



**PROYEK AKHIR - VE180626**

**DIGITALISASI MAGANG DEPARTEMEN TEKNIK  
ELEKTRO OTOMASI BERBASIS WEB MENGGUNAKAN  
FRAMEWORK LARAVEL**

**Khabib Nur Budiono  
NRP. 1031160000049**

**Dosen Pembimbing  
Ir. Joko Susila, MT.  
Ciptian Wieried Priananda S.ST., MT.**

**DEPARTEMEN TEKNIK ELEKTRO OTOMASI  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020**

-----Halaman ini sengaja dikosongkan-----



**FINAL PROJECT - VE180626**

***INTERNSHIP DIGITALIZATION DEPARTEMENT OF  
ELECTRICAL AUTOMATION ENGINEERING WEB  
BASED USING LARAVEL FRAMEWORK***

Khabib Nur Budiono  
NRP. 10311600000049

Supervisor  
Ir. Joko Susila, MT.  
Ciptian Wieried Prianda S.ST., MT.

DEPARTEMENT OF ELECTRICAL AUTOMATION ENGINEERING  
Vocations Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020

-----Halaman ini sengaja dikosongkan-----

## PERNYATAAN KEASLIAN PROYEK AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Proyek Akhir saya dengan judul **Digitalisasi Magang Departemen Teknik Elektro Menggunakan Framework Laravel** adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, 20 Januari  
2020



Khabib Nur Budiono  
NRP. 1031160000049

-----Halaman ini sengaja dikosongkan-----

**DIGITALISASI MAGANG DEPARTEMEN TEKNIK  
ELEKTRO OTOMASI BERBASIS WEB MENGGUNAKAN  
FRAMEWORK LARAVEL**

**PROYEK AKHIR**

Diajukan Guna Memenuhi Sebagian Persyaratan  
Memperoleh Gelar Ahli Madya Teknik  
Pada  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember

Menyetujui:

Dosen Pembimbing I

Dosen Pembimbing II



*3/2020*  
*1/2*

**Ciptian Weriad Priananda, S.ST., MT.**  
NIP. 1990201711060

**SURABAYA**  
**Januari, 2020**

-----Halaman ini sengaja dikosongkan-----

# DIGITALISASI MAGANG DEPARTEMEN TEKNIK ELEKTRO OTOMASI BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL

**Nama : Khabib Nur Budiono**  
**Pembimbing : Ir. Joko Susila, MT.**  
**Ciptian Weried Priyananda S.ST., MT.**

## ABSTRAK

Magang adalah salah satu dari mata kuliah DTEO-FV ITS. Sebelum dilakukan magang para peserta didik diharuskan untuk membuat proposal yang ditujukan pada perusahaan. Ketika magang mahasiswa melakukan *report* kepada dosen pembimbing. Setelah selesai kegiatan magang, mahasiswa diharuskan membuat laporan agar bisa dinilai oleh dosen yang bersangkutan. Tujuan dari pembuatan proyek akhir ini adalah melakukan digitalisasi proses tersebut

Digitalisasi Magang Departemen Teknik Elektro Otomasi dibuat menggunakan bahasa skrip *Hypertext Preprocessor* atau biasa disebut PHP yang menerapkan framework Laravel dan diprogram menggunakan *software* Visual Code Studio. Selain itu untuk memperindah tampilan juga menggunakan bahasa CSS atau *Cascading Style Sheet* dan didukung *bootstrap* klorofil dan *datatables*.

Hasil dari tugas akhir ini dapat diakses pada [www.magangdteo.com](http://www.magangdteo.com). Pada *website* ini akun pengguna mahasiswa bisa ditambahkan dengan register maupun ditambahkan oleh admin. Setelah itu mahasiswa dapat memasukkan data dan berkas yang diperlukan ke *database* web. Dosen dapat melihat data yang dimasukkan oleh mahasiswa dan memberikan nilai pada mahasiswa. Data yang diunggah oleh mahasiswa juga akan bisa diunduh oleh dosen. Selain itu terdapat admin atau operator yang bertujuan sebagai pengawas pada *website* ini.

**Kata kunci : Digitalisasi Magang, Laravel , Basis Data, Website**

-----Halaman ini sengaja dikosongkan-----

**INTERNSHIP INFORMATION SYSTEM DEPARTEMENT OD  
ELECTRICAL AUTOMATION ENGINEERING WEB BASED  
USING LARAVEL FRAMEWORK**

**Name : Khabib Nur Budiono**  
**Supervisor : Ir. Joko Susila, MT**  
**Ciptian Weried Priananda S.ST., MT.**

**ABSTRACT**

*The internship is one of ITS DTEO-FV courses. Before doing an internship the students are required to make a proposal addressed to the company. When the student interns do a report to the supervisor. After completing the internship, students are required to make a report so that it can be assessed by the lecturer in question. The purpose of making this final project is to digitize the process*

*Intership Digitizing the Department of Electrical Engineering Automation is created using the Hypertext Preprocessor scripting language or commonly called PHP which implements the Laravel framework and is programmed using Visual Code Studio software. In addition to beautifying the display also uses CSS or Cascading Style Sheet language and supported by chlorophyll bootstrap and datatables.*

*The results of this final project can be accessed at [www.magangdteo.com](http://www.magangdteo.com). On this website student user accounts can be added by registers or added by the admin. After that students can enter the data and files needed into the web database. Lecturers can see the data entered by students and give grades to students. Data uploaded by students will also be able to be downloaded by lecturers. In addition there are administrators or operators who aim as supervisors on this website.*

**Keywords: Internship Digitizing, Laravel, Database, Website**

-----Halaman ini sengaja dikosongkan-----

## **KATA PENGANTAR**

Segala puji syukur penulis panjatkan kepada Allah SWT yang selalu memberikan kemudahan, kelancaran, dan kerahmatan sehingga penulis dapat menyelesaikan proyek akhir ini dengan baik. Proyek Akhir ini disusun untuk memenuhi persyaratan guna menyelesaikan pendidikan Diploma-III pada Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember, Surabaya dengan judul :

### **Digitalisasi Magang Departemen Teknik Elektro Otomasi Menggunakan *Framework* Laravel**

Dalam pelaksanaan Proyek Akhir dan penyusunan laporan ini penulis memperoleh banyak bantuan serta bimbingan dari berbagai pihak. Oleh karena itu pada kesempatan ini penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah banyak membantu penulis.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Proyek Akhir ini. Akhir kata, semoga Proyek Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 22 Januari 2020

Khajib Nur Budiono

-----Halaman ini sengaja dikosongkan-----

## DAFTAR ISI

SAMPUL LUAR .....	i
SAMPUL DALAM .....	iii
PERNYATAAN KEASLIAN PROYEK AKHIR .....	v
HALAMAN PENGESAHAN .....	vii
ABSTRAK .....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR .....	xvii
DAFTAR TABEL .....	xvii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Tujuan .....	2
1.4 Sistematika Laporan .....	3
1.5 Relevansi .....	3
1.6 Batasan Masalah .....	3
BAB II TEORI PENUNJANG .....	3
2.1 Magang .....	5
2.2 <i>Framework</i> .....	5
2.3 MVC (Model View Controller) .....	6
2.4 Laravel .....	7
2.5 <i>Website</i> .....	10
2.6 Plesk .....	10
2.7 Maria DB .....	12
2.8 PHP .....	13
2.9 PhpMyAdmin .....	13
2.10 MySQL .....	14
2.11 CSS .....	16
2.12 Model Proses .....	16
2.12.1 <i>Usecase Diagram</i> .....	16
2.12.2 <i>Class Diagram</i> .....	18
2.12.3 <i>Activity Diagram</i> .....	18
2.13 Bootstrap .....	19
BAB III PERENCANAAN DAN IMPLEMENTASI .....	23
3.1 Studi Literatur .....	25
3.2 <i>Usecase Diagram</i> .....	25

3.3	Database Menggunakan PhpMyAdmin .....	29
3.4	Pembuatan <i>Website</i> dengan <i>Framework Laravel</i> . .....	31
3.3.1	<i>Routes</i> .....	31
3.3.1	<i>Controller</i> .....	32
3.3.1	<i>View</i> .....	36
BAB IV	PENGUJIAN DAN ANALISIS .....	41
4.1	Rancangan Antarmuka di <i>Web</i> dan <i>Localhost</i> .....	41
4.2	Perbandingan Rancangan Antarmuka di <i>Web</i> dan Localhost .....	50
4.3	Data Mahasiswa Masuk Dalam <i>Database</i> .....	50
4.4	Testing User.....	51
4.5	Mengganti Password Akun.....	52
BAB V	PENUTUP .....	53
5.1	Kesimpulan .....	53
5.2	Saran.....	53
DAFTAR	PUSTAKA .....	55
LAMPIRAN	A .....	A-1
LAMPIRAN	B.....	B-1
RIWAYAT	HIDUP PENULIS.....	B-13

## DAFTAR GAMBAR

Gambar 2. 1	Model Hubunga MVC .....	6
Gambar 3. 1	Diagram Alur Login .....	23
Gambar 3. 2	Diagram Alur <i>Web</i> .....	24
Gambar 3. 3	<i>Usecase</i> Diagram Mahasiswa.....	26
Gambar 3. 4	<i>Usecase</i> Diagram Dosen .....	27
Gambar 3. 5	<i>Usecase</i> Diagram Admin .....	28
Gambar 3. 6	Model Hubungan MVC .....	31
Gambar 3. 7	Listing Program web.php.....	32
Gambar 3. 8	Jenis Jenis <i>Controller</i> .....	33
Gambar 3. 9	Listing Program AuthController.php.....	33
Gambar 3. 10	Listing Program SiteController.php.....	34
Gambar 3. 11	Listing ChangePassword.php .....	34
Gambar 3. 12	Listing Program DosenController.php.....	35
Gambar 3. 13	Listing Program SiswaController.php.....	35
Gambar 3. 14	<i>File View</i> pada <i>Dashboard</i> .....	36
Gambar 3. 15	<i>File View</i> pada <i>Layouts</i> .....	37
Gambar 3. 16	<i>File View</i> pada <i>Auth</i> .....	37
Gambar 3. 17	<i>File View</i> pada Dosen .....	38
Gambar 3. 18	<i>File View</i> pada Mahasiswa .....	38
Gambar 3. 19	<i>File View</i> pada <i>Site</i> .....	39
Gambar 4. 1	Antarmuka Login.....	41
Gambar 4. 2	Antarmuka Registrasi Mahasiswa .....	42
Gambar 4. 3	Antarmuka <i>Dashboard</i> .....	43
Gambar 4. 4	Antarmuka Profil Mahasiswa .....	43
Gambar 4. 5	Antarmuka Edit Siswa .....	44
Gambar 4. 6	Antarmuka Profil Dosen Pribadi .....	45
Gambar 4. 7	Antarmuka Daftar Bimbingan .....	45
Gambar 4. 8	Antarmuka Daftar Dosen .....	46
Gambar 4. 9	Antarmuka Profil Dosen .....	46
Gambar 4. 10	Antarmuka Register Dosen Oleh Admin .....	47
Gambar 4. 11	Daftar Mahasiswa .....	47
Gambar 4. 12	Antarmuka Profil Mahasiswa Pada User Lain .....	48
Gambar 4. 13	Register Mahasiswa Oleh Admin.....	49
Gambar 4. 14	Antarmuka Daftar Berkar-Berkas Mahasiswa .....	49
Gambar 4. 15	Antarmuka Mengganti Password .....	54

Gambar 4. 15 Error Download File Mahasiswa.....	55
Gambar 4. 15 Error Email Ganda.....	55

## DAFTAR TABEL

Tabel 2. 1	Tabel <i>Usecase Diagram</i> .....	17
Tabel 2. 2	Tabel <i>Activity Diagram</i> .....	19
Tabel 3. 1	Tabel User.....	29
Tabel 3. 2	Tabel Mahasiswa.....	29
Tabel 3. 3	Tabel Mapel .....	30
Tabel 3. 4	Tabel Jenis Mapel.....	30
Tabel 3. 5	Tabel Dosen .....	30
Tabel 3. 6	Tabel Relai Mahasiswa dan Mapel .....	30
Tabel 4.1	Tabel Database Mahasiswa.....	50
Tabel 4.2	Tabel Database <i>Upload</i> dan <i>Download</i> .....	51
Tabel 4.3	Tabel Registrasi Mahasiswa .....	52
Tabel 4.4	Tabel Registrasi Mahasiswa Oleh Admin .....	52
Tabel 4.5	Tabel Registrasi Dosen .....	52
Tabel 4.6	Tabel Register Akun Baru .....	53
Tabel 4.7	Pengujian Edit Data Mahasiswa.....	53
Tabel 4.8	Tabel Hasil Ganti Password.....	54



# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pendidikan memiliki peran utama dalam upaya peningkatan kualitas sumber daya manusia menjadi lebih baik. Pendidikan diharapkan mampu untuk membentuk karakter seseorang untuk dapat mengembangkan sikap, prilaku, keterampilan dan kecerdasan. Pendidikan terbagi menjadi dua bidang yaitu pendidikan akademik dan non akademik. Pendidikan akademik meliputi penguasaan dan pengembangan disiplin ilmu pengetahuan, teknologi dan seni. Pendidikan akademik lebih diutamakan sebagai penerapan ilmu pengetahuan yang diarahkan menjadi sebuah profesi. Pendidikan nonakademik meliputi sesuatu hal yang bersifat ilmiah tetapi tidak terikat pada teori tertentu, kemampuan non-akademik tidak dapat diukur secara pasti seperti pendidikan akademik, karena tidak terdapat benar maupun salah didalam pelaksanaannya.

Perguruan tinggi merupakan satuan penyelenggara pendidikan sebagai tingkat lanjut dari jenjang pendidikan menengah atas dengan dasar landasan visi dan misi. Visi merupakan gambaran dan tujuan suatu perguruan tinggi di masa depan, sedangkan misi bagaimana cara untuk mencapai masa depan tersebut. Tri Dharma Perguruan Tinggi merupakan satu diantara misi diseluruh perguruan tinggi di Indonesia. Tri Dharma Perguruan Tinggi menjadi tujuan pencapaian yang dilakukan setiap perguruan tinggi, yang terdiri dari poin, yaitu: 1. Pendidikan dan Pengajaran, yang mana pendidikan menjadi sebuah aktifitas pengembangan diri melalui pengalaman, sedangkan pengajaran aktifitas mengarahkan bagaimana pengembangan diri dengan pengalaman yang dimiliki; 2. Penelitian, Penelitian dalam ruang lingkup mahasiswa merupakan penerapan sebuah teori tertentu untuk mencari jalan keluar sebuah masalah, hal ini menjadi unsur sebuah tugas akhir mahasiswa dari pendidikan di perguruan tinggi; 3. Pengabdian kepada Masyarakat, dalam perguruan tinggi pengabdian kepada masyarakat merupakan kegiatan yang menerapkan ilmu pengetahuan untuk memajukan kesejahteraan masyarakat dan mencerdaskan kehidupan bangsa.

Salah satu kegiatan penelitian sekaligus pengabdian kepada masyarakat yang dilakukan perguruan tinggi adalah Magang.

Magang memiliki berbagai keuntungan. Pada Mahasiswa memiliki manfaat seperti mahasiswa dapat mengaplikasikan dan meningkatkan ilmu yang diperoleh di bangku perkuliahan, menambah wawasan setiap mahasiswa mengenai dunia industri, Menambah dan meningkatkan keterampilan serta keahlian dibidang praktek. Manfaat bagi institut terjalannya kerjasama “bilateral” antara Universitas dengan perusahaan, dapat meningkatkan kualitas lulusannya melalui pengalaman kerja Magang, dikenal di dunia industri. Manfaat bagi Perusahaan adanya kerjasama antara dunia pendidikan dengan dunia industri/ perusahaan sehingga perusahaan tersebut dikenal oleh kalangan akademis. Memperoleh kritikan-kritikan yang membangun dari mahasiswa-mahasiswa yang melakukan Praktek Magang, perusahaan akan mendapat bantuan tenaga dari mahasiswa- mahasiswa yan melakukan praktek, adanya orang yang mengaudit perusahaan tanpa mengeluarkan biaya dengan adanya laporan-laporan magang yang diberikan kepada perusahaan, menghemat pengeluaran, mendapatkan masa percobaan gratis.

