

BAB III

PEMBAHASAN

3.1 Struktur Flowchart

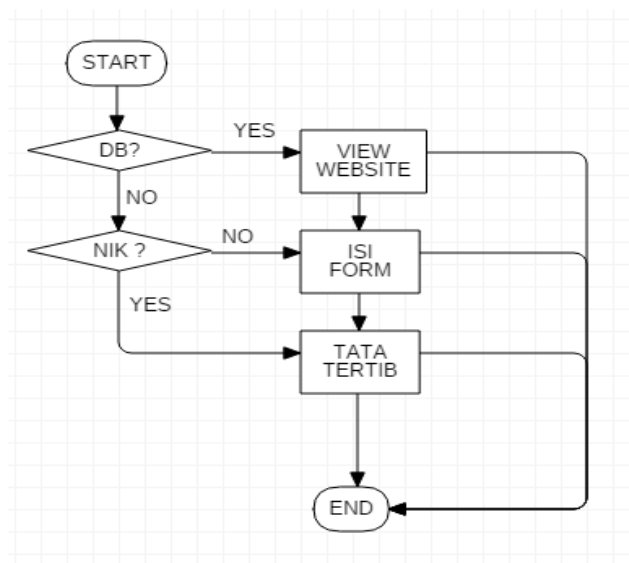
Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur sebuah *Website* yang dibuat

Alur dari Sistem Informasi Kehilangan Barang Berbasis Website diuraikan kedalam beberapa Flowchart

- Flowchart Menu User
- Flowchart Menu Admin
- Flowchart Admin Laporan Menemukan Barang
- Flowchart Admin Laporan Kehilangan Barang
- Flowchart Admin Claim Menemukan / Laporan kehilangan Barang

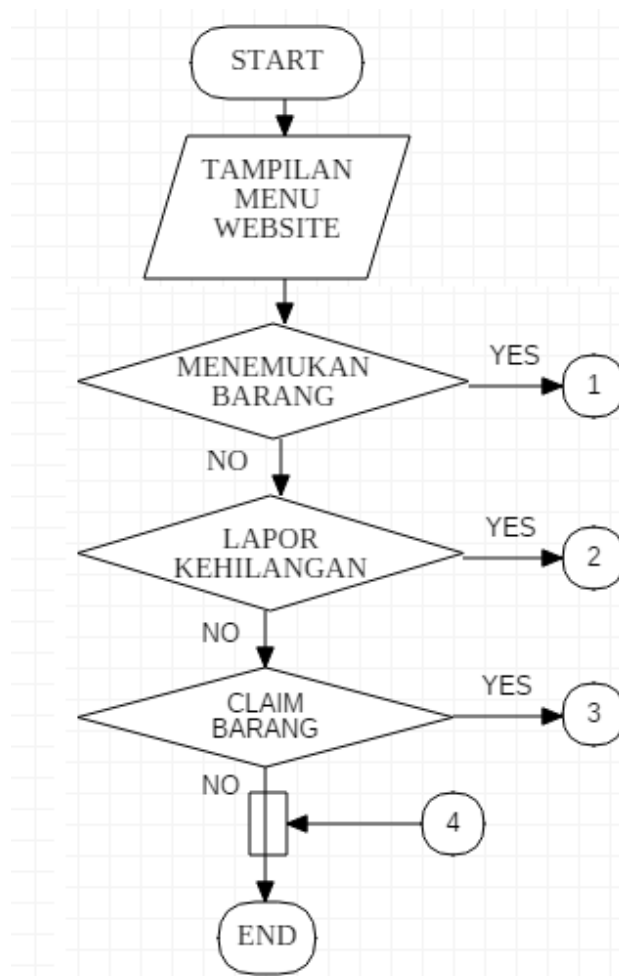
3.1.1. Flowchart Menu User

Flowchart User pada gambar 3.1 menjelaskan bahwa user yang mengakses website dapat melihat list barang , melakukan daftar NIK , dan melihat tata tertib yang berlaku



Gambar 3.1 Struktur Flowchart User

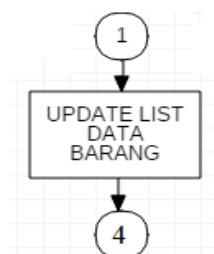
3.1.2. Flowchart Admin



Gambar 3.2 Struktur Flowchart Menu Admin

3.1.3. Flowchart Admin Laporan Menemukan Barang

Flowchart Admin pada gambar 3.2 menjelaskan bahwa setiap barang yang di temukan akan di input oleh admin

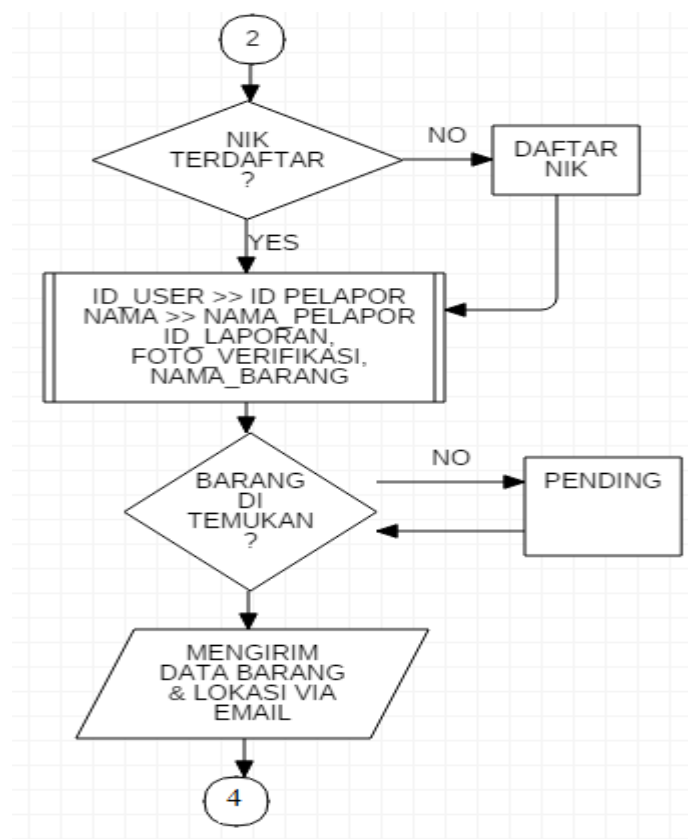


Gambar 3.3 Struktur Flowchart Menemukan

3.1.4. Flowchart Admin Laporan Kehilangan Barang

Flowchart Admin pada gambar 3.3 menjelaskan bahwa setiap orang yang melaporkan kehilangan barang yang NIK nya belum terdaftar, akan di daftarkan terlebih dahulu, sebelum data laporan kehilangan barang di masukan ke dalam *Website* , pelapor di minta untuk meng-*upload* foto verifikasi untuk di cocokan pada barang yang hilang,

Contoh : kehilangan SIM , maka Pelapor meng-*upload* foto barang yang mempunyai hubungan dengan SIM , seperti KTP dsb. bila barang yang dilaporkan ditemukan oleh *user* lain , system *Website* akan mengirimkan email berisi nama barang yang di laporkan dan lokasi pengambilan barang



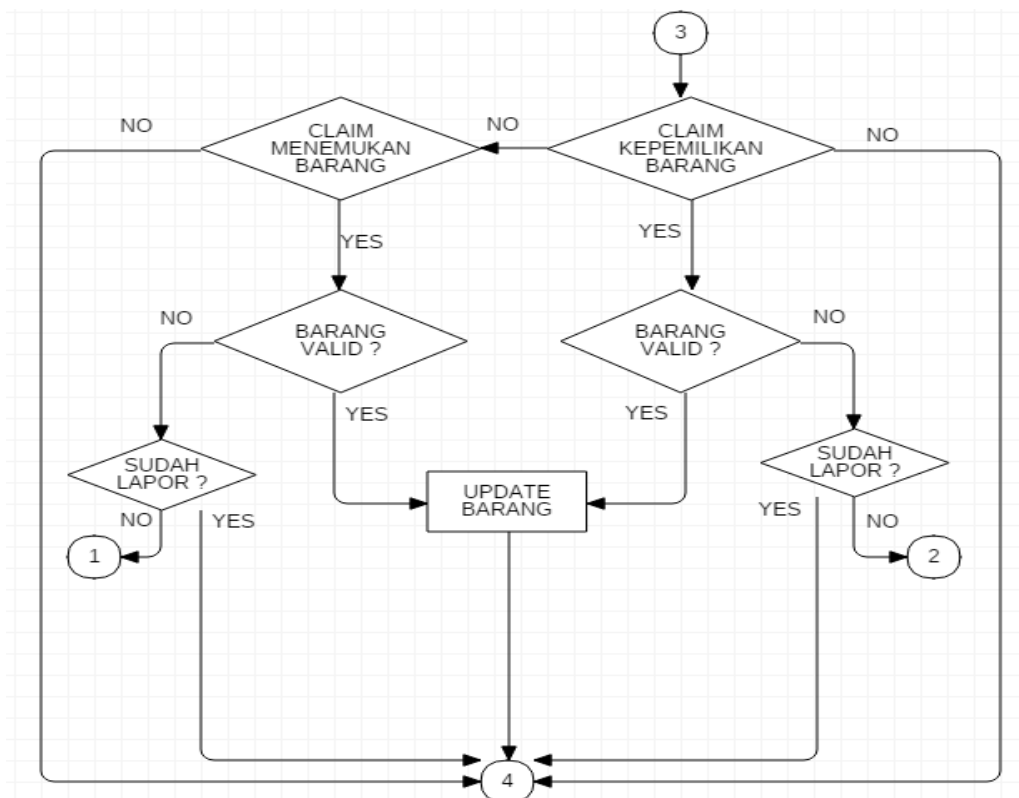
Gambar 3.4 Struktur Flowchart Kehilangan

