Afifah, Z. F. Azzahra, and A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database Sebuah Literature Review," *Intech*, vol. 3, no. 2, pp. 18–22, 2022, doi: 10.54895/intech.v3i2.1682.
- [24] Ismai, "Perancangan Sistem Aplikasi Pemesanan Makanan dan Minuman Pada Cafeteria NO Caffe di Tanjung Balai Karimun Menggunakan Bahasa Pemrograman PHP dan MySQL," *J. Tikar*, vol. 1, no. 2, pp. 192–206, 2020, [Online]. Available: [https://ejurnal.universitaskarimun.ac.id/index.php/teknik\\_informatika/article/download/153/121](https://ejurnal.universitaskarimun.ac.id/index.php/teknik_informatika/article/download/153/121)