Proses pencarian magang adalah dimulai dengan memenuhi persyaratan untuk mengikuti atau mengambil SKS magang yang berjumlah 12 SKS untuk kurikulum lama dan 14 SKS untuk kurikulum baru. Persyaratan untuk mengikuti magang diatur setiap departemen yang ada. Tujuan pembentukan dari Digitalisasi Magang Departemen Teknik Elektro Otomasi adalah mendokumentasikan data-data yang ada apabila dikemudian hari memerlukan data yang ada.

## **1.2 Permasalahan**

Perumusan masalah yang dibahas dalam pengerjaan Proyek Akhir ini adalah sebagai berikut:

1. Dibutuhkannya digitalisasi maupun *database* untuk kegiatan Magang.
2. Bagaimana merancang dan mengembangkan Digitalisasi Magang berbasis web menggunakan framework Laravel

## **1.3 Tujuan**

Tujuan yang dicapai dalam penyelesaian pada Proyek Akhir ini adalah sebagai berikut

1. Merancang dan mengembangkan Digitalisasi Magang berbasis web menggunakan framework Laravel
2. Membuat *database* untuk kegiatan magang.

#### 1.4 Sistematika Laporan.

Untuk pembahasan lebih lanjut, laporan Proyek Akhir ini disusun dengan sistematika sebagai berikut:

##### **Bab I PENDAHULUAN**

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi Proyek Akhir yang dibuat.

##### **Bab II TEORI PENUNJANG**

Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

##### **Bab III PERENCANAAN DAN IMPLEMENTASI**

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi desain mekanik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

##### **Bab IV PENGUJIAN DAN ANALISA**

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

##### **Bab V PENUTUP**

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Proyek Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

#### 1.5 Relevansi

Memudahkan Mahasiswa dalam proses pendaftaran dan pelaporan pelaksanaan Magang, memudahkan dosen memberi nilai, mendapatkan salinan *file* Mahasiswa.

#### 1.6 Batasan Masalah

Beberapa batasan masalah terdapat pada tugas akhir ini, antara lain adalah:

1. Penggunaan *framework* Laravel dalam pembuatan *website* ini.
2. Proyek akhir ini berfokus pada digitalisasi magang DTEO FV-ITS
3. Penggunaan Laravel 6.2 mengakibatkan tidak bisa menggunakan *shared hosting* sehingga *system hosting* yang digunakan adalah VPS dengan Plesk.

## **BAB II**

### **TEORI PENUNJANG**

#### **2.1 Magang**

Menurut KBBI Magang memiliki dua arti. Pertama arti dari kata magang adalah calon pegawai (yang belum diangkat secara tetap serta belum menerima gaji atau upah karena dianggap masih dalam taraf belajar). Kedua arti dari kata magang adalah Calon ahli.

Magang memiliki berbagai keuntungan bagi Mahasiswa. Pada Mahasiswa memiliki manfaat seperti mahasiswa dapat mengaplikasikan dan meningkatkan ilmu yang diperoleh di bangku perkuliahan, menambah wawasan setiap mahasiswa mengenai dunia industri, Menambah dan meningkatkan keterampilan serta keahlian dibidang praktek. Manfaat bagi institut terjalannya kerjasama “bilateral” antara Universitas dengan perusahaan, dapat meningkatkan kualitas lulusannya melalui pengalaman kerja Magang, dikenal di dunia industri.[1]

#### **2.2 Framework**

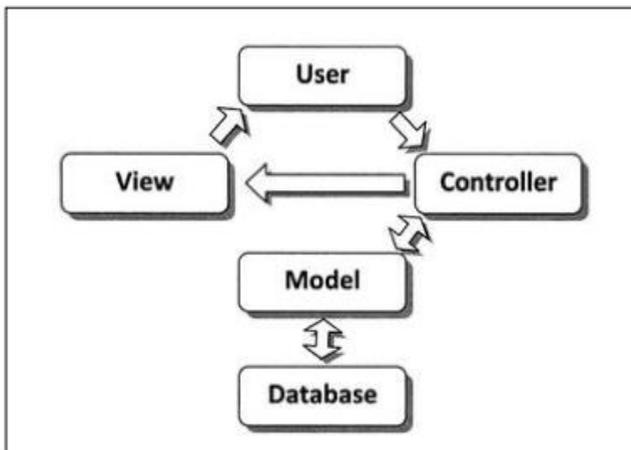
Menurut Wardana (2010:3), Framework adalah kumpulan perintah atau fungsi dasar yang membentuk aturan-aturan tertentu dan saling berinteraksi satu sama lain, sehingga dalam pembuatan aplikasi website harus mengikuti aturan dari framework tersebut. Dengan framework (dalam hal ini framework PHP), kita tidak perlu memikirkan kode perintah/fungsi dasar karena hal-hal penunjang lainnya seperti koneksi database, validasi form, GUI, dan keamanan telah disediakan oleh framework. Keuntungan yang dapat diperoleh dari penggunaan framework adalah :

1. Waktu pembuatan aplikasi kita jauh lebih singkat.
2. Kode aplikasi website menjadi lebih mudah dibaca, karena sedikit dan sifatnya pokok. Detailnya adalah kode dari framework dan ini mungkin tidak perlu dipikirkan, terjamin.
3. Website menjadi lebih mudah diperbaiki, karena tidak perlu fokus ke semua komponen kode website, terutama kode system framework.

4. Kita tidak perlu lagi membuat kode penunjang aplikasi website seperti koneksi database, validasi form, GUI, dan keamanan.10
5. Pikiran menjadi lebih terfokus ke kode alur permasalahan website, apa yang ditampilkan dan layanan apa saja yang diberikan dari aplikasi website tersebut.
6. Jika dikerjakan team work, makan akan lebih terarah karena system framework, mengharuskan adanya keteraturan peletakan kode. Seperti bagian pengambilan database terpisah. [2]

### 2.3 MVC (*Model View Controller*)

Menurut Wardana (2010), MVC adalah sebuah pola pemrograman yang bertujuan memisahkan logika bisnis, logika data, dan logika tampilan (interface), atau secara sederhana memisahkan antara proses, data dan tampilan. MVC mengatur arsitektur sebuah aplikasi. Keuntungan dari MVC adalah kemudahan maintenance, dan pengembangan aplikasi tersebut. Berikut penjelasan dari model hubungan MVC diatas menurut Wardana yang di tulis pada buku Menjadi Master Php dengan *Framework Codeignitor* pada halaman 53.. Wardana (2010) :



Gambar 2.1 Model Hubungan MVC dalam Buku Menjadi Master Php dengan *Framework Codeignitor* pada halaman 53

1. View adalah komponen dari suatu MVC yang bertugas menampilkan apa yang harus ditampilkan ke pengunjung website. Isinya dapat berupa form, table, gambar, animasi ataupun lainnya yang boleh dilihat user (pengunjung).
2. Model adalah komponen MVC yang bertugas mengambil data dari database dan juga memasukkan data ke database. Isi umumnya berupa perintah SQL. Hasilnya dikirimkan ke Controller.
3. Controller adalah komponen MVC yang bertugas mengirim perintah ke model untuk mendapatkan data yang diinginkan. Controller tidak mengetahui bagaimana data tersebut diambil dari database, karena Controller tidak berisi kode perintah SQL. Karena itu adalah tugas model, Controller mengolah data dari inputan user dan data dari model kemudian data olahan tersebut dikirimkan ke view untuk ditampilkan sesuai aturan Controller. Controller merupakan penghubung antara model dan view [2]

## 2.4 Laravel

Menurut (Pribadi, 2016), Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, dan untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaks yang ekspresif, jelas dan menghemat waktu.

Artisan adalah baris perintah yang ada di laravel untuk menjalankan sebuah perintah tertentu yang diakses melalui CMD (command prompt). Kita bisa membuat banyak hal dengan artisan, baik membuat model, controller ataupun yang lainnya dengan mudah dan cepat.

Untuk mengetahui lebih lanjut mengenai fungsi artisan di laravel cukup jalankan perintah "php artisan" di dalam project Anda, nanti akan tampil daftar list apa saja yang dapat dilakukan dengan perintah artisan di laravel.

```
php artisan help //informasi bantuan
```

```
php artisan tinker //untuk mengakses shell di laravel
php artisan -v //untuk cek versi laravel yang digunakan
php artisan make:model nama//membuat model tabel
php artisan make:controller nama//membuat controller
php artisan serve //untuk menjalankan laravel
php artisan migrate //untuk migrate database
php artisan make:auth //untuk membuat otentifikasi
php artisan make:migration nama//untuk membuat migrasi
```

Laravel menawarkan beberapa keuntungan ketika Anda mengembangkan website menggunakan dasar framework ini. Pertama, website menjadi lebih scalable (mudah dikembangkan). Kedua, terdapat namespace dan tampilan yang membantu Anda untuk mengorganisir dan mengatur sumber daya website. Ketiga, proses pengembangan menjadi lebih cepat sehingga menghemat waktu karena Laravel dapat dikombinasikan dengan beberapa komponen dari framework lain untuk mengembangkannya website.

Composer merupakan tool yang di dalamnya terdapat dependencies dan kumpulan library. Seluruh dependencies disimpan menggunakan format file composer.json sehingga dapat ditempatkan di dalam folder utama website. Inilah mengapa composer terkadang dikenal dengan dependencies management. Pertanyaannya lain, apa itu dependencies management? Misalnya Anda mempunyai sebuah website yang membutuhkan sebuah library. Saya ambil contoh library untuk mengimplementasikan validasi dan proteksi untuk spamming, yaitu Google reCaptcha. Tentu saja untuk menyediakan Google reCaptcha tidak bisa menggunakan satu library saja, tapi membutuhkan beberapa library. Anda tidak mungkin menginstall satu per satu library. Composer membantu Anda untuk menginstall library yang dibutuhkan oleh library Google reCaptcha. Jadi jika menggunakan composer Anda tinggal menginstall library Google reCaptcha dan secara otomatis library lain akan terinstall. Begitu pun ketika ingin memperbarui library, Anda cukup menggunakan perintah “\$ composer update” dan satu per satu library akan diperbarui secara otomatis.

Keuntungan menggunakan Laravel. Template Layout Ringan Laravel Framework menyediakan template layout yang ringan dan terdapat juga fitur untuk menambahkan template

tambahan yang ringan. Developer dapat membuat dan melakukan pengembangan menggunakan layout yang cukup powerful. Berbagai macam layout ini ini dapat developer gunakan dengan menambahkan berbagai macam CSS, gambar, dan teks dengan struktur kode yang lengkap. Template layout mempunyai peranan yang sangat penting di dalam segala macam aplikasi. Di aplikasi Laravel sendiri, template engine mempunyai banyak bagian di dalam proses pembuatan template. Mempunyai Banyak Library Object Oriented Framework Laravel mempunyai library Object Oriented yang sangat banyak. Selain itu, Laravel juga mempunyai berbagai macam library yang tidak ada di dalam framework PHP populer lainnya. Salah satu library yang paling penting adalah library otentikasi. Anda memerlukan library ini untuk menyediakan fitur otentikasi yang canggih. Library otentikasi Laravel juga menyediakan Bcrypt hashing yang merupakan salah satu fitur tercanggih di bidang otentikasi. Mendukung Framework MVC Laravel adalah salah satu dari sekian framework yang mendukung framework MVC (Model-View-Controller). MVC menangani hampir keseluruhan bagian aplikasi Laravel. Pemisahan antara logika dan tampilan aplikasi sangat mungkin bisa dilakukan menggunakan framework MVC. Selain itu, performa yang optimal, dokumentasi yang lengkap, dan berbagai macam aplikasi tambahan tersedia di dalam framework Laravel yang terintegrasi dengan MVC ini. Tersedia Tool Artisan Developer terkadang membutuhkan interaksi antara Laravel dengan framework lain menggunakan command line yang berguna untuk membuat dan menangani environment di proyek Laravel. Laravel mempunyai tool Artisan (sudah dibahas di bagian sebelumnya) yang mana merupakan tool command line. Fungsi utama dari Artisan adalah melakukan tugas repetisi dan memproses pemrograman yang kompleks yang mana terkadang developer terkadang mengabaikannya selama proyek berjalan. Dengan bantuan Artisan, developer dapat membuat dan melakukan migrasi database dengan mudah.

Pengelolaan struktur database dapat ditangani selama proses migrasi. Pembagian Modul Secara Independen Satu aplikasi Laravel dapat menggunakan lebih dari 20 modul library dan setiap modul terbagi menjadi beberapa modul individu. Modul-modul ini

secara independen mengadopsi prinsip PHP modern dan menjaga fungsi penting dari aplikasi Laravel. Modul tersebut akan membuat aplikasi menjadi lebih informatif, modular, dan responsif. Pengecekan Menggunakan Unit Testing Pembuatan aplikasi Laravel terkadang menggunakan konsep “Unit Testing”. Dengan bantuan Unit Testing, jumlah pengetesan dalam jumlah banyak dapat dilakukan dengan tujuan untuk memastikan perubahan baru dapat dibuat oleh developer tanpa harus merusak satu pun fungsi aplikasi dan membuat aplikasi lebih responsif. [4]

## **2.5 Website**

Menurut Hidayat (2010:2), Website atau situs dapat diartikan sebagai kumpulan halaman-halaman yang digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya, baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait , yang masing-masing dihubungkan dengan jaringan-jaringan halaman. Menurut No Name (2010:1). Website merupakan sebuah halaman berisi informasi yang dapat dilihat jika komputer Anda terkoneksi dengan internet.

Dari definisi di atas dapat disimpulkan bahwa website adalah halaman-halaman yang digunakan untuk menampilkan informasi yang dapat dilihat saat anda terkoneksi dengan internet. [5]

## **2.6 Plesk**

Pengoperasian pada web ini saya menggunakan Plesk. Plesk adalah salah satu perangkat lunak untuk pengaturan pada web hosting. Pada rilis pertama, Plesk berada di bawah perusahaan AS Plesk Inc dan dirancang di Novosibirsk, Rusia. Plesk saat ini tersedia bagi OS server Windows dan Linux. Keduanya adalah produk yang berbeda, walaupun diberi merek yang sama yaitu Plesk. Dengan aplikasi control panel hosting, seseorang dapat mengelola akun hostingnya dengan lebih mudah yaitu melalui browser. Termasuk di antaranya mengunggah berkas-berkas ke akun hosting, menginstal aplikasi web, membuat akun surel, dan masih banyak lagi.

Perbedaan antara kedua Control Panel yang sering digunakan di Indonesia, yaitu cPanel & Plesk. Masih banyak user di Indonesia yang masih belum tahu perbedaan antara keduanya itu.

Sebelumnya, buat Cloud Users yang belum tahu apa itu cPanel & Plesk, berikut ini pengertian cPanel & Plesk.

cPanel/Plesk merupakan sebuah control panel web hosting yang memberikan tampilan grafis dan peralatan automasi yang dibuat untuk memudahkan proses hosting di sebuah situs web. cPanel/Plesk dibuat untuk memudahkan pengguna atau pengelola hosting atau situs web dalam membuat, menginstall, dan memanajemen database, aplikasi, ataupun script. Lalu Control panel itu untuk apa? Control Panel digunakan untuk mengelola website yang ada di dalam server atau hosting. Dengan control panel, kita bisa melihat statistik, menginstall ataupun menghapus website, mengelola email, dan masih banyak lagi, yang pastinya berhubungan dengan pengelolaan website yang dimiliki. Berikut ini beberapa perbedaan dan kelebihan cPanel dengan Plesk:

User Interface / Antarmuka Pengguna. Secara tampilan, plesk memiliki tampilan yang bersih dan terpusat, sehingga lebih mudah digunakan dan memiliki fungsi dan tools yang terpusat, seperti pada instalasi aplikasi.