3.1.5. Flowchart Admin Claim Menemukan / Laporan keilangan Barang

Flowcart Admin pada gambar 3.4 menjelaskan bahwa setiap user yang ingin meng-*claim* suatu barang yang NIK nya belum terdaftar, akan di daftarkan terlebih dahulu,

user yang melakukan *claim* kepemilikan barang akan langsung di berikan barang yang terkait bila foto barang yang ada di website memiliki hubungan dengan foto verifikasi yang di bawa, jika tidak maka user bisa membuat laporan kehilangan barang.

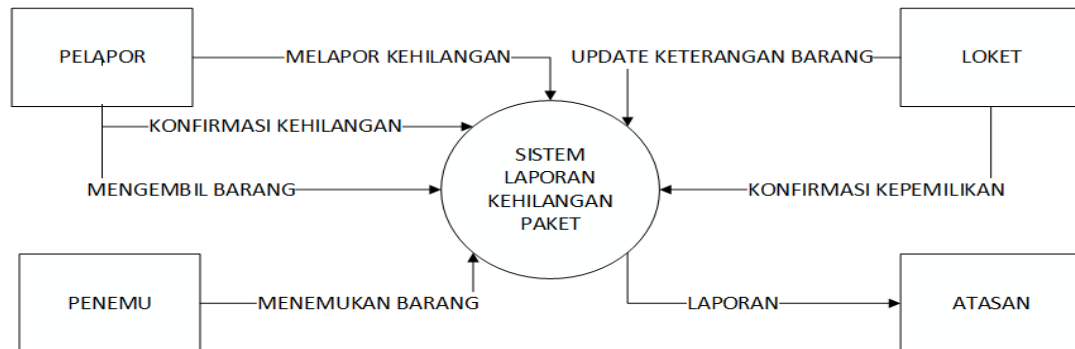
user yang melakukan *claim* menemukan barang akan langsung di-*input* datanya apa bila foto verifikasi dari seorang pelapor memiliki hubungan dengan barang yang di temukan, jika tidak maka user bisa membuat laporan menemukan barang.



Gambar 3.5 Struktur Flowchart Claim kepemilikan & menemukan barang

3.2 DFD Level Context

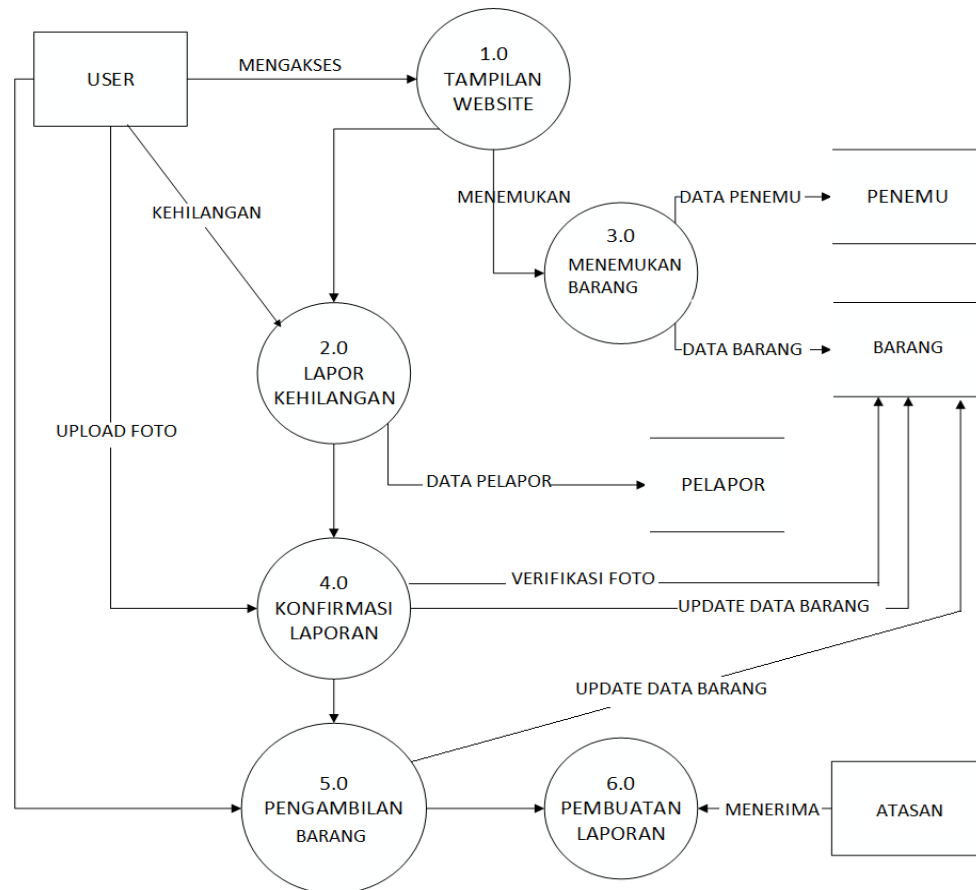
Dari gambar di bawah di jelaskan bahwa pelapor harus melaporkan kehilangan harus melakukan konfirmasi kepemilikan terlebih dahulu agar dapat mengambil barang atau melaporkan barang kehilangan bila konfirmasi kepemilikan nya tidak cocok.



Gambar 3.6 DFD Level Context

3.3 DFD

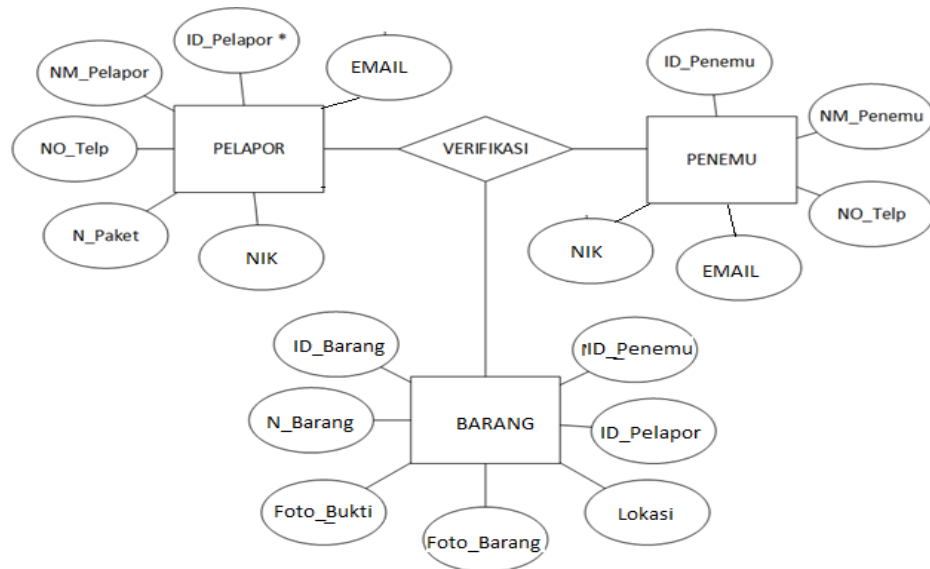
1. User mengakses website , user dapat melihat lihat barang-barang yang di temukan atau barang-barang yang hilang.
2. yang claim / melaporkan kehilangan barang akan digolongkan menjadi user PELAPOR pelapor harus membawa bukti kepemilikan berupa foto.
3. User yang claim menemukan / menemukan barang akan di golongan menjadi user PENEMU.
4. Proses konfirmasi dimana petugas melakukan verifikasi antara foto barang dengan foto yang di upload oleh pelapor
5. Proses pengambilan barang
6. Proses pembuatan laporan



Gambar 3.7 DFD

3.4 Entity Relationship Diagram

ERD merupakan sebuah penggambaran atau simulasi dalam keterkaitan antar entitas. Penentuan keterikatan entitas penuh atau sebagian ditelusuri. Aplikasi yang penulis buat memiliki penggambaran ERD sebagai berikut:



Gambar 3.8 Entity Relationship Diagram

3.5 Perancangan Database

Berdasarkan aturan *mapping* yang mana ERD mempunyai binary relationship M:1, maka N mempunyai Participation Constraint partial dan selanjutnya membuat relasi baru dimana *primary key*nya merupakan gabungan dari *primary key* kedua tipe entitas yang berelasi.