Pada cPanel, untuk yang tidak terbiasa akan merasa sulit dalam manage cPanel. Akan tetapi, jika sudah terbiasa maka cPanel akan mudah digunakan. Tools dan Installer, secara tools, Plesk dan cPanel sama-sama memiliki banyak sekali tools yang berfungsi untuk melihat statistik, mengelola email, database, dan yang lainnya. Untuk installer, Plesk memiliki fitur otomatis install aplikasi atau CMS yang sudah diberikan secara terpusat. Berbeda dengan cPanel, yang biasanya menggunakan softaculous untuk otomatis install aplikasi atau CMS dan harus memiliki softaculous license untuk mengaktifkannya. Namun biasanya, penyedia memberikan cPanel yang sudah include dengan softaculous. Jadi tidak lagi memerlukan softaculous license untuk menggunakan otomatis install aplikasi atau CMS di cPanel.

Akses User Plesk memiliki satu login untuk administrator dan pengguna. Jadi dengan Plesk Control Panel, kemudahan membuat e-mail dan subdomain akan lebih terasa. Panel Admin di Plesk terlihat lebih professional. Pada cPanel, terdapat Web Host Manager (WHM) yang dapat mempermudah dalam pembuatan user dan memisahkan login dan hak akses dari masing-masing user. OS Support Plesk dapat dijalankan di Sistem Operasi Linux ataupun Windows. Sedangkan cPanel, hanya berjalan pada CentOS, RedHat dan CloudLinux dan tidak bisa dijalankan pada OS Windows. Menghapus Panel Menghapus

cPanel tidak memungkinkan. Hal ini membutuhkan memformat server dan menginstal ulang sistem operasi. Plesk dapat dilakukan penghapusan dengan menjalankan script uninstall yang tersedia untuk Plesk tapi itu tidak dianjurkan. Migrasi

Kedua control panel ini, cPanel dan Plesk Control Panel memiliki kemudahan migrasi yang sama. Tapi tentu migrasi ini biasanya berlaku untuk control panel yang sejenis. Missal : cPanel ke cPanel dan atau Plesk ke Plesk. Migrasi antar control panel juga mungkin terjadi, tapi tentu tidak semudah migrasi dengan control panel yang sama. Permintaan Dukungan Plesk memiliki dukungan formulir permintaan yang cukup rinci yang membutuhkan informasi yang lebih luas termasuk produk, versi produk, versi OS, lingkungan kerja dan masalah.

Dalam cPanel terdapat akses ke layanan dukungan dari host Web Anda melalui email dengan cara memasukkan nama, alamat email Anda dan masalah. [6]

## **2.7 Maria DB**

Pada plesk database menggunakan Maria DB. Sedangkan Sql diimport dari localhost. MariaDB adalah sistem manajemen database relasional yang dikembangkan dari MySQL. MariaDB dikembangkan oleh komunitas pengembang yang sebelumnya berkontribusi untuk database MySQL. MariaDB tetap mempertahankan kompatibilitas dan API layaknya MySQL dulu. Jika di MySQL ada InnoDB maka di MariaDB ada XtraDB yang menjadi mesin penyimpanan baru. Adapun Aria digunakan untuk transaksi database transaksional maupun non-transaksional.

Pengembangan MariaDB sekarang dipimpin oleh Michael "Monty" Widenius, salah satu founder MySQL AB dan Monty ProgramAB. Setelah MySQL diakuisisi, Michael membangun sistem manajemen database baru dengan nama MariaDB. Penamaan MariaDB menggunakan salah satu anaknya Maria. Tak beda jauh dengan MySQL, MySQL juga dinamai dengan salah satu nama anaknya yaitu My. Untuk awal mula penomoran versi, MariaDB mengikuti skema penomoran MySQL yakni 5.5. Setelah versi 5.5, pengembang MariaDB memutuskan untuk 'lompat jauh' dengan memberi versi terbaru mereka dengan penomoran 10. Tidak hanya penomoran versinya saja, fitur-fitur major pun dibangun dalam MariaDB. Saat ini versi terbaru MariaDB yang stabil adalah MariaDB 10.1.

API dan Protokol MariaDB juga kompatibel dengan apa yang ada pada MySQL. Namun tidak hanya itu, fitur untuk dukungan native operasi non-blocking dan pelaporan progress juga ditambah. Artinya semua connector, library dan aplikasi yang bekerja pada MySQL, dapat bekerja pada MariaDB. Fedora juga mengambil langkah cepat dengan mengganti MySQL dengan MariaDB sebagai database bawaannya. Fedora telah memasang MariaDB sejak Fedora 19 dengan keyakinan bawah MySQL akan semakin menjadi produk yang tertutup dan komersial setelah diakuisis oleh Oracle. [7]

## **2.8 PHP**

Menurut No Name (2011:3), PHP (Hypertext Preprocessor) adalah bahasa pemrograman berbasis web yang memiliki kemampuan untuk memproses dan mengolah data secara dinamis. PHP dapat dikatakan sebagai sebuah server-side embedded script language, artinya semua sintaks dan perintah program yang anda tulis akan sepenuhnya dijalankan oleh server, tetapi dapat disertakan pada halaman HTML biasa. Pada umumnya, semua aplikasi yang dibangun menggunakan PHP akan memberikan hasil pada web browser, tetapi prosesnya secara keseluruhan dijalankan pada server. Kelebihan-kelebihan PHP : PHP bersifat open source. PHP adalah aplikasi bahasa web yang diperoleh secara gratis.

1. PHP mudah dipelajari dibandingkan produk lain yang mempunyai fungsi yang sama, seperti : Java Server Page atau C-based CGI. PHP mempunyai sintaks yang sangat mudah dan user-friendly.
2. PHP bersifat embedded. Penulisan script PHP dan HTML menyatu sehingga mudah dalam pembuatannya.
3. PHP dapat dijalankan di banyak Platform, seperti : UNIX dan WINDOWS.
4. PHP meningkatkan kecepatan dari proses script.15
5. PHP mempunyai fleksibilitas yang tinggi, menyamai high level programming language seperti bahasa C.
6. Life Cycle yang singkat, sehingga PHP selalu up to date mengikuti perkembangan teknologi internet. [8]

## **2.9 PhpMyAdmin**

phpMyAdmin adalah sebuah software gratis berbasis scripting language PHP yang bertujuan untuk memudahkan kita mengelola

database MySQL. Tanpa phpMyAdmin, Anda perlu menggunakan terminal untuk mengelola database anda. Sementara, dengan phpMyAdmin Anda tidak perlu susah lagi mengelola database MySQL Anda karena phpMyAdmin memiliki user interface grafis. phpMyAdmin mendukung banyak operasional MySQL, MariaDB, dan Drizzle sehingga Anda bisa menggunakannya untuk mengelola database, columns, tables, indexes, users, dan masih banyak lagi. Di saat yang sama, Anda juga bisa meng-execute SQL statement apapun secara langsung. Sampai saat ini, phpMyAdmin sudah diterjemahkan ke 72 bahasa. PhpMyAdmin sendiri sudah berumur 15 tahun dan memenangkan beberapa penghargaan. Seperti yang sudah disebutkan di atas, Anda juga tidak perlu khawatir karena phpMyAdmin mendukung fitur-fitur MySQL seperti MySQL users dan privileges management, stored procedures and triggers management, maintenance server, tables, dan databases, mengeksekusi, mengedit, dan meng-bookmark statement SQL apapun, dan masih banyak lagi. Berikut adalah beberapa fitur phpMyAdmin yang membuat software satu ini banyak digunakan: phpMyAdmin memiliki interface yang user-friendly dan intuitive yang mudah dipelajari. Dengan begitu, orang-orang dapat mempelajari dan membiasakan diri dengan fitur-fiturnya dengan mudah dan cepat. phpMyAdmin memperbolehkan user memanfaatkan kebanyakan fungsi MySQL, termasuk mengelola database, menjalankan queries MySQL, mengeksekusi statement MySQL, mengelola user dan permission dan masih banyak lagi. Anda juga bisa mengimport dan mengexport data dari dan ke berbagai format. Dengan pre-defined fuctions, Anda bisa mengubah format data-data yang sudah tersimpan phpMyAdmin bisa membuat graphics database dalam bentuk PDF, bersamaan dengan beberapa queries yang kompleks dibantu dengan query-by-example. Melakukan pencarian pada database. [9]

## **2.10 MySQL**

MySQL merupakan database server opens source yang cukup popular keberadaanya. Dengan berbagi keunggulan yang dimiliki, membuat sofware database ini banyak digunakan oleh para praktisi untuk membangun suatu project. Wahana Komputer (2010:02).

Menurut Kristanto (2010:12), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang multithread dan multi-user. MySQL adalah Relational Database Management

System (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap Orang bebas untuk menggunakan MySQL.

Berdasarkan definisi diatas dapat disimpulkan bahwa MySQL adalah database server open source atau sistem manajemen database (DBMS) yang digunakan untuk membangun sebuah project dengan menggunakan bahasa pemrograman.

Menurut Puspitasari (2011:19), MySQL memiliki beberapa kelebihan, antara lain :16

1. Multiuser. MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
2. Performance tuning. MySQL memiliki kecepatan bagus dalam menagani query sederhana, yaitu dapat memproses lebih banyak SQL per satuan waktu.
3. Jenis Kolom. MySQL memiliki tipe kolom yang kompleks, seperti signed/unsigned integer, float, double, char, text, date, timestamp, dan masih banyak lagi.
4. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh dan mendukung perintah Select dan Where dalam perintah query.
5. Keamanan. MySQL memiliki keamanan yang bagus karena beberapa lapisan keamanan seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang lengkap serta sandi terenkripsi.
6. Skalabilitas dan pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu, batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
7. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protocol TCP/IP. Unix soket (UNIX), atau Named Pipes(NT).
8. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa.<sup>17</sup>
9. Antar Muka. MySQL memiliki interface terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

10. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS X Server, Slolaris, Amiga dan masih banyak lagi.
11. Open Source. MySQL didistribusikan secara open source, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
12. Klien dan Peralatan. MySQL dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.
13. Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya, misalnya Oracle. [10]

## 2.11 CSS (*Cascading Style Sheet*)

Menurut Kurniawan (2012:58), CSS adalah teknologi standar yang digunakan oleh hampir semua browser modern saat ini. CSS tidak hanya dapat digunakan untuk memformat tampilan halaman seperti warna font, warna background, ll, tapi CSS juga dapat digunakan untuk mengubah layout/tata letak halaman.

Mudah bagi kita untuk mengetahui mana website yang menggunakan CSS dan mana yang tidak. Kita pasti pernah membuka website, tapi sesaat kemudian gagal loading dan malah menampilkan background putih yang didominasi teks hitam dan biru. Situasi ini menggambarkan bahwa CSS pada situs tersebut tidak dapat di-load dengan benar atau situs sama sekali tidak memiliki CSS. Umumnya, situs seperti ini hanya menerapkan HTML

Singkatnya, dengan CSS, Anda tidak perlu mendeskripsikan tampilan dari masing-masing elemen secara berulang-ulang. Anda tidak membuang-buang waktu, kode yang digunakan pun lebih singkat, dan error dapat diminimalisir.

Pada buku ini membahas Referensi Property CSS ini kita akan membahas cara penggunaan berbagai property CSS, serta tips dan trik untuk menggunakan property tersebut secara efisien. Diharapkan tutorial ini bisa menjadi bahan referensi untuk mempelajari CSS, termasuk CSS3. CSS BOX, Salah satu konsep dasar yang wajib dipahami di dalam CSS adalah CSS Box Model. Dalam tutorial yang ada di dalam buku ini kita akan lebih dalam apa itu CSS Box Model.

[11]

## 2.12 Model Proses

### 2.12.1 Use Case Diagram

Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan use case. Berikut simbol-simbol use case

Tabel 2.1 Tabel *Use Case Diagram*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
3		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>Sistem</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

1. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan digitalisasi yang akan dibuat diluar digitalisasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang.
2. Use case merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. Sehingga akses setiap actor pada unit diketahui dengan baik oleh pembaca buku ataupun pengguna web.

### **2.12.2 Class Diagram**

Menurut Rosa dan Shalahuddin (2014:141), Diagram kelas atau class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki. Operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut ini :

1. Kelas main  
Kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.
2. Kelas yang menangani tampilan sistem  
Kelas yang mendefinisikan dan mengatur tampilan ke pemakai
3. Kelas yang diambil dari pendahuluan use case Kelas yang menangani fungsi-fungsi yang harus ada di ambil dari pendefinisian use case.
4. Kelas yang diambil dari pendefinisian data Kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data. Jenis-jenis kelas diatas juga dapat digabungkan satu sama lain sesuai dengan pertimbangan yang dianggap baik<sup>28</sup> asalkan fungsi-fungsi yang sebaiknya ada pada struktur kelas tetap ada.

### 2.12.3 Activity Diagram

Menurut Rosa dan Shalahuddin (2014: 161), Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal – hal berikut :

1. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
2. Urutan atau pengelompokkan tampilan dari sistem / user interface dimana setiap aktivitas dianggap memiliki sebuah rancangan antar muka tampilan.
3. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.
4. Rancangan menu ditampilkan pada perangkat.2 [12]

Tabel 2.2 Tabel Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		<i>Initial Node</i>	Bagaimana objek dibentuk atau diawali.
4		<i>Activity Final Node</i>	Bagaimana objek dibentuk dan dihancurkan
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

### 2.13 Bootstrap

Bootstrap, framework paling populer di dunia. Fungsi Bootstrap untuk mengembangkan website dengan desain yang responsive dan

mobile-first. Hampir setiap website yang tersebar di internet menggunakan Bootstrap sebagai komponennya. Ini membuktikan bahwa Bootstrap dipercaya dan diterima oleh banyak pengembang.

Salah satunya Twitter. Twitter menyuguhkan tampilan sederhana bernuansa modern, dan tetap nyaman dilihat. Sosial media ini menjadi awal perkembangan Bootstrap, sebelum tersebar di seluruh dunia.

Bootstrap adalah framework CSS yang dikhususkan untuk pengembangan front-end website. Framework ini mempunyai nama asli Twitter Blueprint. Ada kata ‘Twitter’ karena pada awalnya dikembangkan untuk sosial media Twitter yang sangat terkenal saat ini dengan pengguna hampir 326 juta orang. Sebelum Bootstrap muncul, sudah ada framework lain yang tersedia. Hanya saja tingkat konsistensi dalam proses pengembangan buruk. Selain itu juga butuh biaya perawatan mahal.

Berdasarkan permasalahan ini, akhirnya para pengembang menemukan Bootstrap. Berbeda dengan framework sebelumnya, Bootstrap dikenal konsisten dan lebih sederhana.

Bootstrap adalah salah satu framework yang memungkinkan developer dapat mengembangkan website dengan mudah dan cepat. Developer hanya perlu memanggil class tertentu untuk membuat tombol, panel, tabel, pesan peringatan, dan lain sebagainya.

Bootstrap terdiri dari beberapa file. File pada Bootstrap berisi kumpulan baris kode tersusun dari CSS dan JavaScript yang berbentuk class. Jadi ketika Anda menggunakan Bootstrap untuk mengembangkan website, membuat satu tombol tidak perlu menyusun beberapa baris kode karena tinggal memanggil salah satu class saja.