Table 3.1 Tabel User

No	Nama Field	Tipe Data	Panjang Data		
1	Id	Int (AI)	10	Not null	Primary key
2	nik	Varchar	15	Not null	
3	telp	Varchar	10	Not null	
4	name	Varchar	20	Not null	
5	email	Varchar	20	Not null	
6	password	Varchar	15	Not null	
7	role	Enum (petugas/user)		Not null	
8	lokasi	Varchar	20	Not null	

Tabel 3.2 Tabel Barang

x	Nama Field	Tipe Data	Panjang Data		
1	Id	Int (AI)	10	Not null	Primary key
2	pelapor_ID	Varchar	10	Null	Foreign key
3	penemu_ID	Varchar	10	Null	Foreign key
4	nama	Varchar	15	Not null	
5	foto_barang	Images		Null	
6	foto_bukti	Images		Null	
7	lokasi	Varchar	20	Not null	
8	ket	Varchar	20	Not null	

Pada penggambaran ERD serta proses transformasinya maka pada DBMS MySQL akan dibentuk beberapa tabel. Pembentukan field dimulai terlebih dahulu dengan membuat sebuah database untuk menampung tabel – tabel yang akan digunakan sebagai tempat menampung data.

Sesuai dengan hasil pemetaan yang dihasilkan dari ERD maka tabel – tabel yang dihasilkan adalah user dan barang. Tabel user mempunyai tugas sebagai penampung data pengguna , data user di bagi menjadi 2 role yaitu :

- 1 . petugas memiliki hak akses untuk melakukan verifikasi data barang , dan dapat membantu user melakukan laporan menemukan barang atau laporan kehilangan barang , laporan barang yang di buat oleh setiap petugas akan mengambil data lokasi user petugas ke data lokasi barang
2. user akan di golongan menjadi 2 user ya itu penemu bagi user yang menemukn barang , dan pelapor bagi user yang melaporkan kehilangan , id user akan di jadikan penemu_id apa bila menemukan barang dan akan di jadikan pelapor_id bila melaporkan kehilangan

Pada database barang terdapat Foreign key dari table user berupa pelapor_id sebagai tanda barang di laporkan hilang dan penemu_id sebagai tanda barang di temukan . lokasi barang di dapat dari lokasi petugas yang melakukan input laporan kehilangan atau laporan menemukan barang

3.6 Perancangan Website dengan Framework Laravel

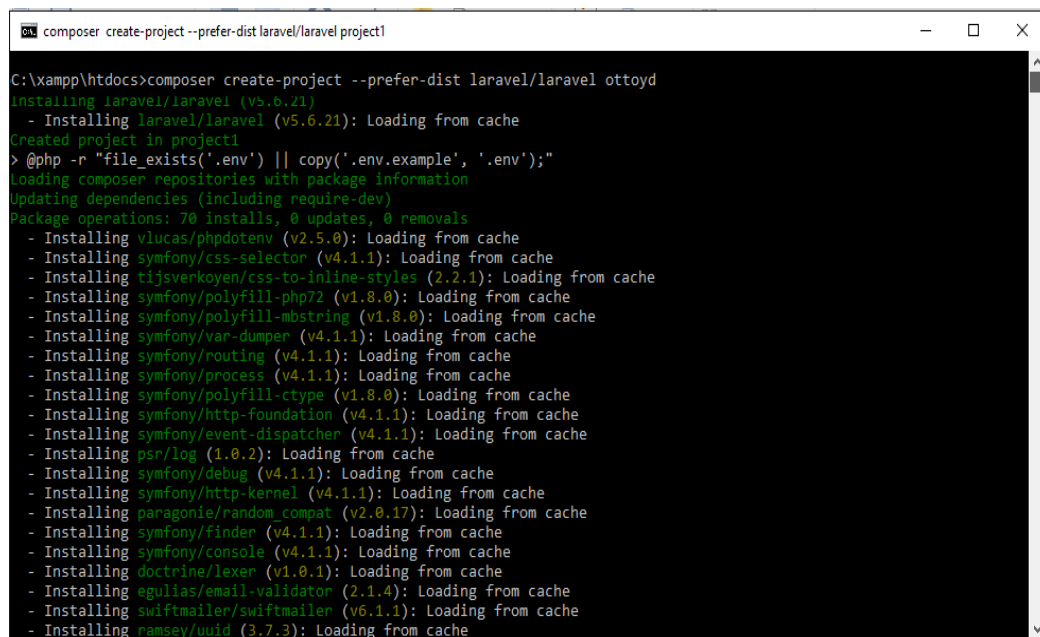
Sebuah aplikasi atau karya apapun dapat dipastikan mempunyai tampilan yang mempunyai tujuan tertentu untuk user. Di dalam website ini pun mempunyai peranan untuk pengguna dapat dengan mudah berinteraksi. Tampilan dalam karya yang penulis buat menggunakan tag dasar HTML (Hypertext Markup Language) dan PHP menggunakan Framework Laravel

Buka CMD pada folder yang di inginkan

Lalu masukan code :

`composer create-project --prefer-dist laravel/laravel ottoyed << nama project`

composer akan otomatis mendownload file yang di butuhkan

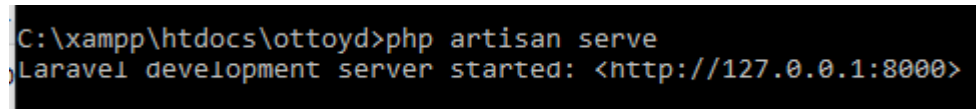


```

C:\xampp\htdocs>composer create-project --prefer-dist laravel/laravel ottoyed
Installing laravel/laravel (v5.6.21)
- Installing laravel/laravel (v5.6.21): Loading from cache
Created project in project1
> @php -r "file_exists('.env') || copy('.env.example', '.env');"
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 70 installs, 0 updates, 0 removals
- Installing vlucas/phpdotenv (v2.5.0): Loading from cache
- Installing symfony/css-selector (v4.1.1): Loading from cache
- Installing tijsverkoyen/css-to-inline-styles (2.2.1): Loading from cache
- Installing symfony/polyfill-php72 (v1.8.0): Loading from cache
- Installing symfony/polyfill-mbstring (v1.8.0): Loading from cache
- Installing symfony/var-dumper (v4.1.1): Loading from cache
- Installing symfony/routing (v4.1.1): Loading from cache
- Installing symfony/process (v4.1.1): Loading from cache
- Installing symfony/polyfill-ctype (v1.8.0): Loading from cache
- Installing symfony/http-foundation (v4.1.1): Loading from cache
- Installing symfony/event-dispatcher (v4.1.1): Loading from cache
- Installing psr/log (1.0.2): Loading from cache
- Installing symfony/debug (v4.1.1): Loading from cache
- Installing symfony/http-kernel (v4.1.1): Loading from cache
- Installing paragonie/random_compat (v2.0.17): Loading from cache
- Installing symfony/finder (v4.1.1): Loading from cache
- Installing symfony/console (v4.1.1): Loading from cache
- Installing doctrine/lexer (v1.0.1): Loading from cache
- Installing egulias/email-validator (2.1.4): Loading from cache
- Installing swiftmailer/swiftmailer (v6.1.1): Loading from cache
- Installing ramsey/uuid (3.7.3): Loading from cache
  
```

Gambar 3.9 Membuat project Laravel

Perintah di bawah berfungsi untuk menjalankan server untuk website yang di buat , setelah itu kita bisa mencoba nya dengan memasukan localhost:8000 pada web browser



```

C:\xampp\htdocs\ottoyde>php artisan serve
Laravel development server started: <http://127.0.0.1:8000>
  
```

Gambar 3.10 PHP artisan serve

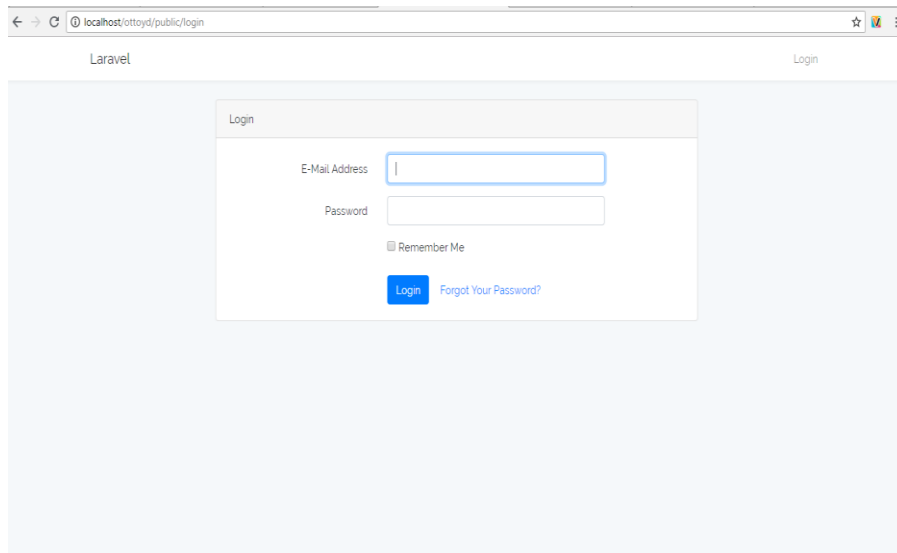
```
C:\xampp\htdocs\project>php artisan make:auth  
Authentication scaffolding generated successfully.
```

Gambar 3.11 PHP Artisan Auth

Perintah pada gambar 3.11 berfungsi untuk membuat user Authentication pada folder

resource > view > auth seperti :

Interface login.blade (Admin)



The screenshot displays a web browser window with the address bar showing 'localhost/ottoydi/public/login'. The page layout includes the word 'Laravel' on the top left and 'Login' on the top right. The main content area features a login form with the following elements: a title 'Login', an 'E-Mail Address' input field, a 'Password' input field, a 'Remember Me' checkbox, a blue 'Login' button, and a 'Forgot Your Password?' link.

Gambar 3.12 Interface Login

Pada gambar 3.12 menampilkan menu untuk Admin melakukan Login. Pada tampilan tersebut Admin memasukan email dan password yang sudah ter-registrasi agar dapat mengakses halaman berikutnya

The screenshot shows a web browser window with the address bar displaying 'localhost/ottoytd/public/register'. The page has a header with 'Laravel' on the left and 'Ugai Register' on the right. The main content is a registration form titled 'Register' with the following fields: NIK, Name, E-Mail Address, Telepon, Password, Confirm Password, and Lokasi. A blue 'Register' button is at the bottom of the form.

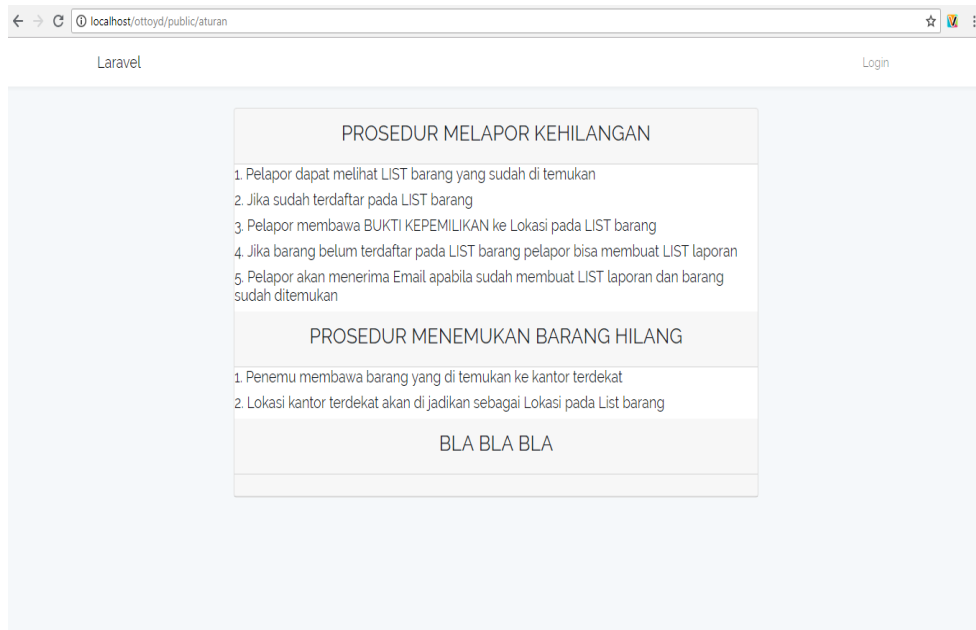
Gambar 3.13 Interface Registerasi

Pada gambar 3.13 menampilkan menu untuk Admin melakukan registrasi yang terdiri dari NIK, Name , Email , Telpon , Password, dan Lokasi , yang akan digunakan untuk Authentifikasi Login dan akan di simpan di table User dengan Role Admin



Gambar 3.14 Tampilan Home

Pada gambar 3.14 didalam folder view tersebut penulis merancang beberapa tampilan yang akan digunakan dengan aplikasi ini. Berikut ini adalah tampilan – tampilan yang dibuat :



Gambar 3.15 Interface aturan



Gambar 3.16 Home (ADMIN)

Home Login							
Id Laporan	Pelapor	Barang	Tanggal Lapor	Tanggal Claim	Lokasi	Keterangan	Status
2	AMIR	KARTU KTP	2018-08-05 15:30:08	2018-08-05 15:30:51	POLDA JABAR	KTP DENGAN NAMA BUDI	Success
3	Ugai	ktp	2018-08-06 05:15:51	2018-08-06 05:58:30	POLDA JABAR	ditemukan ktp bernama budi	Success
11		SIM MOTOR	2018-08-09 00:53:29		POLDA JABAR	Sim bernama Budi	
12	AMIR		2018-08-09 00:54:29			KTP BUDI	

Gambar 3.17 Interface item (USER)

Home				Ugai ▾ Register			
Id Laporan	Pelapor	Barang	Tanggal Lapor	Tanggal Claim	Lokasi	Keterangan	Status
2	AMIR	KARTU KTP	2018-08-05 15:30:08	2018-08-05 15:30:51	POLDA JABAR	KTP DENGAN NAMA BUDI	Success
3	Ugai	ktp	2018-08-06 05:15:51	2018-08-06 05:58:30	POLDA JABAR	ditemukan ktp bernama budi	Success
11		SIM MOTOR	2018-08-09 00:53:29		POLDA JABAR	Sim bernama Budi	claim
12	AMIR		2018-08-09 00:54:29			KTP BUDI	claim

Gambar 3.18 Interface barang (ADMIN)

Interface barang.blade (ADMIN) Claim Barang

The screenshot shows a web browser window with the URL `localhost/ottoy/public/barang/34/edit`. The page title is 'Laravel' and the user is 'Ugai'. The main content is a form titled 'Laporan' with the following fields:

- NIK**: A text input field.
- Foto Bukti/Barang**: A file upload area with a 'Choose File' button and the text 'No file chosen'.
- Lokasi**: A text input field containing 'POLDA JABAR'.
- Submit**: A blue button at the bottom of the form.