Bootstrap sampai dengan saat ini masih memperbaiki sistemnya. Tidak hanya berhenti di angka 3, tapi sudah masuk ke versi 4. Pada versi 4 ini ada beberapa pembaharuan lebih banyak berhubungan dengan tema. Contohnya saja perubahan tampilan styling, warna, fonts, dan lain sebagainya. Perubahan yang terjadi antara lain:

1. Global

Bootstrap menyediakan ukuran font global (default). Bootstrap 3 menggunakan ukuran font 14px, sedangkan pada Bootstrap 4 diganti dengan ukuran yang lebih besar, yaitu 16px. Jadi, ketika menggunakan Bootstrap 3 dan diperbarui menjadi 4, tulisan akan terlihat lebih besar dibandingkan dengan sebelumnya. Jadi ketika beralih dari Bootstrap 3

ke 4, Anda perlu menyesuaikan lagi ukuran pixel menjadi lebih kecil supaya sesuai dengan ukuran sebelumnya. Selain itu, font bawaan pun diganti. Bootstrap 4 menggunakan font fallback Helvetica Neue, Arial, dan Sans-Serif menggantikan Bootstrap 3 yang sebelumnya menggunakan font Helvetica Neue, Sans-Serif, Helvetica, dan Arial.

## 2. Grid

Bootstrap 4 tidak memerlukan lagi col- pada setiap class grid. Ini berbeda dengan Bootstrap 3 yang sebelumnya menggunakan col- di depan, misalnya ketika menggunakan class “col-md-offset-3” untuk membatasi kolom. Bootstrap 4 menambah satu sistem grid menjadi 5, yaitu col-sm, col-md, col-lg, col-xl, dan col-\*. berbeda dengan Bootstrap 3 yang hanya menggunakan 4 sistem grid. Bootstrap 4 menghilangkan col-xs dan diwakilkan oleh col-sm untuk perangkat super kecil (extra small) atau menggunakan col-\* untuk mendefinisikan “semua perangkat” tanpa harus menentukannya. Jadi kesimpulannya, penggunaan baris kode Bootstrap 4 lebih ringkas dibandingkan dengan Bootstrap 3.

## 3. Tabel

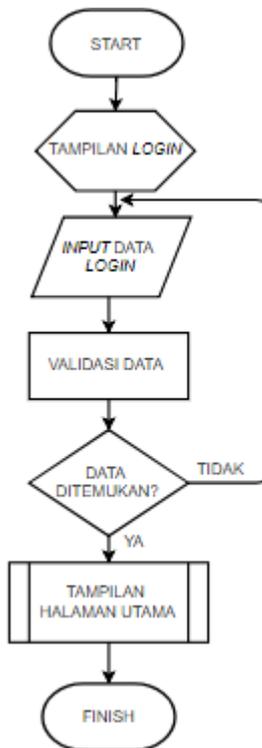
Bootstrap 4 mendukung gaya untuk header atau judul tabel –yang sebelumnya di Bootstrap 3 tidak didukung. Selain mendukung gaya header, Bootstrap 4 juga dapat memberikan warna tabel header menjadi gelap(.thead-dark) atau terang (.thead-light). Di Bootstrap 4 untuk membuat table responsive langsung di dalam tag <table>. Berbeda dengan Bootstrap 3 yang perlu menambahkan class .table-responsive di dalam tag <div>. Selain itu, di Bootstrap 4 bisa menentukan warna tabel menjadi gelap dengan menggunakan .table-dark.

**-----Halaman ini sengaja dikosongkan-----**

### BAB III

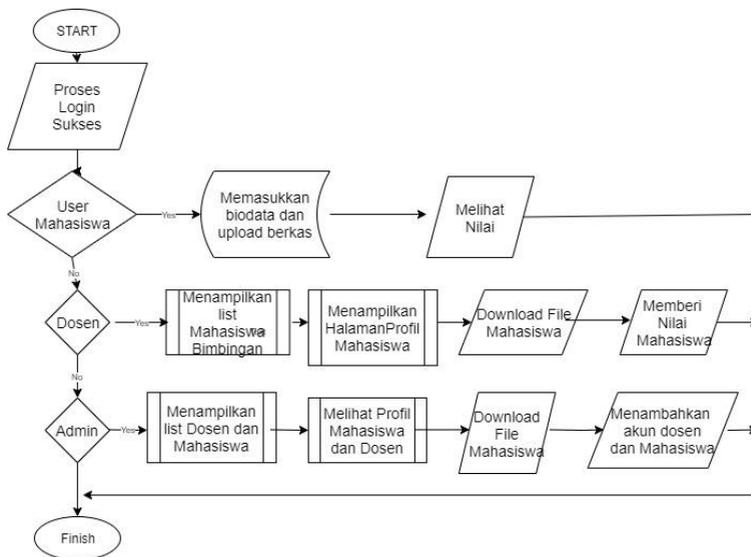
## PERENCANAAN DAN IMPLEMENTASI

Digitalisasi Magang Departemen Elektro Otomasi Berbasis Web Menggunakan *Framework* Laravel yang dirancang pada Proyek Akhir ini bertujuan untuk mempermudah proses pengumpulan dan penilaian pada Mata Kuliah Magang ini. Selain itu dengan pada Digitalisasi ini disertai *Database* yang berguna sebagai tempat penyimpanan data yang ada sehingga memudahkan pencarian *file* jika dibutuhkan.



Gambar 3.1 Diagram Alur Login

Gambar 3.1 tersebut menjelaskan mengenai proses *login* pada tampilan awal di *website*. Pada halaman ini di sajikan berupa *form login* yang mengharuskan kita untuk mengisi data *login* berupa *username* dan *password* yang dimasukkan sesuai data yang telah di simpan. Jika data sesuai dengan *database*, maka proses *login* berhasil dan akan masuk ke halaman selanjutnya, ialah halaman utama. Namun, jika *username* dan *password* tidak sesuai dengan data yang telah tersimpan, maka tidak bisa masuk ke dalam halaman utama. *Form login* ini berguna untuk sebagai sistem keamanan dari *website* ini, jadi tidak semua orang dapat mengakses *website* ini dengan semena-mena.



Gambar 3.2 Diagram Alur Web Berdasarkan User

Pada Gambar 3.2, menjelaskan mengenai *flowchart* pada menu utama. *Flowchart* ini diawali setelah proses *login* berhasil dilakukan. Pada maka setiap *user* yang login akan mendapatkan fitur yang berbeda beda. Agar setiap *user* memiliki Batasan untuk mengakses suatu halaman.

Studi literatur dilakukan dengan mengumpulkan data-data yang dibutuhkan sistem. Data yang dibutuhkan sistem diperoleh dari hasil

diskusi dengan Dosen Pembimbing dan Dosen Penanggung Jawab mata kuliah Magang.

Bab ini membahas perancangan dan realisasi dari perangkat keras dan perangkat lunak Perancangan sistem pada Proyek Akhir ini fokus pada

Pada Digitalisasi ini tidak terdapat perancangan mekanik maupun perancangan elektrik. Perancangan sistem merupakan tahapan rencana pengembangan sistem ke dalam bentuk desain yang digunakan untuk memudahkan pengguna melihat rancangan sistem yang dibuat. Langkah-langkah yang digunakan untuk merancang sistem yaitu merancang *usecase diagram*. Langkah pertama dalam perancangan sistem adalah menentukan alur dari sistem yang akan dikembangkan. Perancangan sistem dijelaskan menggunakan diagram agar pengguna dapat membaca dan memahami alur proses sistem.

Perangkat lunak yang dibutuhkan dalam pembuatan tampilan *web* dari Sistem Informasi Magang Departemen ELEKTRO Otomasi Menggunakan *Framework* Laravel

- a. Xampp
- b. *Web Editor : Visual Code* dengan *Laravel 6.2*
- c. *Browser : Google Chrome* (untuk melihat hasil dari tampilan halaman HTML)

### 3.1 Studi Literatur

Studi literatur dilakukan dengan mengumpulkan data-data yang dibutuhkan sistem. Data yang dibutuhkan sistem diperoleh dari hasil diskusi dengan Dosen Pembimbing dan juga Dosen penanggung jawab magang. Data yang diperoleh digunakan untuk menganalisis alur pelaksanaan kegiatan mengetahui kinerja sistem yang sudah ada, mengetahui kelemahan sistem yang masih belum bisa ditangani, dan menemukan solusi dalam menangani kelemahan sistem.

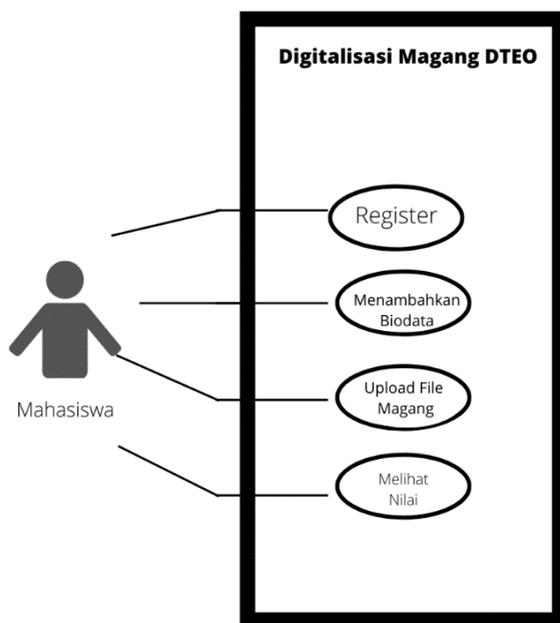
### 3.2 Diagram Usecase

Digitalisasi Magang Departemen Teknik Elektro terdapat tiga user yaitu Mahasiswa, Dosen Pembimbing, serta Dosen Penanggung Jawab Magang sebagai Admin. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada use case yaitu pendefinisian apa yang disebut aktor dan *use case*. *Use Case Diagram* adalah gambaran grafis dari beberapa atau

semua actor, use case, dan interaksi diantaranya yang memperkenalkan suatu sistem. Use case diagram tidak menjelaskan secara detail tentang penggunaan use case, tetapi hanya memberi gambaran singkat hubungan antara use case, aktor, dan sistem. Di dalam use case ini akan diketahui fungsi-fungsi apa saja yang berada pada sistem yang dibuat

a. *Usecase* Diagram Mahasiswa

*Usecase* Diagram Digitalisasi Magang Departemen Elektro Otomasi sebagai Mahasiswa ditunjukkan pada Gambar 3.3.

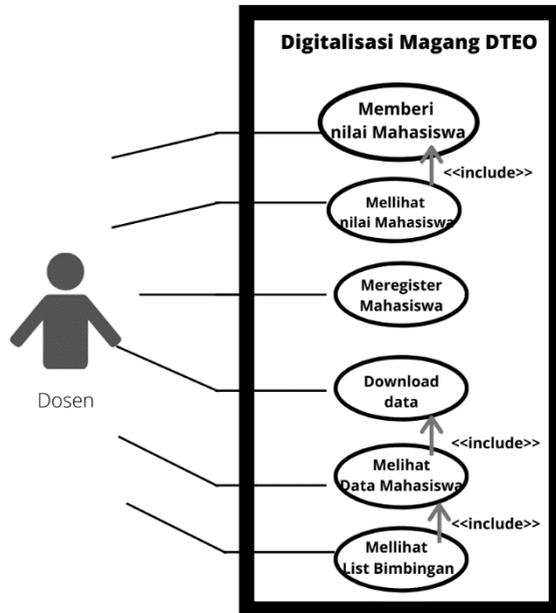


Gambar 3.3. *Usecase* Diagram Mahasiswa

Pada user Mahasiswa bisa melakukan register, memasukkan data selama magang, *upload* berkas, dan melihat nilainya. Dari diagram diatas menampilkan kegunaan setiap *user* Mahasiswa. Pada alamat edit data Menambahkan data dan *upload* dilakukan bersama.

b. *Usecase* Diagram Dosen

*Usecase* Diagram Digitalisasi Magang Departemen Elektro Otomasi sebagai Dosen ditunjukkan pada Gambar 3.4.

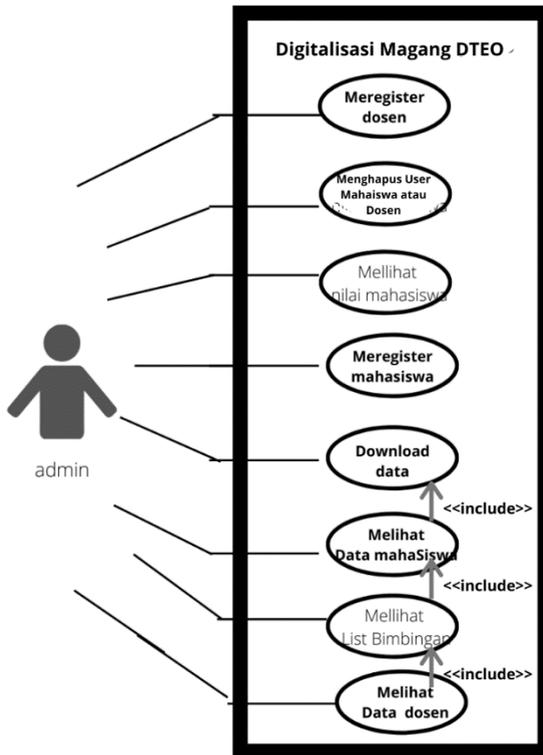


Gambar 3.4. *Usecase* Diagram Dosen

Pada user Dosen bisa melakukan melihat list siswa yang dibimbing, memasukkan nilai, melihat nilai, *download* data Mahasiswa, dan melihat data Mahasiswa

b. *Usecase* Diagram Admin

*Usecase* Diagram Digitalisasi Magang Departemen Elektro Otomasi untuk User Admin ditunjukkan pada Gambar 3.5. Dengan adanya *usecase* maka akan mengetahui perbedaan akses yang dimiliki setiap jenis *user*



Gambar 3.5. Usecase Diagram Admin

Pada user Dosen bisa melakukan melihat list siswa yang dibimbing, memasukkan nilai, melihat nilai, *download* data Mahasiswa, melihat data Dosen, Menambah dosen, Menghapus *user* dosen, dan melihat data Mahasiswa. Pada user ini memiliki akses paling banyak jika dibandingkan dengan *user* yang lain dan juga admin memiliki jumlah *user* paling sedikit

Itulah *usecase* diagram pada setiap *user* pada web ini. Dengan adanya *usecase* ini maka mempermudah dalam mengakses *web* ini. Untuk akses paling sedikit ada pada *user* Mahasiswa. Itu terjadi karena Mahasiswa hanya melakukan *upload* dan melengkapi data yang ada. Selain itu Dosen dapat memasukkan nilai.

### 3.3 Database Pada PhpMyAdmin

#### 1. Daftar Tabel Pada Database

Pada database terdapat 5 tabel utama yaitu Tabel *User* adalah tabel ini sebagai database untuk login pada web ini. Selain itu ada Tabel Mahasiswa, Tabel Mapel, Tabel Dosen, dan Tabel Mapel\_Dosen.

#### 2. User PhpMyAdmin

Tabel 3.1 Tabel User PhpMyAdmin

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
2	role	varchar(45)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
3	name	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	email	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
5	email_verified_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
6	password	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
7	remember_token	varchar(100)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
8	created_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
9	updated_at	timestamp		Ya	NULL				Ubah Hapus Lainnya

Pada gambar tabel ini adalah tabel dimanamengatur *users* untuk login. Di dalam gambar tersebut terdapat *role* yang bertujuan untuk membatasi akses ketika *login*. Selain itu juga terdapat email sebagai *username*.