Gambar 3.19 Interface item Claim barang (ADMIN)

3.7 Migrasi Database pada Laravel

Penggunaan database dalam aplikasi ini merupakan sebuah keharusan. Karena setiap data yang masuk harus dapat disimpan secara baik untuk dapat ditampilkan pada halaman lihat seluruh proyek. Maka dari itu, sebelum melakukan operasi atau manipulasi database, penulis akan menjelaskan apa saja yang harus dikonfigurasi di dalam Migration dengan menggunakan perintah :

Php artisan migrate

untuk mengatur database pilih file 2018_05_12_060428_create_barang_table.php di dalam folder database > migration Ketika terbuka file tersebut maka akan terlihat konfigurasi dasar seperti di bawah ini:

```

1  <?php
2
3  use Illuminate\Support\Facades\Schema;
4  use Illuminate\Database\Schema\Blueprint;
5  use Illuminate\Database\Migrations\Migration;
6
7  class CreateUsersTable extends Migration
8  {
9      /**
10       * Run the migrations.
11       *
12       * @return void
13       */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('nik')->unique();
19             $table->string('telp');
20             $table->string('name');
21             $table->string('email')->unique();
22             $table->string('password');
23             $table->enum('role', ['Petugas', 'User'])->default('User');
24             $table->rememberToken();
25             $table->timestamps();
26         });
27     }
28

```

Gambar 3.20 Migration table user

untuk mengatur database pilih file 2018_05_12_060428_create_barang_table.php di dalam folder database > migration Ketika terbuka file tersebut maka akan terlihat konfigurasi dasar seperti di bawah ini:

```

1 <?php
2
3 use Illuminate\Support\Facades\Schema;
4 use Illuminate\Database\Schema\Blueprint;
5 use Illuminate\Database\Migrations\Migration;
6
7 class CreateUsersTable extends Migration
8 {
9     /**
10      * Run the migrations.
11      *
12      * @return void
13      */
14     public function up()
15     {
16         Schema::create('users', function (Blueprint $table) {
17             $table->increments('id');
18             $table->string('nik')->unique();
19             $table->string('telp');
20             $table->string('name');
21             $table->string('email')->unique();
22             $table->string('password');
23             $table->enum('role', ['Petugas', 'User'])->default('User');
24             $table->rememberToken();
25             $table->timestamps();
26         });
27     }
28 }

```

Gambar 3.21 Migration table barang

3.8 Penggunaan Controller

Controller adalah sebuah file php yang bertugas untuk mengoordinasikan antara file dalam view serta operasi database pada model. Operasi yang ditangani adalah seperti pindah tampilan, mengatur session atau sesi di setiap halaman tampilan, serta menyinergikan tampilan dengan data yang diolah pada model. Nama controller dalam aplikasi website kehilangan barang ini adalah

3.9.1. Home Controller

```

13 class HomeController extends Controller
14 {
15     public function __construct()
16     {
17         //$this->middleware('auth');
18     }
19
20     public function index(Request $request)
21     {
22         if (Auth::check()) {
23             return redirect('/admin');
24         }
25         return view('home');
26     }
27
28     public function aturan(Request $request)
29     {
30         return view('aturan');
31     }
32
33     public function item()
34     {
35         $barangs = Barang::where('pelapor_id',null)->get();
36         return view ('barang', ['barangs'=>$barangs]);
37     }
38
39     public function daftar(Request $request)
40     {
41         return view('auth.register')->with([
42             'nik' => session()->get('nik'),
43             'barang_id' => session()->get('barang_id'),
44             'pelapor_id' => session()->get('pelapor_id'),
45             'penemu_id' => session()->get('penemu_id'),
46             'registered' => false
47         ]);
48     }
49
50     public function simpanUser(Request $request)
51     {
52         $data = $request->all();
53         $user = User::create([
54             'nik' => $data['nik'],
55             'name' => $data['name'],
56             'email' => $data['email'],
57             'telp' => $data['telp'],
58             'password' => Hash::make('secret'),
59         ]);
60     }

```

Gambar 3.22 Home Controller

Home controller berfungsi untuk membedakan hak akses setiap user bila hasil Auth::check tidak valid maka interface akan di alihkan ke home.blade , user dapat mengakses menu aturan.blade dan item.blade

3.9.2. Admin Controller

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use App\User;
6  use App\Barang;
7  use App\Mail\BarangEmail;
8  use Illuminate\Http\Request;
9  use Illuminate\Support\Facades\Mail;
10 use Illuminate\Support\Facades\Hash;
11
12 class AdminController extends Controller
13 {
14
15     public function __construct()
16     {
17         $this->middleware('auth');
18     }
19
20     public function index(Request $request)
21     {
22         return view('admin');
23     }
24
25     public function daftar(Request $request)
26     {
27         return view('auth.register')->with([
28             'nik' => session()->get('nik'),
29             'barang_id' => session()->get('barang_id'),
30             'pelapor_id' => session()->get('pelapor_id'),
31             'penemu_id' => session()->get('penemu_id'),
32             'registered' => false
33         ]);
34     }
35 }

```

Gambar 3.23 Admin Controller (1)

Admin controller berfungsi untuk membedakan hak akses setiap Admin diberikan akses Auth interface yang akan di tampilkan adalah admin.blade , admin menggunakan function daftar yang dimana dapat mengakses halaman auth register.blade untuk mendaftarkan admin baru , dan melakukan

```

36 public function simpanUser(Request $request)
37 {
38     $data = $request->all();
39     $user = User::create([
40         'nik' => $data['nik'],
41         'name' => $data['name'],
42         'email' => $data['email'],
43         'telp' => $data['telp'],
44         'password' => Hash::make('secret'),
45     ]);
46
47     $barang = Barang::find($data['barang_id']);
48     $pelapor = $data['pelapor_id'];
49     $penemu = $data['penemu_id'];
50
51     if ($pelapor) {
52         $barang->pelapor_id = $user->id;
53     }
54     else {
55         $barang->penemu_id = $user->id;
56     }
57
58     if ($barang->foto_bukti && $barang->foto_barang) {
59         Mail::to($barang->pelapor)->send(new BarangEmail($barang));
60     }
61
62     $barang->save();
63     return redirect('/barang');
64 }
65 }

```

Gambar 3.24 Admin Controller (2)

Function simpanuser digunakan untuk mengupdate data barang bila ada seorang user yang menemukan barang , melapor kehilangan , claim menemukan barang dan claim kepemilikan barang , jika sebuah laporan kehilangan sudah di buat foto bukti yang akan masuk ke dalam list barang , jika suatu barang di temukan dan foto barang di upload ke list barang , maka system akan mengirim email dengan mengambil data nama barang dari table barang , Lokasi dari table barang, nama user dari table user dan email dari table user

3.9.3. Barang Controller

```

12  class BarangController extends Controller
13  {
14      public function index()
15      {
16          $barangs = Barang::all();
17          return view ('barang', ['barangs'=>$barangs]);
18      }
19      public function kehilangan()
20      {
21          $pelapor = Auth::user();
22          return view ('laporan', ['pelapor'=>$pelapor]);
23      }
24      public function menemukan()
25      {
26          $penemu = Auth::user();
27          return view ('laporan', ['penemu'=>$penemu]);
28      }
29      public function store(Request $request)
30      {
31          $path = $request->file('foto')->store('public/barang');
32          $input = $request->all();
33          $nik = $input['nik'];
34
35          $user = User::where('nik', $nik)->first();
36          if($input['pelapor_id'])
37          {
38              $input['foto_bukti'] = $path;
39              if ($user) {
40                  $input['pelapor_id'] = $user->id;
41              }
42          }
43          else if($input['penemu_id'])
44          {
45              $input['foto_barang'] = $path;
46              if ($user) {
47                  $input['penemu_id'] = $user->id;
48              }
49          }
50      }
51  }

```

Gambar 3.25 Barang Controller 1

Controller barang mengambil data barang dari fuction index , data pelapor dari fuction kehilangan , data penemu dari fuction menemukan dan menentukan Selanjut nya fuction store akan bertugas untuk menata status barang , jika foto yang di upload adalah foto_bukti maka id user yang di ambil adalah pelapor_id. Jika foto yang di upload adalah foto barang maka id user yang di ambil adalah penemu_id

```

51     unset($input['foto']);
52     unset($input['_token']);
53     unset($input['nik']);
54     $barang = Barang::create($input);
55
56
57     if (!$user)
58     {
59         return redirect('/daftar')->with([
60             'nik' => $nik,
61             'barang_id' => $barang->id,
62             'pelapor_id' => $input['pelapor_id'],
63             'penemu_id' => $input['penemu_id']
64         ]);
65     }
66
67     return redirect('/barang');
68 }
69 public function show(Barang $barang)
70 {
71     //
72 }
73 public function edit(Barang $barang)
74 {
75     return view ('laporan', [
76         'barang' => $barang,
77         'pelapor' => $barang->pelapor,
78         'penemu' => $barang->penemu
79     ]);
80 }

```

Gambar 3.26 Barang Controller (2)

foto akan di input ke data barang pada barang.blade, jika user belum terdaftar maka halaman akan di alihkan ke daftar.blade jika pendaftaran selesai maka data foto yang sudah di upload tidak perlu di upload ulang , data user dan foto akan otomatis terdaftar pada barang.blade

Function update adalah function yang di gunakan saat proses claim, jika salah claim yang di lakukan pada laporan yang memiliki id user penemu_id maka proses claim akan di golongankan menjadi claim kepemilikan barang . proses pengiriman email tidak akan dilakukan, karena berdasarkan prosedur penemu sudah ada di lokasi

jika salah claim yang di lakukan pada laporan yang memiliki id user pelapor_id maka proses claim akan di golongankan menjadi claim menemukan barang , proses pengiriman email akan dilakukan, karena berdasarkan prosedur pelapor sudah tidak ada di lokasi, dan perlu di beritau bahwa barang yang di laporkan hilang sudah ada yang menemukan