#### 3. Tabel Mahasiswa

Tabel 3.2 Tabel Mahasiswa

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	bigint(20)		UNSIGNED	Tidak	Tidak ada			Ubah Hapus Lainnya
2	user_id	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
3	Nama	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
4	Angkatan	int(11)			Tidak	Tidak ada			Ubah Hapus Lainnya
5	NRP	varchar(255)	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
6	Nama_Perusahaan	text	utf8mb4_unicode_ci		Tidak	Tidak ada			Ubah Hapus Lainnya
7	Alamat_Perusahaan	text	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
8	Tanggal_Mulai	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
9	Tanggal_Selesai	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
10	Deskripsi	text	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
11	Buku1	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
12	Buku2	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
13	Proposal	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
14	Logbook	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
15	Video_Magang	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
16	avatar	varchar(255)	utf8mb4_unicode_ci	Ya	NULL				Ubah Hapus Lainnya
17	dosen_id	int(11)		Ya	NULL				Ubah Hapus Lainnya
18	created_at	timestamp		Ya	NULL				Ubah Hapus Lainnya
19	updated_at	timestamp		Ya	NULL				Ubah Hapus Lainnya

Tabel ini berfungsi untuk menyimpan data yang berkaitan dengan data yang dimasukkan oleh Mahasiswa.

#### 4. Tabel Mapel

Tabel 3.3 Tabel Mapel

#	Nama	Jenis	Penyortiran	Atribut	Tak Ternilai	Bawaan	Komentar	Ekstra	Tindakan
1	id	int(11)		Tidak	Tidak ada				Ubah Hapus Lainnya
2	nama	varchar(191)	utf8mb4_general_ci	Tidak	Tidak ada				Ubah Hapus Lainnya
3	created_at	timestamp		Tidak	CURRENT_TIMESTAMP				Ubah Hapus Lainnya
4	updated_at	datetime		Tidak	Tidak ada				Ubah Hapus Lainnya

Tabel 3.4 Tabel Jenis Mapel

Options									
				id	nama	created_at	updated_at		
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	8	Dosen Pembimbing	2019-12-11 00:25:11	0000-00-00 00:00:00	
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	9	Dosen Penguji 1	2019-12-11 00:25:11	0000-00-00 00:00:00	
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	10	Dosen Penguji 2	2019-12-11 00:25:46	0000-00-00 00:00:00	
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	11	Dosen Penguji 3	2019-12-11 00:25:46	0000-00-00 00:00:00	
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	12	Dosen Penguji 4	2019-12-11 00:26:12	0000-00-00 00:00:00	
<input type="checkbox"/>	Ubah	Edit	Copy	Delete	13	Nilai dari Perusahaan	2019-12-11 00:26:12	0000-00-00 00:00:00	

Tabel 3.4 berguna untuk tempat penyimpanan jenis nilai yang akan diberikan. Data yang ada pada tabel ini selalu sama atau tidak perlu diubah. Isi dari tabel tersebut bisa dilihat pada Tabel 3.4.

#### 5. Tabel Dosen

Tabel 3.5 Tabel Dosen

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	nama	varchar(100)	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	3	user_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	4	telpon	varchar(15)	utf8mb4_general_ci	No	None			Change Drop More
<input type="checkbox"/>	5	alamat	text	utf8mb4_general_ci	Yes				Change Drop More
<input type="checkbox"/>	6	created_at	timestamp		No	current_timestamp()			Change Drop More
<input type="checkbox"/>	7	updated_at	datetime		No	None			Change Drop More

Tabel kali ini adalah tabel dimana data-data *user* Dosen Pembimbing dimasukkan. Pada tabel ini ditampilkan pada profil Dosen Pembimbing.

#### 6. Tabel Relasi Mahasiswa dan Mapel

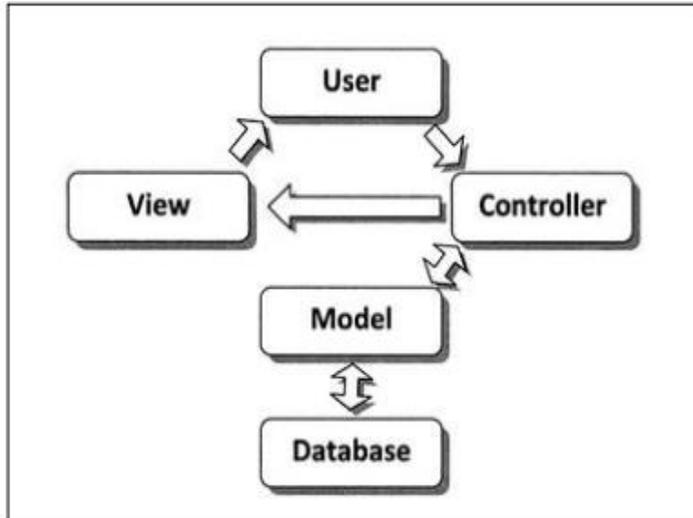
Tabel 3.6 Tabel Relasi Mahasiswa dan Mapel

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT	Change Drop More
<input type="checkbox"/>	2	siswa_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	3	mapel_id	int(11)		No	None			Change Drop More
<input type="checkbox"/>	4	nilai	int(11)		No	None			Change Drop More
<input type="checkbox"/>	5	created_at	timestamp		No	current_timestamp()			Change Drop More
<input type="checkbox"/>	6	updated_at	datetime		No	None			Change Drop More

Ini adalah tabel hasil relasi antara tabel Mahasiswa dan Tabel Mapel. Selain itu nilai yang diberikan akan masuk ke tabel ini

### 3.4 Pembuatan Web dengan Framework Laravel

Menurut (Pribadi, 2016), Laravel adalah sebuah framework PHP yang dirilis di bawah lisensi MIT, dibangun dengan konsep MVC (model view controller). Berikut ini adalah proses pembuatan program dari setiap fungsi yang dibuat.



Gambar 3.6 Model Hubungan MVC

1. View adalah komponen dari suatu MVC yang bertugas menampilkan apa yang harus ditampilkan ke pengunjung website. Isinya dapat berupa form, table, gambar, animasi ataupun lainnya yang boleh dilihat user (pengunjung).
2. Model adalah komponen MVC yang bertugas mengambil data dari database dan juga memasukkan data ke database. Isi umumnya berupa perintah SQL. Hasilnya dikirimkan ke Controller.
3. Controller adalah komponen MVC yang bertugas mengirim perintah ke model untuk mendapatkan data yang diinginkan. Controller tidak mengetahui bagaimana data tersebut diambil

dari database, karena Controller tidak berisi kode perintah SQL. Karena itu adalah tugas model, Controller mengolah data dari inputan user dan data dari model kemudian data olahan tersebut dikirimkan ke view untuk ditampilkan sesuai aturan Controller. Controller merupakan penghubung antara model dan view

### 3.4.1 Routes

*Routes* merupakan bagian yang sangat penting ketika berurusan dengan aplikasi terutama yang berbasis web. Jika menggunakan framework dalam mengerjakan sebuah web, router berfungsi menangani *request* aplikasi kemudian mengarahkan aplikasi untuk memanggil halaman/*resource* tertentu. Munculannya sebuah halaman ketika kita mengetikkan sebuah alamat di kolom domain juga diatur disini.

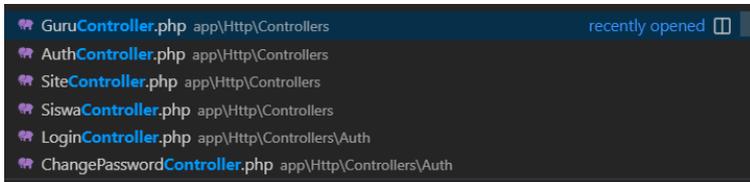
Contoh *route* pada *line* 19 di gambar 3.7. Dengan fungsi pada *route* tersebut. Jika kita menekan menu register pada halaman awal akan di *direct* ke *controller* yang bernama 'SiteController@register'.

```
routes > web.php -
10 | contains the "web" middleware group. Now create something great!
11 |
12 | */
13 |
14 | use Illuminate\Support\Facades\Route;
15 |
16 | Route::get('/', function () {
17 |     return view('welcome');
18 | });
19 | Route::get('/register', 'SiteController@register')->name('register');
20 | Route::post('/postregister', 'SiteController@postregister');
21 |
22 | Route::get('/login', 'AuthController@login')->name('login');
23 | Route::post('/postlogin', 'AuthController@postlogin');
24 | Route::get('/logout', 'AuthController@logout');
25 |
26 | Route::get('/change-password', 'Auth\ChangePasswordController@index')->name('password.change');
27 | Route::post('/change-password', 'Auth\ChangePasswordController@changepassword')->name('password.update');
28 |
29 | //download
30 | Route::get('/siswa/{id}/download/logbook', 'SiswaController@downloadlogbook');
31 | Route::get('/siswa/{id}/download/proposal', 'SiswaController@downloadproposal');
32 | Route::get('/siswa/{id}/download/buku1', 'SiswaController@downloadbuku1');
33 | Route::get('/siswa/{id}/download/buku2', 'SiswaController@downloadbuku2');
34 | Route::get('/siswa/{id}/download/video_magang', 'SiswaController@downloadVideo_Magang');
35 |
```

Gambar 3.7 Listing Program route/web.php

### 3.4.2 Controller

*Controller* mengambil permintaan, memarsing permintaan, menginisialisasi, memanggil model dan kemudian mengambil responnya, dan mengirimkannya ke *View*. Controller sendiri adalah tempat mengolah fungsi yang diminta oleh *route* sebelum ditampilkan ke *view*. Pada web ini *controller* yang ada pada gambar 3.7



Gambar 3.7 Jenis Jenis Controller

a. *Auth dan Login*

Pada *controller* ini mengatur, menganalisa dan memasukkan logika tentang masalah *authentication* saat *login*. *Controller* ini mengatur tentang *login,logout* dan *view* saat *post login*. Dengan *auth* juga kita bisa membatasi akses setiap jenis pengguna atau *user*. Pada *controller* ini juga terdapat halaman yang pertama diakses setelah login yaitu halaman *Dashboard*. Sehingga halaman ini bisa diakses semua jenis pengguna.

```

app > Http > Controllers > AuthController.php > PHP Intelephense > AuthController > login
1 <?php
2
3 namespace App\Http\Controllers;
4 use Auth;
5 use Illuminate\Http\Request;
6
7 class AuthController extends Controller
8 {
9     public function login()
10     {
11         return view('auths.login');
12     }
13
14     public function postlogin(Request $request)
15     {
16         if(Auth::attempt($request->only('email','password')))
17         {
18             return redirect('/dashboard');
19         }
20         return redirect('/login');
21     }
22     public function logout()
23     {
24         Auth::logout();
25         return redirect('/login');
26     }
  
```

Gambar 3.8 Listing Program AuthController.php

b. *SiteController.php*

Pada halaman *home* dan *register* mahasiswa, halaman tersebut diatur pada *controller ini*. mengambil permintaan, memarsing permintaan, menginisialisasi, memanggil model dan kemudian mengambil responnya. Site Controller bisa dilihat pada Gambar 3.9. Dalam *controller* ini terdapat juga fungsi *post register*. Dimana pada *postregister* mahasiswa yang baru mendaftar ada.

```

app > Http > Controllers > SiteController.php > PHP IntelliSense > SiteController > postregist
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use Illuminate\Support\Str;
7
8  class SiteController extends Controller
9  {
10     public function register()
11     {
12         return view('sites.register');
13     }
14     public function postregister(Request $request)
15     {
16         $user = new \App\User;
17         $user->role = 'siswa';
18         $user->name = $request->Nama;
19         $user->email = $request->email;
20         $user->password = bcrypt($request->password);
21         $user->remember_token = Str::random(60);
22         $user->save();
23
24         //Insert ke table siswa
25         $request->request->add(['user_id'=> $user->id ]);
26         $siswa = \App\Siswa::create($request->all());

```

Gambar 3.10 SiteController.php

c. ChangePasswordController.php

Sesuai Namanya, pada *controller* ini hanya mengatur tentang mengganti Password pada user yang telah *login*. Pada *controller* ini memiliki dua fungsi yaitu memverifikasi user yang akan mengganti password dengan password yang ada dan juga fungsi dimana data masuk ke *database*.

```

app > Http > Controllers > Auth > ChangePasswordController.php > PHP IntelliSense > ChangePasswordController > changepassword
1  <?php
2
3  namespace App\Http\Controllers\Auth;
4
5  use App\User;
6  use App\Http\Controllers\Controller;
7  use Illuminate\Http\Request;
8  use Illuminate\Support\Facades\Auth;
9  use Illuminate\Support\Facades\Hash;
10
11 class ChangePasswordController extends Controller
12 {
13     public function index()
14     {
15         return view('auth.password.change');
16     }
17     public function changepassword(Request $request)
18     {
19         $this->validate($request,[
20             'oldpassword' => 'required',
21             'password' => 'required|confirmed'
22         ]);
23
24         $hashedPassword = Auth::user()->password;
25         if(Hash::check($request->oldpassword,$hashedPassword)){
26             $user = User::findOrFail($id);

```

Gambar 3.11 ChangePasswordController.php

#### d. DosenController.php

```
app > Http > Controllers > GuruController.php > PHP Intelephense > GuruController > create > @ Request
1 <?php
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Guru;
7 use App\View;
8 use Illuminate\Support\Str;
9
10 class GuruController extends Controller
11 {
12     public function profile($id)
13     {
14         $guru = Guru::find($id);
15         return view('guru.profile', ['guru' => $guru]);
16     }
17     public function index(Request $request)
18     {
19         $data_guru = Guru::all();
20         return view('guru.index', ['data_guru' => $data_guru]);
21     }
22     public function create(Request $request)
23     {
24         //dd($request->all());
25     }
26     //Insert ke table users

```

Gambar 3.12 Listing Program DosenController.php

Pada *controller* yang mengatur tampilan yang berkaitan dengan Dosen diatur dengan fungsi yang dimasukkan di *controller* ini. Fungsi yang terdapat di *controller* ini adalah *profile*, *index*, *create*, *delete* dan profil pribadi. Fungsi *profile* adalah *controller* yang berisi tentang halaman profil yang menampilkan data dari Dosen. *Index* fungsi yang mengatur halaman yang bertujuan untuk mengetahui jumlah Dosen dan halaman *list* Dosen. Fungsi *create* dan *delete* adalah fungsi yang berguna untuk menambahkan dan menghapus *user* Dosen. Aktifitas ini hanya bisa diakses oleh Admin.

#### e. SiswaController.php

```
app > Http > Controllers > SiswaController.php > ...
1
2
3 namespace App\Http\Controllers;
4
5 use Illuminate\Http\Request;
6 use App\Siswa;
7 use Illuminate\Support\Str;
8 use Illuminate\Support\Facades\Storage;
9 use DataTables;
10 use App\Exports\SiswaExport;
11 use Maatwebsite\Excel\Facades\Excel;
12
13 class SiswaController extends Controller
14 {
15     public function index(Request $request)
16     {
17         if($request->has('cari')){
18             $data_siswa = Siswa::where('nama','LIKE','%'.$request->scari.'%')->paginate();
19         }else{
20             $data_siswa = Siswa::all();
21         }
22         return view('siswa.index', ['data_siswa' => $data_siswa]);
23     }
24     public function create(Request $request)
25     {

```

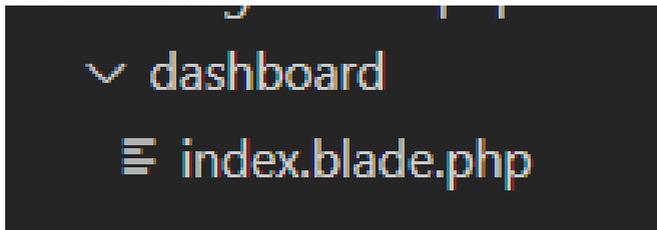
Gambar 3.13 Listing Program SiswaController.php

Fungsi yang terdapat pada *controller* ini adalah *index*, *create*, *edit*, *update*, *delete*, *profile*, tambah nilai, hapus nilai, *download* dan *export*. *Index* dipanggil oleh *view* ketika kita akan melihat halaman dimana *list* Mahasiswa ada. Fungsi *create* berguna untuk membuat user siswa yang hanya bisa dilakukan oleh admin. *Delete* berguna menghapus *user* Mahasiswa beserta data yang berkaitan dengannya. Selain itu ada tambah dan hapus nilai dimana tempat Dosen Pembimbing memasukkan nilai yang ada pada data diri Mahasiswa. Fungsi *Download* bertujuan agar data siswa bisa di *download*. *Export* sendiri digunakan mengexport dan mendownload file dan dijadikan *ffile* dalam *microfost excel*.

### 3.4.3 View

*View* pada Laravel menggunakan blade. Untuk bisa menggunakan blade di proyek laravel Anda, hanya ada satu hal yang harus dilakukan. Ganti nama file view Anda, misalnya dari *sample.php* menjadi *sample.blade.php*. Selanjutnya, laravel akan menerjemahkan tag-tag blade menjadi tag-tag PHP yang bersesuaian. Tidak ada proses aneh-aneh didalamnya. Kode `@if(true)` akan diterjemahkan menjadi `<?php if(true): ?>`. Dengan adanya proses ini kita dimudahkan dan fokus untuk membuat suatu tampilan.

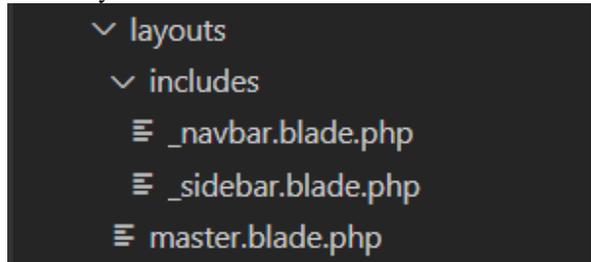
#### a. View Pada Dashboard



Gambar 3.14 *File View* pada *Dashboard*

Pada *view* di *Dashboard* terdapat satu jenis file blade, itu dikarenakan pada halaman dashboard hanya membutuhkan satu halaman saja yaitu untuk `magangdteo.com/index/dashboard`. Sehingga hanya membuat satu *file* blade saja. Pada halaman *dashboard* sendiri bisa diakses oleh semua *user*. Sehingga halaman ini adalah halaman pertama yang dikunjungi setelah login.

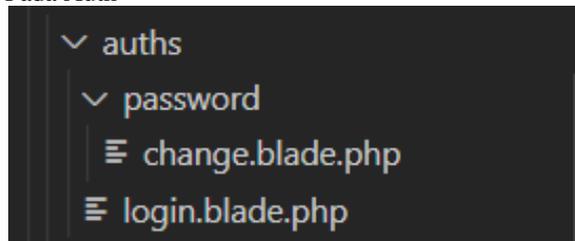
## b. View Pada Layouts



Gambar 3.15 *File View* pada Layouts

Pada *view layouts* terdapat *file blade* yang berguna untuk membuat sidebar dan navbar pada web ini. Dengan adanya *file* ini jika ingin menampilkan sidebar dan navbar di suatu halaman maka kita hanya perlu memanggil fungsi yang ada pada *file* tersebut. Navbar merupakan *header* pada halaman *web* ini. Pada navbar sendiri terdapat menu pilihan ganti *password* atau *logout*. *Sidebar* berguna untuk mempercantik tampilan dan menampilkan tombol ke halaman *dashboard*, profil, sampai daftar siswa. Master sendiri berguna sebagai tempat dari tampilan tambahan jika diinginkan bisa dipanggil atau digunakan di halaman lain.

## c. View Pada Auth

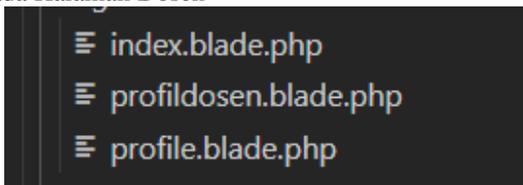


Gambar 3.16 *File View* Pada Auths

Pada *auths* sendiri terdapat halaman yang berguna untuk autentifikasi. Pada *website* ini sendiri terdapat dua halaman autentifikasi. Yaitu halaman untuk login user maupun halaman untuk ganti *password*. Pada sistem ini terdapat tiga jenis user. Dengan adanya

login yang berguna untuk autentifikasi ini kita bisa memisahkan menu setiap *user*.

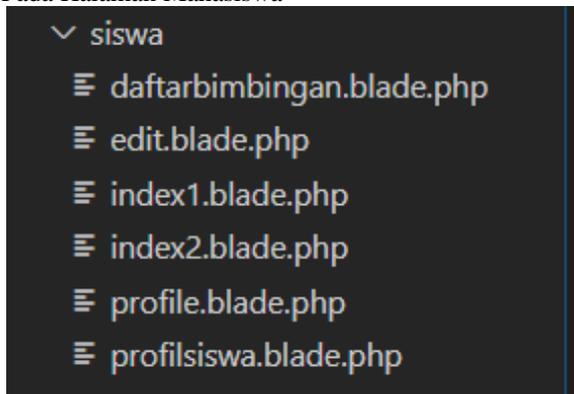
d. View Pada Halaman Dosen



Gambar 3.17 File Pada Antarmuka *User* Dosen

User dosen dapat mengakses beberapa halaman. Untuk *file* yang ada pada dokumen `view/dose` ada tiga seperti Gambar 3.17. Untuk *file* `index.blade.php` menampilkan *list* dosen. Untuk halaman ini hanya bisa diakses oleh Admin. Selain itu `profile.blade.php` adalah halaman yang menampilkan profil Dosen, untuk halaman ini profil ini hanya diakses oleh *user* sebagai Admin. Sedangkan `profildosen.blade.php` adalah halaman profil pribadi yang hanya bisa diakses Dosen tersebut.

e. View Pada Halaman Mahasiswa

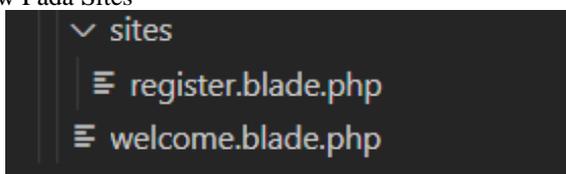


Gambar 3.18 View Pada Halama Mahasiswa

`Daftarbimbingan.blade.php` adalah *file* yang berisi halama list bimbingan dari seorang Dosen. Dengan mengakses halaman ini maka Dosen bisa masuk ke profil Mahasiswa, dengan masuk ke profil Mahasiswa. Dosen dapat melihat Biodata, File, dan memberikan nilai

pada Mahasiswa tersebut. Pada *file* edit.blade.php berisi *listing program* halaman bagi Mahasiswa yang ingin memnambah maupun mengedit data Mahasiswa tersebut. Untuk index1.blade.php dan index2.blade.php berisi *list* daftar Mahasiswa. Perbedaan dari kedua program tersebut adalah. Jika index 1 berguna untuk *list* data Mahasiswa meliputi nama, NRP, angkatan sampai dengan tanggal magang. Index 2 berguna untuk melihat *file* yang di-upload Mahasiswa secara sekilas. Halaman ini bertujuan untuk menyortir jika ada Mahasiswayang belum upload. Profile.blade.php berguna menampilkan halaman profil siswa pada *user* Admin dan Dosen. Dengan adanya halaman ini Admin atau Dosen bisa melakukan *download* berkas yang telah di-*upload* Mahasiswa.

f. View Pada Sites



Gambar 3.19 View Pada Site

Sites ini terdapat dua *file* yang berguna untuk menampilkan *landing page* maupun halaman *register* Mahasiswa.

**-----Halaman ini sengaja dikosongkan-----**

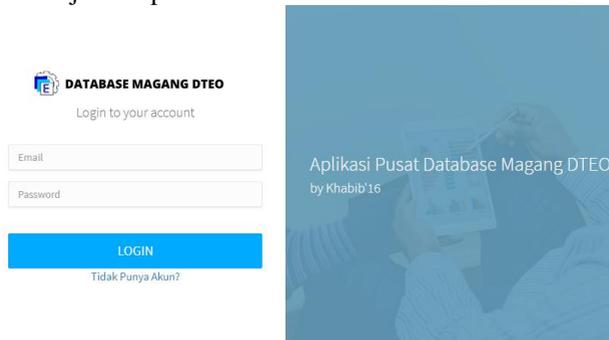
## BAB IV PENGUJIAN DAN ANALISIS

### 4.1 Rancangan Antarmuka Pada Localhost dan Web

Antarmuka merupakan tampilan sistem yang berfungsi sebagai penghubung antara sistem Magang dengan pengguna. Pada tampilan kali ini akan menampilkan antarmuka pada *localhost*. Rancangan antarmuka dari digitalisasi dibuat mengikuti perubahan dan penambahan fitur dalam digitalisasi Magang Departemen Teknik Elektro Otomasi sebelumnya. Setelah itu akan kita upload ke web dengan domain [www.magangdteo.com](http://www.magangdteo.com). Tampilan pada *website dan localhost* sama. Server yang digunakan menggunakan VPS atau *virtual private server*. Pada bab ini akan menjelaskan tentang pengujian setiap data yang ada.

#### 1. Rancangan Antarmuka Login

Halaman login pengguna diarahkan untuk memasukan username dan password untuk login sistem. Desain rancangan antarmuka login sistem ditunjukkan pada Gambar 4.1.



Gambar 4.1 Antarmuka Login

#### 2. Antarmuka Register

Halaman ini ditujukan untuk Mahasiswa yang akan membuat akun di Digitalisasi Magang Departemen Teknik Elektro Otomasi yang ditunjukkan pada Gambar 4.2.

 **DATABASE MAGANG DTEO**

Form Registrasi untuk Mahasiswa

Nama Lengkap
ID Dosen Pembimbing
Angkatan
NRP (14 Angka)
Nama Perusahaan Magang

Pastikan Data Benar Sebelum Register

email
Password

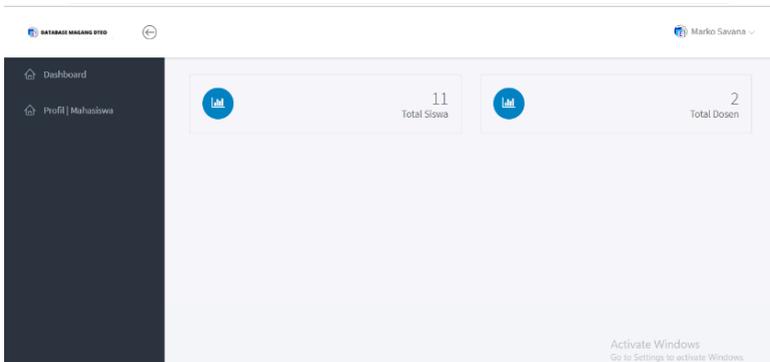
**REGISTER →**

Gambar 4.2 Antarmuka Register

Pada halaman ini Mahasiswa harus memasukkan Nama, Dosen Pembimbing, Angkatan, NRP, Nama Perusahaan Magang maupun email dan Password. Untuk bisa mengakses halaman ini bisa dengan menekan tombol register di halaman utama dari *domain website* ini. Setelah Mahasiswa berhasil melakukan Register maka *user* akan *redirect* ke halaman login. Itu dilakukan agar setelah melakukan register dapat login ke akun Mahasiswa tersebut. Data *user* akan dimasukkan ke *database*.

3. Antarmuka Mahasiswa
  - a. Antarmuka Dashboard

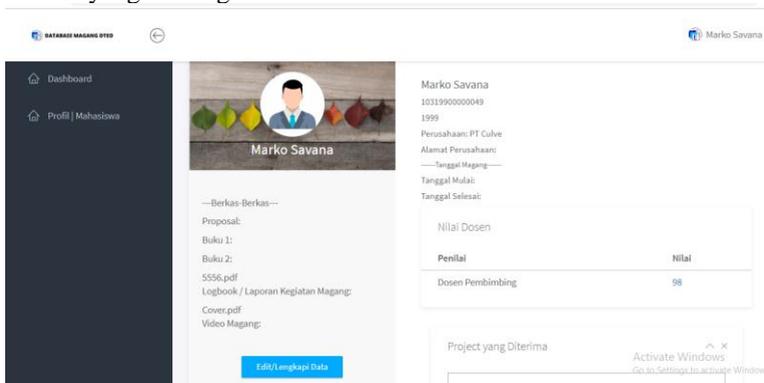
Pada Antarmuka Dashboard bisa diakses semua *user*. Pada halaman ini berisi jumlah Mahasiswa dan Dosen Pembimbing. Antarmuka *Dashboard* ditunjukkan pada gambar 4.3



Gambar 4.3 Antarmuka Dashboard

b. Antarmuka Profil

Pada Profil Mahasiswa akan menampilkan data diri Mahasiswa. Antarmuka Profil Mahasiswa dapat dilihat pada Gambar 4.4.. Halaman ini hanya bisa dikunjungi oleh Mahasiswa itu yang bersangkutan.



Gambar 4.4 Antarmuka Profil Mahasiswa

Pada Antarmuka ini menampilkan biodata Mahasiswa sampai file yang ada. Selain itu Mahasiswa juga dapat melihat nilai yang diberikan dari Dosen Pembimbing sampai Dosen Penguji

c. Antarmuka Edit Data Mahasiswa

Pada Edit Mahasiswa akan menampilkan data diri Mahasiswa. Antarmuka Profil Mahasiswa dapat dilihat pada Gambar 4.5.

The image shows a web form titled "EDIT DATA MAHASISWA". The form consists of several sections, each with a label and an input field. The labels and their corresponding values are: "Nama" with "Marko Savana"; "ID Dosen Pembimbing" with "18"; "Angkatan" with "1999"; "NRP" with "10319900000049"; "Nama Perusahaan" with "PT Culve"; and "Alamat Perusahaan" with "Alamat Perusahaan".

Gambar 4.5 Antarmuka Edit Data Mahasiswa

Pada Antarmuka ini ada form yang digunakan untuk upload biodata Mahasiswa sampai *file* yang ada. Selain itu Mahasiswa juga dapat *upload file* yang dibutuhkan. Pada halaman ini hanya bisa diakses oleh Mahasiswa. Data yang dimasukkan akan di-*input* ke database

4. Antarmuka Dosen Pembimbing

a. Antarmuka Profil Dosen

Pada Profil Mahasiswa akan menampilkan data diri Pembimbing. Antarmuka Profil Dosen Pembimbing dapat dilihat pada Gambar 4.6.



Imung Nugroho S.Pd.

ID: 17

No Telp

0867987877777

Alamat

gang makam blok e8 Keputih

[Daftar Mahasiswa yang Dibimbing](#)

Gambar 4.6 Antarmuka Profil Dosen Pembimbing.

Pada Antarmuka ini menampilkan data diri dan ada tombol untuk melihat daftar Mahasiswa bimbingan. Tujuan dari adanya fitur tersebut adalah agar Dosen Pembimbing hanya bisa Input nilai pada Mahasiswa yang dibimbing.

b. Antarmuka Daftar Mahasiswa Bimbingan

Pada Daftar Mahasiswa Bimbingan akan menampilkan list dari daftar mahasiswa bimbingan Pembimbing. Antarmuka Daftar Mahasiswa Bimbingan dapat dilihat pada Gambar 4.7

NAMA	Angkatan	NRP	NAMA PERUSAHAAN
Alfan Pratama	2000	1031160000099	PT Indocrane
khabib	2022	1111111111111111111	PTku
Khabib Nur Budiono	2016	1031160000049	TernakNesia Tekno Farm
NANAkomatsu	2016	1111111111111111111	PTku
Pratamanu	2016	111111111	PT Nippon Japan
Pratamanu	2022	111111111111114	PT Nippon Japak

Showing 1 to 6 of 6 entries

Previous 1 Next

Gambar 4.7 Antarmuka Daftar Mahasiswa Bimbingan

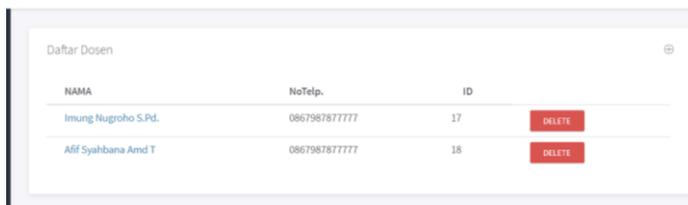
Pada Antarmuka ini menampilkan data diri tombol daftar Mahasiswa bimbingan. Hasil dari antarmuka ini adalah dengan

munculnya daftar nama Mahasiswa. Terdapat kolom *search* untuk memilah data yang diinginkan

## 5. Antarmuka Admin

### a. Antarmuka Daftar Dosen

Pada antarmuka ini menampilkan list dosen yang ada. Selain itu juga terdapat tombol yang berguna untuk menambahkan *user* Dosen Pembimbing. Selain itu juga bisa melihata profil setiap Dosen Pembimbing dengan menekan nama dari Dosen Pembimbing. Berikut ini adalah tampilan dari Antarmuka Daftar Dosen.