```

81 public function update(Request $request, Barang $barang)
82 {
83     $ kirimEmail = false;
84     $path = $request->file('foto')->store('public/barang');
85     $nik = $request->get('nik');
86     $user = User::where('nik', $nik)->first();
87
88     $type = 1;
89     if ($barang->foto_bukti)
90     {
91         $barang->penemu_id = $user ? $user->id : $request->user()->id;
92         $barang->foto_barang = $path;
93
94         $ kirimEmail = true;
95         $type = 0;
96     }
97     else
98     {
99         $barang->pelapor_id = $user ? $user->id : $request->user()->id;
100        $barang->foto_bukti = $path;
101        $type = 1;
102    }
103
104
105    if (!$user)
106    {
107        $barang->lokasi = $barang->lokasi ? $barang->lokasi : $request->get('lokasi');
108        $barang->save();
109        return redirect('/daftar')->with([
110            'nik' => $nik,
111            'barang_id' => $barang->id,
112            'penemu_id' => $type == 0 ? $barang->penemu_id : null,
113            'pelapor_id' => $type == 1 ? $barang->pelapor_id : null
114        ]);
115    }
116    else
117    {
118        // if ($ kirimEmail) {
119        //     Mail::to($barang->pelapor)->send(new BarangEmail($barang));
120        // }
121
122        $barang->lokasi = $barang->lokasi ? $barang->lokasi : $request->get('lokasi');
123        $barang->save();
124
125        return redirect('/barang');
126    }
127 }

```

Gambar 3.27 Barang Controller 3

Pengiriman email akan di lakukan dengan cara mengambil email dari user pelapor , memberitahu nama barang yang di laporkan, dan lokasi barang

Setelah proses kedua foto_bukti dan foto_barang di upload maka data laporan akan di hapus dari barang.blade atau list barang untuk user , data barang masih terdaftar di item.blade atau list barang untuk admin

3.9 Proses Input laporan barang hilang

Orientasi aplikasi ini yang menggunakan konsep membatasi hak akses pengunjung dengan sistem merupakan hasil kombinasi dari model dan controller. Proses input data barang hanya bisa dilakukan oleh petugas

Pada saat petugas / Admin memilih salah satu menu : Laporan Menemukan atau Laporan Kehilangan

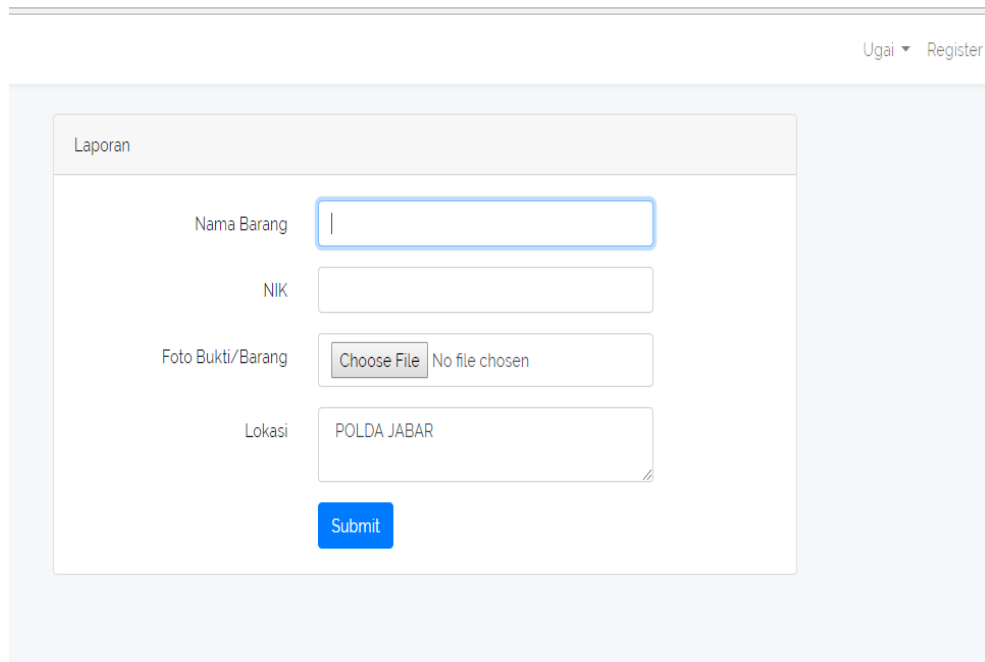


Gambar 3.28 Menu Admin