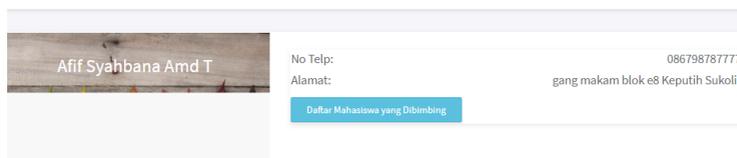


NAMA	NoTelp.	ID	
Imung Nugroho S.Pd.	0867987877777	17	DELETE
Afif Syahbana Amd T	0867987877777	18	DELETE

Gambar 4.8 Antarmuka Daftar Dosen

### b. Antarmuka Profil Dosen Pembimbing

Supaya bisa masuk ke dalam Profil Dosen Pembimbing dengan cara menekan nama yang terdaftar di dalam list. Tampilan antarmuka Profil Dosen Pembimbing ada pada Gambar 4.9. Halaman ini hanya bisa dilihat oleh *user* dengan role Admin. Untuk halaman profil dengan *user* Dosen mempunyai halaman yang berbeda.



Afif Syahbana Amd T	No Telp: 0867987877777
	Alamat: gang makam blok e8 Keputih Sukolilo
<a href="#">Daftar Mahasiswa yang Dibimbing</a>	

Gambar 4.9. Antarmuka Profil Dosen Pembimbing

c. Antarmuka Register Dosen

Pada register ini bertujuan untuk menambah *user* dengan *role* sebagai Dosen Pembimbing. Halaman ini hanya bisa diakses oleh *user* dengan *role* Admin. Data yang dimasukkan akan disimpan di *database*. Pada *database* sendiri akan tersimpan pada Tabel Dosen dan Tabel User . Data pada Tabel Dosen berguna untuk pengolahandata pada fungsi yang ditampilkan di web sedangkan pada Tabel Dosen hanyauntuk proses Login Tampilan pada antarmuka Register Dosen dapat dilihat dari Gambar 4.10

Daftarkan Dosen x

**nama**

**NO. Telp.**

**Alamat**

**Email**

**ID (masukkan angka setelah dosen terakhir di register)**

Gambar 4.10 Antarmuka Register Dosen

d. Antarmuka Daftar Mahasiswa

Pada antarmuka ini menampilkan list Mahasiswa yang ada. Selain itu juga terdapat tombol yang berguna untuk menambahkan *user* Mahasiswa. Selain itu juga bisa melihat profil setiap Dosen Pembimbing dengan menekan nama dari Mahasiswa. Berikut ini adalah tampilan dari Antarmuka Daftar Mahasiswa.

Database Magang DTEO Download Export Table As Text

Show 10 entries Search:

NAMA	Angkatan	NRP	NAMA PERUSAHAAN	AKSI
Alfan Pratama	2000	10311600000099	PT Indocrane	<a href="#">EDIT</a> <a href="#">DELETE</a>
ipan	1999	111111111111114	bri	<a href="#">EDIT</a> <a href="#">DELETE</a>
khabin	2016	11111111111111111	PTku	<a href="#">EDIT</a> <a href="#">DELETE</a>
khabin	2016	11111111111111111	eee	<a href="#">EDIT</a> <a href="#">DELETE</a>
khabin	2022	11111111111111111	PTku	<a href="#">EDIT</a> <a href="#">DELETE</a>
Khabin Nur Budiono	2016	10311600000049	TernakNesia Tekno Farm	<a href="#">EDIT</a> <a href="#">DELETE</a>
Marko Savana	1999	10319900000049	PT Culve	<a href="#">EDIT</a> <a href="#">DELETE</a>

Gambar 4.11 Antarmuka Daftar Mahasiswa

e. Antarmuka Profil Mahasiswa

Supaya bisa masuk ke dalam Profil Mahasiswa dengan cara menekan nama yang terdaftar di dalam list. Tampilan antarmuka Profil Mahasiswa ada pada Gambar 4.12



**Alfan Pratama**

---Berkas-Berkas---

Proposal:

[DOWNLOAD](#)

Buku 1:

[DOWNLOAD](#)

Buku 2:

[DOWNLOAD](#)

Logbook atau Laporan Kegiatan Magang

[DOWNLOAD](#)

Video Magang

**Alfan Pratama**  
 10311600000099  
 2000  
 Nama Perusahaan : PT Indocrane  
 Alamat Perusahaan :  
 ---Tanggal Magang---  
 Tanggal Mulai :  
 Tanggal Selesai :  
[Tambah Nilai](#)

Nilai

Penilai	Nilai	
Dosen Pembimbing	90	<a href="#">DELETE</a>

Activate Window  
Go to Settings to activate this application.

Gambar 4.12 Antarmuka Profil Mahasiswa

Pada profil ini Admin bisa mengunduh file yang tersedia atau nilai yang sudah dimasukkan.

f. Antarmuka Register Mahasiswa

Pada register ini berujuan untuk menambah *user* dengan *role* sebagai Mahasiswa. Halaman ini hanya bisa diakses oleh *user* dengan *role* Admin. Tampilan pada antarmuka Register Mahasiswa dapat dilihat dari Gambar 4.13

The form consists of the following fields:

- Nama
- ID Dosen Pembimbing
- Angkatan
- NRP (jika kurang dari 14 char data tidak masuk)
- Nama Perusahaan
- Email

Buttons: Close, Submit

Gambar 4.13 Antarmuka Register Mahasiswa

g. Antarmuka Daftar Berkas-Berkas Mahasiswa

Berikut ini adalah tampilan dari Antarmuka Daftar Mahasiswa. Perbedaannya pada table ini bertujuan untuk mensortir *file* apakah ada yang belum *upload* berkas

Table: Daftar Berkas-Berkas Mahasiswa

NAMA	Proposal	Buku 1	Buku 2	Logbook	Video Magang
Allan Pratama					
Ipan	Cover (1).pdf	Transkrip Nilai, Khabib Nur Budiono.doc		E:\xampp\htdocs\php273F.tmp	logo-magang.png
khabib				Dark Blue Co Working Simple Presentation.png	
khabib				transkrip.jpg	
khabib					
Khabib Nur Budiono					
Marko Savana			5556.pdf	Cover.pdf	

Gambar 4.14 Antarmuka Daftar Berkas-Berkas Mahasiswa

## 4.2 Perbandingan Antarmuka Web dengan Localhost

Perbandingan diatas adalah perbandingan dari tampilan di Localhost dan Web pada Digitalisasi ini. Dengan alamat dan database sebagai berikut:

Alamat Localhost : localhost/laravel-crud/public

Alamat Web : magangdteo.com

Database Localhost : laravel-crud

Database Web : admin\_magangdteo2

Terjadinya semua persamaan tampilan diatas dikarenakan untuk *upload* menggunakan aplikasi FileZilla sehingga dapat *upload* dengan baik. Apabila tidak dapat *upload* dengan baik dan terlewat satu *file* maka akan merusak semua halaman *web*.

## 4.2 Data Mahasiswa Masuk

Tabel 4.1 Tabel Database Mahasiswa

id	user_id	Nama	Angkatan	NRP	Nama_Perusahaan
19	11	Pratamanu	2016	11111111	PT Nippon Japan
23	24	Pratamanu	2022	111111111111114	PT Nippon Japak
26	50	khabib	2016	1111111111111111111	PTku
27	51	sfgsdfg34	2016	1111111111111111111	ini nama pers
28	52	NANAKomatsu	2016	1111111111111111111	Ptku
30	54	khabib	2022	1111111111111111111	PTku
31	57	ipan	1999	111111111111114	bri
32	58	Khabib Nur Budiono	2016	10311600000049	TernakNesia Tekno Farm
33	61	Alfan Pratama	2000	10311600000099	PT Indocrane
34	62	Marko Savana	1999	10319900000049	PT Culve
35	65	Khabib Lagi Lagi	1999	10311600000049	nnon
36	66	testinguser	2017	12345678912345	Jalan besar
37	68	testinguser	2017	12345678912345	Jalan besar
38	71	Khabib Nur Budiono	2016	10311600000049	PT Ternaknesia

Pada tabel ini terdapat 19 kolom untuk memasukkan data yang dibutuhkan ke dalam tabel Mahasiswa ini, yaitu kolom berisi id, user\_id, Nama, Angkatan, NRP, Nama\_Perusahaan, Alamat\_perusahaan, Tanggal\_Mulai, Tanggal\_Seleasai, Deskripsi, Buku1, Buku2, Proposal, Logbook, Video\_Magang, avatar, dosen\_id, created\_at dan updated\_at. Alasan dibalik nilai id yang tidak dari satu karena ini adalah tabel hasil dari impor database di localhost. Untuk data yang ada, sama seperti data yang ditampilkan pada *web*. Pada kolom Buku1, Buku2, Proposal, Logbook, Video\_Magang dan avatar akan berisi nama file yang di *upload* sedangkan berkas akan yang di *upload* disimpan dalam dokumen public oleh laravel.

Tabel 4.2 Tabel Database *upload* dan *download*

id	Tanggal_Seleasai	Deskripsi	Buku1	Buku2	Proposal	Logbook	Video_Magang	avatar
03 des 1999	NULL	NULL	NULL	NULL	NULL	E:\xampp\tmp\phpEF1F tmp	NULL	NULL
45	NULL	NULL	E:\xampp\tmp\php7303 tmp	E:\xampp\tmp\php7323 tmp	NULL	NULL	NULL	osaka_cas1512.png
NULL	NULL	NULL	NULL	NULL	NULL	Dark Blue Co Working Simple Presentation.png	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	transkrip.jpg	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	hfoahfioahsiothosahfiojadbfuifuncioncndiofnra	Transkrip Nilai_Khabib Nur Budiono.doc	NULL	NULL	Cover (1).pdf	E:\xampp\tmp\php273F tmp	logo-magang.png	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	5556.pdf	NULL	NULL	Cover.pdf	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Pada tabel 4.2 ini *file* untuk *upload* dan *download* dimasukkan. Tabel ini ada dalam tabel Mahasiswa. Karena kolom yang sangat banyak jika dijadikan dalam satu tabel tidak akan jelas terlihat. Untuk *upload* ini terdapat dalam kolom 11-16. *File* pada kolom ini akan ditampilkan pada Gambar 4.14.

### 4.3 Data User

Pada web terdapat tiga jenis *user*. User tersebut adalah Admin, Dosen dan Mahasiswa. Pembuatan User admin bisa ditambah secara manual ke dalam database yang ada, melalui login pada Plesk.

Untuk User Dosen bisa di register dengan car ditambahkan oleh admin. Pada register Mahasiswa bisa dilakukan oleh Mahasiswa sendiri dengan mendaftar pada [magangdteo.com/register](http://magangdteo.com/register). Selain itu juga bisa ditambahkan oleh Dosen maupun Admin. Pada proses register data yang dimasukkan akan disimpan pada dua tabel. Tabel tersebut adalah Tabel User dan tabel jika register Mahasiswa maka akan masuk Tabel Mahasiswa dan jika *register* dosen akan masuk ke Tabel Dosen.

Tabel 4.3 Register Mahasiswa

No	Nama	Status
1	Nana Komatsu	Berhasil
2	Khabib Nur b	Berhasil
3	Ipan	Berhasil
4	Marko Savana	Berhasil
5	Pratamanu	Berhasil

Dilihat dari tabel 4.3 pembuatan user Mahasiswa melalui Register sendiri berhasil.

Tabel 4.4 Register Mahasiswa oleh Admin

No	Nama	Status
1	Testinguser	Berhasil
2	Aldo Pradana	Berhasil
3	Khabib	Berhasil
4	Khabib Lagi Lagi	Berhasil
5	Alfan Pratama	Berhasil

Dilihat dari tabel 4.4 pembuatan user Mahasiswa melalui Register oleh Admin berhasil. Pada pembuatan akaun Mahasiswa bisa dilakukan dengan dua cara.

Tabel 4.5 Register Dosen oleh Admin

No	Nama	Status
1	Imung Nugroho S.Pd.	Berhasil
2	Afif Syahbana S.T M.T	Berhasil
3	Alwan Arib	Berhasil
4	Tanara Chevin	Berhasil
5	Alfan Satria	Berhasil

Dilihat dari tabel 4.5 pembuatan user Dosen melalui Register oleh Admin berhasil.

Pada pengujian tabel di atas dapat dilihat pengujian untuk membuat akun yang bertujuan untuk login sebagai *user* sesuai fungsi perorangan.

Tabel 4.6 Register Akun Baru

NO	Register Akun	Status
1	Pembuatan Akun Oleh Mahasiswa	Berhasil
2	Pembuatan Akun Mahasiswa oleh Admin	Berhasil
3	Pembuatan Akun Dosen oleh Admin	Berhasil

Dari pengujian yang dilakukan pada Tabel 4.3, Tabel 4.4 dan Tabel 4.5 dapat disimpulkan bahwa ketiga register atau membuat akun berhasil.

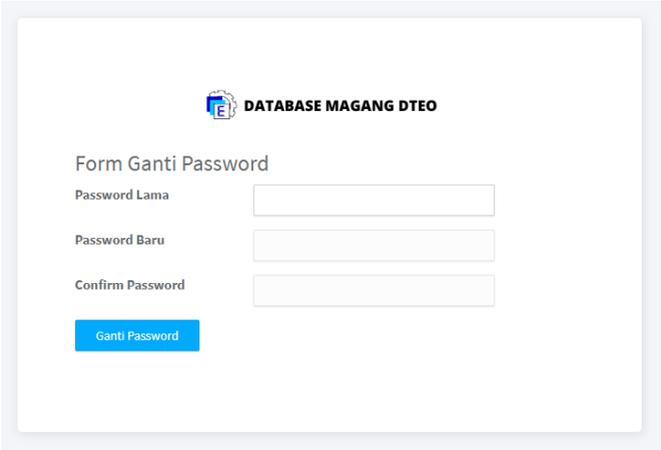
Tabel 4.7 Pengujian Edit Data Mahasiswa

Nama	NRP	Angkatan	Nama dan Alamat Perusahaan	Tanggal Mulai dan Selesai
Aldo Pradana	Berhasil	Berhasil	Berhasil	Berhasil
Alfan Pratama	Berhasil	Berhasil	Berhasil	Berhasil
Arif Rahman	Berhasil	Berhasil	Berhasil	Berhasil
Birly Kiyep	Berhasil	Berhasil	Berhasil	Berhasil
Ipan	Berhasil	Berhasil	Berhasil	Berhasil
Khabib	Berhasil	Berhasil	Berhasil	Berhasil
Khabib Nur Budiono	Berhasil	Berhasil	Berhasil	Berhasil
Khabib Lagi	Berhasil	Berhasil	Berhasil	Berhasil
Noval	Berhasil	Berhasil	Berhasil	Berhasil
Marko Savana	Berhasil	Berhasil	Berhasil	Berhasil
Nana Komatsu	Berhasil	Berhasil	Berhasil	Berhasil
Pratamanu B	Berhasil	Berhasil	Berhasil	Berhasil
Sgdf	Berhasil	Berhasil	Berhasil	Berhasil
Test User	Berhasil	Berhasil	Berhasil	Berhasil
Testinguser	Berhasil	Berhasil	Berhasil	Berhasil

Pada Tabel di atas adalah hasil dari sebuah pengujian edit data dari Mahasiswa yang sudah memiliki akun. Untuk edit data sendiri juga bisa menambahkan atau upload berkas yang telah ditentukan. Hasil dari mengedit data sendiri akan dimasukkan ke *database*.