```
<a href="{{ url('/barang/menemukan') }}"></a>
<a href="{{ url('/barang/kehilangan') }}"></a>
<a href="{{ url('/item') }}"></a>
```

Gambar 3.29 Code Menu Admin

Sistem akan menyambungkan url ke barang/menemukan bila admin memilih Laporan Menemukan Lokasi akan diambil dari lokasi Admin yang bertugas



Ugai ▾ Register

Laporan

Nama Barang

NIK

Foto Bukti/Barang No file chosen

Lokasi

Gambar 3.30 Menu Admin Laporan Menemukan

Sistem akan menyambungkan url ke barang/Kehilangan bila admin memilih Laporan Kehilangan . tidak menggunakan lokasi karena barang belum di temukan

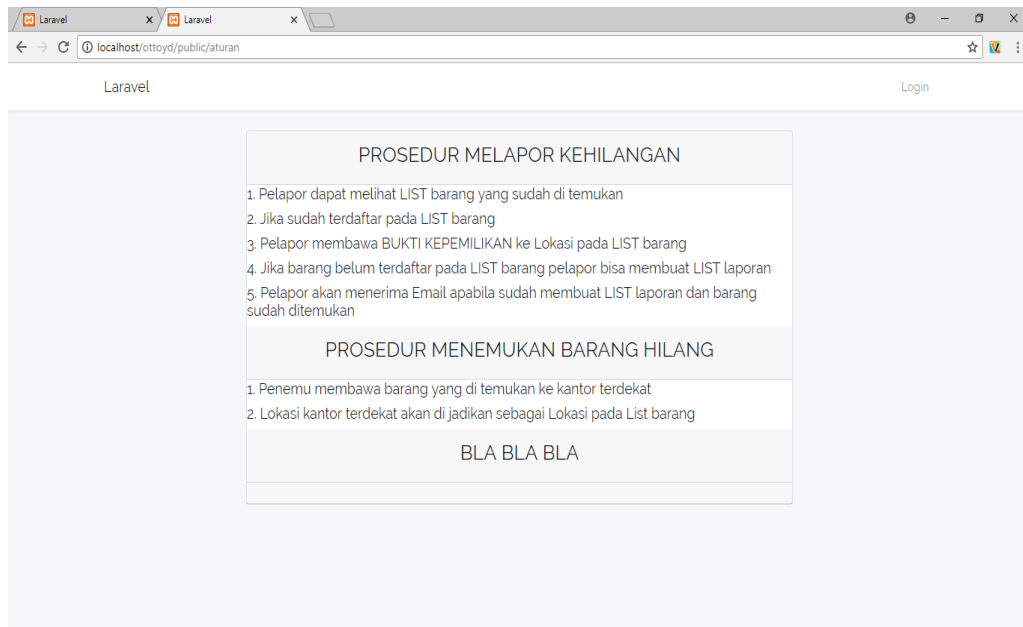
3.11. Uji Coba Penggunaan

Tampilan awal untuk user jika pertama kali mengunjungi website



Gambar 3.31 Menu User

user dapat melihat tata tertib yang berlaku



Gambar 3.32 Aturan User

user dapat melihat barang barang yang sudah di temukan

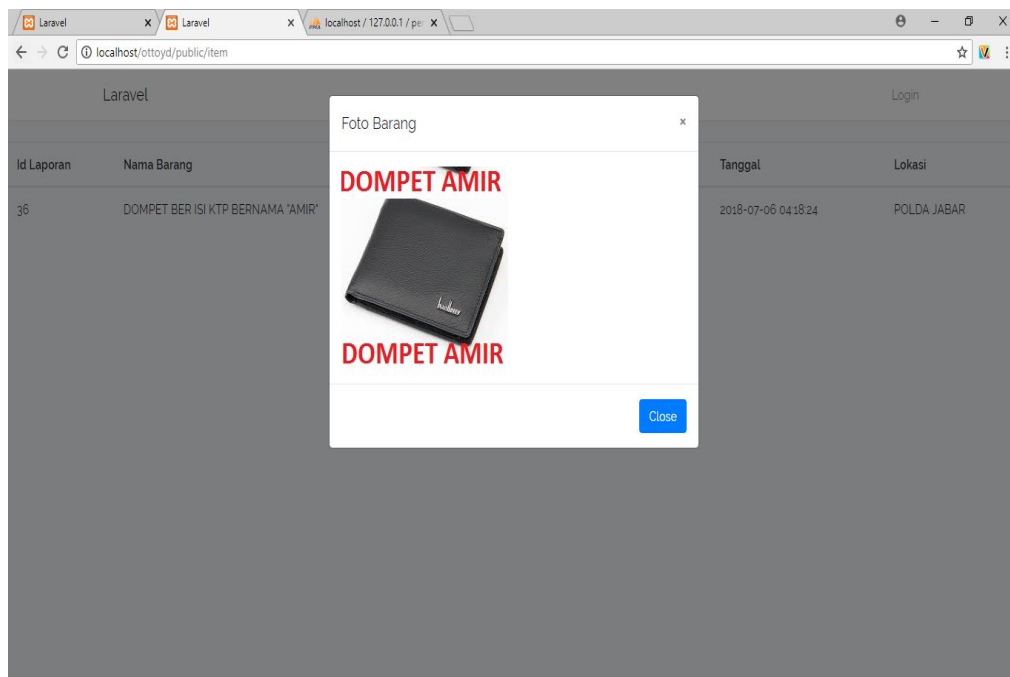
Home							
Login							
Id Laporan	Pelapor	Barang	Tanggal Lapor	Tanggal Claim	Lokasi	Keterangan	Status
2	AMIR	KARTU KTP	2018-08-05 15:30:08	2018-08-05 15:30:51	POLDA JABAR	KTP DENGAN NAMA BUDI	Success
3	Ugai	ktp	2018-08-06 05:15:51	2018-08-06 05:58:30	POLDA JABAR	ditemukan ktp bernama budi	Success
11		SIM MOTOR	2018-08-09 00:53:29		POLDA JABAR	Sim bernama Budi	
12	AMIR		2018-08-09 00:54:29			KTP BUDI	

Gambar 3.33 Daftar barang User

jika ingin meng – claim user dapat datang ke lokasi yang ada di dalam data contoh :

1. seorang bernama amir menjatuhkan dompet di sekitar wilayah jawa barat (misal) di dalam nya ada ktp amir
2. dompet di temukan oleh amar dan di berikan ke porles terdekat ya itu polres jawa barat
3. amir membuka website dan menemukan dompet nya ada di nomor laporan #36

user dapat melihat kondisi barang dengan cara klik pada nama penemu



Gambar 3.34 Foto barang

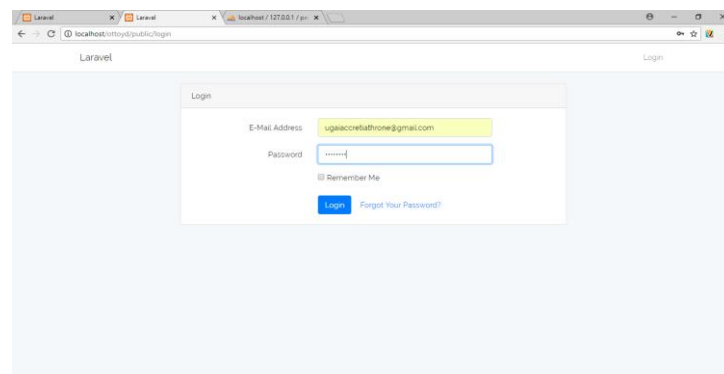
Sistem pelayanan

amir membawa kartu keluarga untuk di cocokan dengan ktp yang berada di dalam dompet



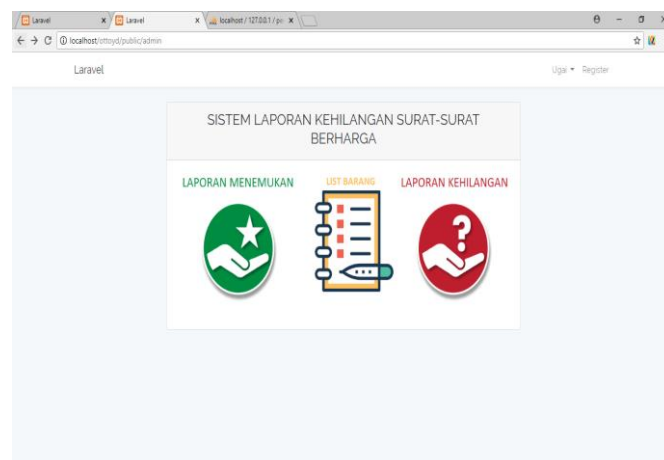
Gambar 3.35 Foto Bukti

petugas memiliki id untuk melakukan login dan memiliki lokasi masing-masing



Gambar 3.36 Login Admin

Interface petugas

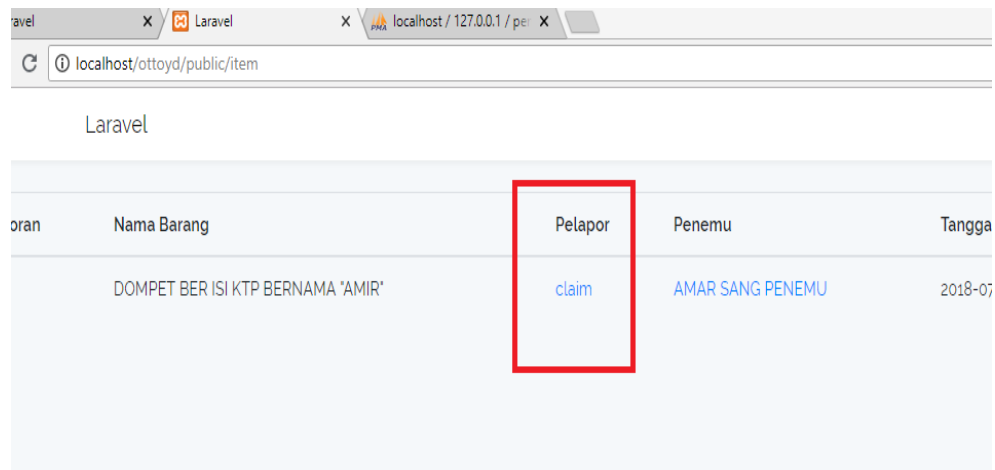


Gambar 3.37 Menu Admin

terdapat 3 menu :

list barang

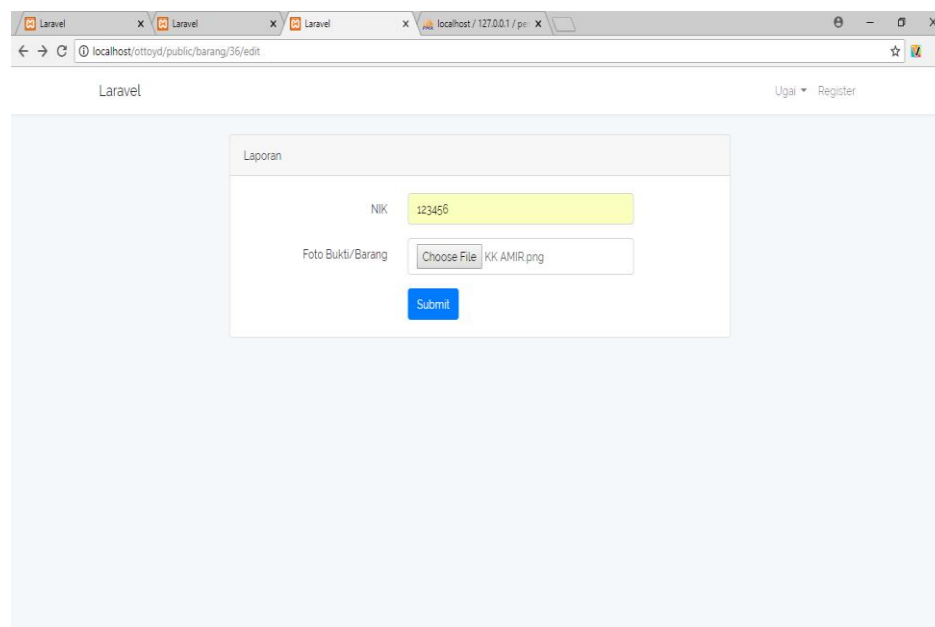
untuk melayani user yang ingin meng-claim barang yang sudah di temukan seperti yang di lakukan oleh amir



Nomor	Nama Barang	Pelapor	Penemu	Tanggal
	DOMPET BER ISI KTP BERNAMA "AMIR"	claim	AMAR SANG PENEMU	2018-07

Gambar 3.38 List barang Admin

jika petugas sudah mem-verifikasi kepemilikan dompet milik amir maka petugas akan meng upload foto bukti kepemilikan dari amir , yang di gunakan di sini adalah ktru keluarga amir



Laporan

NIK: 123456

Foto Bukti/Barang: KK AMIR.png

Gambar 3.39 claim kepemilikan barang

jika user yang nik nya belum terdaftar maka user akan langsung otomatis mendaftar pada saat itu juga

Register

NIK tidak terdaftar, harap daftar terlebih dahulu

NIK

Name

E-Mail Address

Telepon

[Register](#)

Gambar 3.40 Pendaftaran NIK bila belum terdaftar

jika semua sudah selesai maka tampilan nomor laporan #36 akan menjadi seperti ini

36	DOMPET BER ISI KTP BERNAMA 'AMIR'	AMIR	AMAR SANG PENEMU	2018-07-06 04:18:24	POLDA JABAR
----	-----------------------------------	------	------------------	---------------------	-------------

Gambar 3.41 List barang Admin

nomor laporan #36 akan hilang dari interface user

Laravel Login

Id Laporan	Nama Barang	Pelapor	Penemu	Tanggal	Lokasi

Gambar 3.42 List barang User

laporan menemukan

untuk melayani user yang menemukan barang dan membawa barang yang di temukan ke porles terdekat

Laporan

Nama Barang

NIK

Foto Bukti/Barang No file chosen

Lokasi

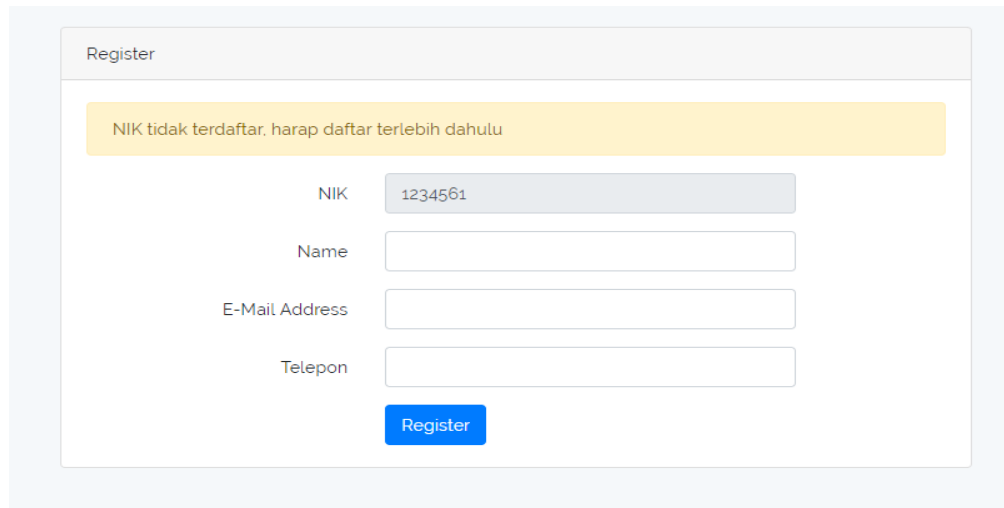
Gambar 3.43 Laporan Menemukan

lokasi barang akan otomatis terisi dari lokasi petugas yang melayani user yang menemukan barang

users	nik	telp	name	email	password	role	lokasi
performance_schema	123123	02187908195	Ugai	ugalaccretlathrone@gmail.com	\$2y\$10\$46l0mNNs2xY5K9EUyPTNU LYgZa0/jpGzUR83CJmS0O...	User	POLDA JABAR
	123456	02187908195	AMIR	amir@yahoo.com	\$2y\$10\$015hDI6J85LrukYgAsGyuur/Ht2yOHcr9CWlzw1RHOL...	User	NULL

Gambar 3.44 Database Lokasi Admin

jika user yang nik nya belum terdaftar maka user akan langsung otomatis mendaftar pada saat itu juga



Register

NIK tidak terdaftar, harap daftar terlebih dahulu

NIK 1234561

Name

E-Mail Address

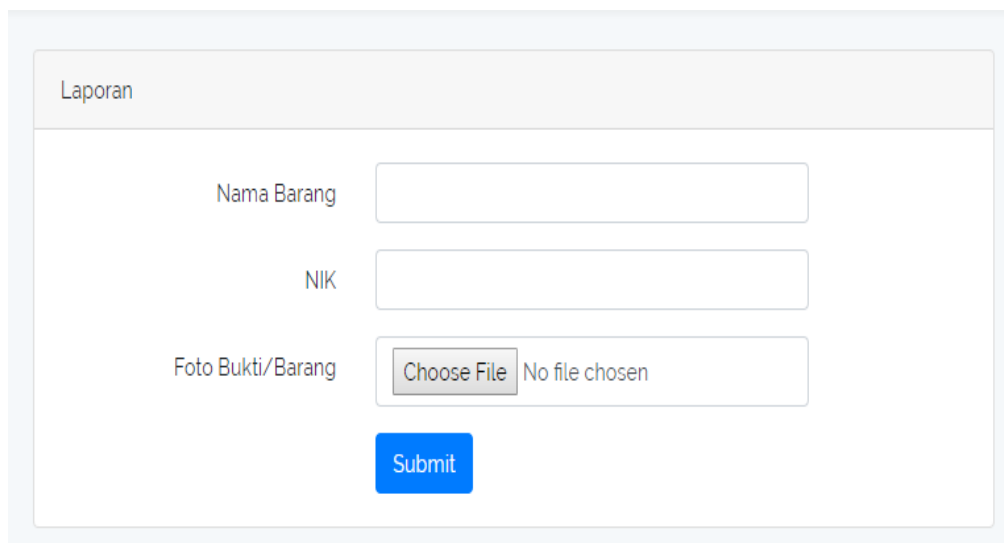
Telepon

Register

Gambar 3.45 Pendaftaran NIK bila belum terdaftar

laporan kehilangan

untuk melayani user yang kehilangan barang dan melapor ke porles terdekat



Laporan

Nama Barang

NIK

Foto Bukti/Barang

Choose File No file chosen

Submit

Gambar 3.46 Laporan Kehilangan

jika user yang nik nya belum terdaftar maka user akan langsung otomatis mendaftar pada saat itu juga

Register

NIK tidak terdaftar, harap daftar terlebih dahulu

NIK 1234561

Name

E-Mail Address

Telepon

Register

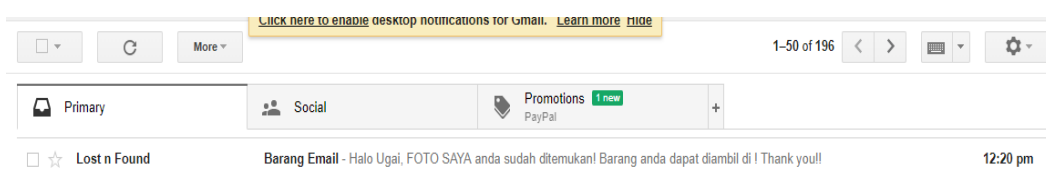
Gambar 3.47 Pendaftaran NIK bila belum terdaftar

laporan tentang kehilangan akan di buat dengan nomor laporan #37
nama penemu dan lokasi di kosongkan karena barang belum ada yang menemukan

37	FOTO SAYA	Ugai	claim	2018-07-06 05:11:16
----	-----------	------	-------	---------------------

Gambar 3.48 Laporan Kehilangan

jika barang yang hilang sudah ada yang menemukan maka , pelapor akan menerima kiriman email ber isi lokasi barang



Gambar 3.49 Menerima Email

tampilan list barang nomor #37

Id Laporan	Nama Barang	Pelapor	Penemu	Tanggal	Lokasi
34	SIM MOTOR	AMIR	claim	2018-07-06 04:11:30	
36	DOMPET BER ISI KTP BERNAMA 'AMIR'	AMIR	Ugai	2018-07-06 04:18:24	POLDA JABAR
37	FOTO SAYA	Ugai	AMIR	2018-07-06 05:11:16	POLDA JABAR

Gambar 3.50 Laporan Admin (barang.blade)