#### 4.4 Mengganti Password Akun

Pada menu ganti password bisa di akses melalui *navbar* lalu akan masuk ke halaman berikut



The screenshot shows a web form titled "Form Ganti Password" for "DATABASE MAGANG DTEO". It includes three input fields: "Password Lama", "Password Baru", and "Confirm Password". A blue button labeled "Ganti Password" is positioned below the fields.

Gambar 4.15 Tampilan Mengganti Password

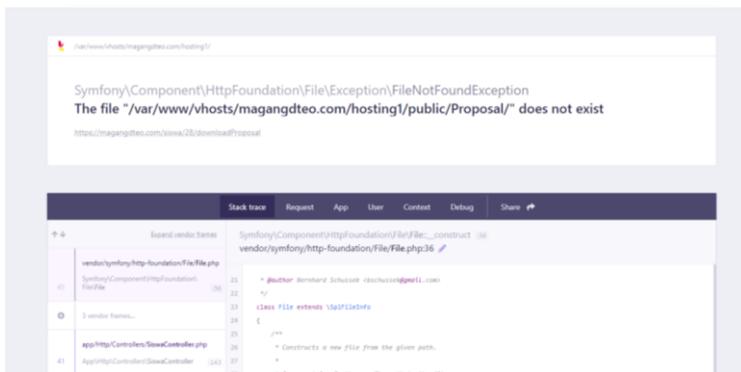
Hasil dari Form Ganti Password bisa dilihat di tabel 4.8. Dari tabel bisa dilihat bahwa setiap jenis user bisa melakukan ganti password

Tabel 4.8 Tabel Hasil Form Ganti Password

Nama	Jenis User	Password Lama	Password Baru	Status
Khabib	Admin	Rahasia	rahasiakita	Berhasil
Imung Nugroho S.Pd.	Dosen	vokasidteo	Rahasia	Berhasil
Marko Savana	Mahasiswa	2016	1999	Berhasil

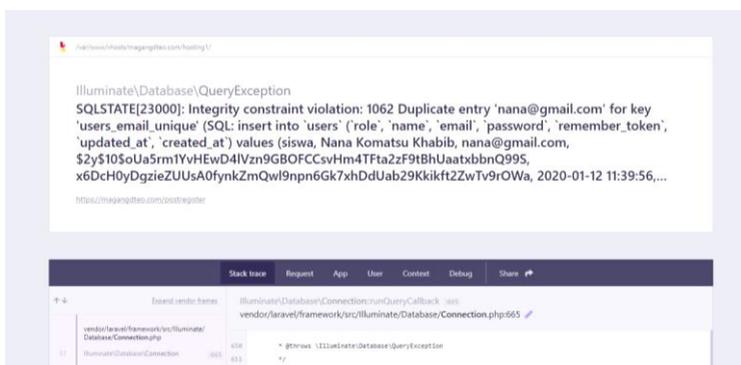
#### 4.5 Error

Error menampilkan suatu kesalahan dalam pembuatan suatu fungsi di dalam sebuah web. Pada web ini terdapat dua tindakan yang akan mengeluarkan error. Error tersebut akan ditampilkan pada pembahasan berikut ini.



Gambar 4.16 Error Download File Mahasiswa.

Pada error ini menampilkan error ketika pada suatu *user* siswa belum melakukan *upload* berkas namun *user* lain melakukan *download* pada *file* kosong tersebut. Sehingga ketika Admin melakukan download pada *file* pada Mahasiswa yang belum melakukan *upload* maka error ini terjadi.



Gambar 4.17 Error Email Ganda

Pada eror ini terjadi jika ada yang ingin melakukan pendaftaran sebuah akun namun email yang digunakan untuk mendaftarkan sudah dipakai user yang ada. Maka akan menampilkan error ini.

-----Halaman ini sengaja dikosongkan-----

## **BAB V**

### **PENUTUP**

Dari hasil yang telah didapatkan selama proses dan pembuatan serta proses analisis data untuk Proyek Akhir ini, maka dapat diambil beberapa kesimpulan dan saran yang berguna untuk perbaikan dan pengembangan agar nantinya bisa bermanfaat.

#### **3.1 Kesimpulan**

Dari hasil penelitian dan pembahasan dapat disimpulkan bahwa:

1. Pembuatan web menggunakan *Framework* Laravel yang merupakan MVC atau *Model View Controller* membantu dalam proses pembuatan web akan tetap untuk *upload* ke *hosting* sulit dikarenakan ada *requirement* software di dalam *hosting*.
2. Digitalisasi ini dibuat agar memudahkan pengolahan data magang.

#### **3.2 Saran**

Mengingat adanya keterbatasan dari penulis, beberapa saran yang perlu dilakukan agar *website* ini siap untuk dipakai antara lain:

1. Pengujian yang dilakukan oleh banyak *user*.
2. Penambahan autentifikasi perihal login. Jika login salah *password* maka dapat pemberitahuan.
3. Menyambungkan dengan Email sebagai autentifikasi lanjutan.

-----Halaman ini sengaja dikosongkan-----

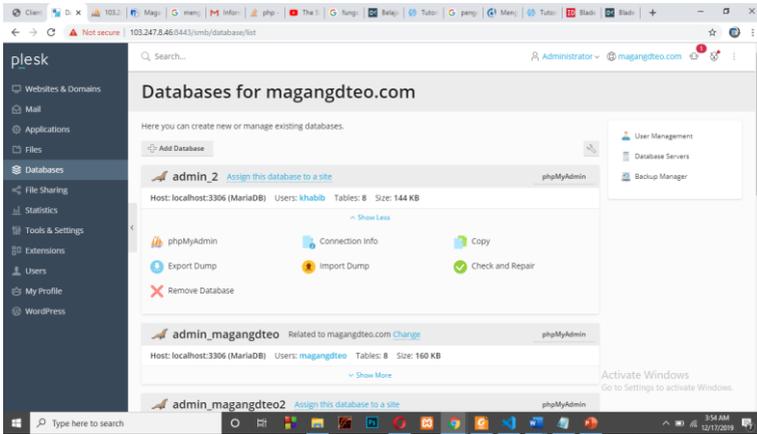
## DAFTAR PUSTAKA

- [1] Abdulkarim, A., 2006. Pendidikan Kewarganegaraan. Bandung: Grafindo
- [2] Wardana, S. M., 2010. Menjadi Master PHP dengan Framework Codeigniter. Jakarta: Elex Media Komputindo
- [3] Kadir, A. 2008. Dasar Pemrograman Web Dinamis Menggunakan PHP Bandung, Elex Media Komputindo.
- [4] Pribadi, A., 2016. Konsep dan Implementasi Pemrograman Laravel 5. Yogyakarta:Lokomedia.
- [5] Hidayat, Rahmat. 2010. Cara Praktis Membangun Website Gratis : Pengertian Website. Jakarta : PT Elex Media Komputindo Kompas, Granedia
- [6] Indriasari, S., 2012. Digitalisasi Berbasis Web Untuk Membantu Kegiatan Tracer Study Program Diploma Institut Pertanian Bogor. pp. 84-102.
- [7] Andre, P., 2017. MySql dan MariaDB untuk Pemula. Jakarta : Duniaikom
- [8] Abdullah, Rohi, 2018. Pemrograman Web Untuk Pemula. Bandung. Elex Media Komputindo.
- [9] Andre, P., 2017. MySql dan MariaDB untuk Pemula. Jakarta : Duniaikom
- [10] Aryanto, 2016. Pengelahan Database MySQL. Yogyakarta: Deepublish
- [11] Andre, P., 2019. CSS Uncover. Jakarta : Duniaikom
- [12] Widodo, B. P., & Purnomo, H. D. (2016). Perancangan Aplikasi Pencarian Layanan Kesehatan Berbasis HTML Geolocation. Jurnal Sistem Komputer.
- [13] Andre, P., 2018. Bootstrap Uncover, Belajar Bootstrap 4. Jakarta : Duniaikom

-----Halaman ini sengaja dikosongkan-----



# Plesk



## LAMPIRAN B

SiswaExport.php

<?php

```
namespace App\Exports;
```

```
use App\Siswa;
```

```
use Maatwebsite\Excel\Concerns\FromCollection;
```

```
use Maatwebsite\Excel\Concerns\WithMapping;
```

```
use Maatwebsite\Excel\Concerns\WithHeadings;
```

```
class SiswaExport implements FromCollection, WithMapping,  
WithHeadings
```

```
{
```

```
/**
```

```
 * @return \Illuminate\Support\Collection
```

```
 */
```

```
public function collection()
```

```
{
```

```
    return Siswa::all();
```

```
}
```

```
public function map($siswa): array
```

```
{
```

```
    return [
```

```
        $siswa->Nama,
```

```
        $siswa->Angkatan,
```

```
        $siswa->NRP,
```

```
        $siswa->Nama_Perusahaan,
```

```
        $siswa->Alamat_Perusahaan,
```

```
        $siswa->Tanggal_Mulai,
```

```
        $siswa->Tanggal_Selesai,
```

```
    ];
```

```
}
```

```
public function headings(): array
```

```
{
```

```
    return [
```

```
        'Nama',
```

```
        'Angkatan',
        'Nrp',
        'Nama Perusahaan Magang',
        'Alamat Perusahaan',
        'Tanggal Mulai',
        'Tanggal Selesai',
    ];
}
}
```

### Helpers/Global.php

```
<?php
Use App\Siswa;
Use App\Guru;

function totalSiswa()
{
    return siswa::count();
}
function totalDosen()
{
    return Guru::count();
}
```

### AuthController.php

```
<?php

namespace App\Http\Controllers;
use Auth;
use Illuminate\Http\Request;

class AuthController extends Controller
{
    public function login()
```

```

    {
        return view('auths.login');
    }

    public function postlogin(Request $request)
    {
        if(Auth::attempt($request->only('email','password')))
        {
            return redirect('/dashboard');
        }
        return redirect('/login');
    }
    public function logout()
    {
        Auth::logout();
        return redirect('/login');
    }
}

```

DashboardController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
class DashboardController extends Controller
```

```

{
    public function Index()
    {
        return view('dashboard.index');
    }
}

```

## GuruController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use App\Guru;
```

```
use App\Mapel;
```

```
use Illuminate\Support\Str;
```

```
class GuruController extends Controller
```

```
{
```

```
    public function profile($id)
```

```
    {
```

```
        $guru = Guru::find($id);
```

```
        return view('guru.profile',['guru' => $guru]);
```

```
    }
```

```
    public function index(Request $request)
```

```
    {
```

```
        $data_guru = Guru::all();
```

```
        return view('guru.index',['data_guru' => $data_guru]);
```

```
    }
```

```
    public function create(Request $request)
```

```
    {
```

```
        //dd($request->all());
```

```
        //Insert ke table users
```

```
        $user = new \App\User;
```

```
        $user->role = 'dosen';
```

```
        $user->name = $request->nama;
```

```
        $user->email = $request->email;
```

```
        $user->password = bcrypt('vokasidteo');
```

```
        $user->remember_token = Str::random(60);
```

```
        $user->save();
```

```

//Insert ke table guru
$request->request->add(['user_id'=> $user->id ]);
$dosen = Guru::create($request->all());

return redirect('/dosen');

}
public function delete($id)
{
    $guru = Guru::find($id);
    $guru->delete($guru);
    return redirect('/dosen')->with('sukses','Data Berhasil Dihapus');
}

public function profildosen()
{
    //$siswa = auth()->user()->siswa;
    //auth()->user()->guru->nama
    return view('guru.profil Dosen');
}
}

```

SiswaController.php

```
<?php
```

```

namespace App\Http\Controllers;

use Illuminate\Http\Request;
use App\siswa;
use Illuminate\Support\Str;
use Illuminate\Support\Facades\Storage;
use DataTables;

```

```

use App\Exports\SiswaExport;
use Maatwebsite\Excel\Facades\Excel;

class SiswaController extends Controller
{
    public function index1(Request $request)
    {
        if($request->has('cari')){
            $data_siswa = siswa::where('Nama','LIKE','% ' . $request-
>cari. '%')->paginate();
        }else{
            $data_siswa = siswa::all();
        }
        return view('siswa.index1',['data_siswa' => $data_siswa]);
    }
    public function create(Request $request)
    {
        //dd($request->all());
        $this->validate($request,[
            'NRP' => 'required|min:14',
            'avatar' => 'mimes:jpeg,bmp,png',
        ]);
        //Insert ke table users
        $user = new \App\User;
        $user->role = 'siswa';
        $user->name = $request->Nama;
        $user->email = $request->email;
        $user->password = bcrypt('rahasia');
        $user->remember_token = Str::random(60);
        $user->save();

        //Insert ke table siswa
        $request->request->add(['user_id'=> $user->id ]);
        $siswa = siswa::create($request->all());
        if($request->hasFile('avatar')){

```

```

        $request->file('avatar')->move('images/',$request-
>file('avatar')->getClientOriginalName());
        $siswa->avatar          =          $request->file('avatar')-
>getClientOriginalName();
        $siswa->save();
    }
    return redirect('/siswa');
}
public function edit($id)
{
    $siswa = siswa::find($id);
    return view('siswa/edit',['siswa' => $siswa]);
}
public function update(Request $request,$idsiswa)
{
    //dd($request->all());
    $siswa = siswa::find($idsiswa);
    $siswa->update($request->all());
    if($request->hasFile('avatar')){
        $request->file('avatar')->move('images/',$request-
>file('avatar')->getClientOriginalName());
        $siswa->avatar          =          $request->file('avatar')-
>getClientOriginalName();
        $siswa->save();

    }
    if($request->hasFile('Logbook')){
        $request->file('Logbook')->move('Logbook/',$request-
>file('Logbook')->getClientOriginalName());
        $siswa->Logbook          =          $request->file('Logbook')-
>getClientOriginalName();
        $siswa->save();

    }
    if($request->hasFile('Proposal')){

```

```

        $request->file('Proposal')->move('Proposal/', $request-
>file('Proposal')->getClientOriginalName());
        $siswa->Proposal = $request->file('Proposal')-
>getClientOriginalName();
        $siswa->save();
    }

    if($request->hasFile('Buku1')){
        $request->file('Buku1')->move('Buku1/', $request-
>file('Buku1')->getClientOriginalName());
        $siswa->Buku1 = $request->file('Buku1')-
>getClientOriginalName();
        $siswa->save();
    }

    if($request->hasFile('Buku2')){
        $request->file('Buku2')->move('Buku2/', $request-
>file('Buku2')->getClientOriginalName());
        $siswa->Buku2 = $request->file('Buku2')-
>getClientOriginalName();
        $siswa->save();
    }

    if($request->hasFile('Video_Magang')){
        $request->file('Video_Magang')-
>move('Video_Magang/', $request->file('Video_Magang')-
>getClientOriginalName());
        $siswa->Video_Magang = $request->file('Video_Magang')-
>getClientOriginalName();
        $siswa->save();
    }

    return redirect('/dashboard');
}
public function delete($id)
{
    $siswa = siswa::find($id);

```

```

        $siswa->delete($siswa);
        return redirect('/siswa')->with('sukses','Data Berhasil Dihapus');
    }
    public function profile($id)
    {
        $siswa = siswa::find($id);
        $matapelajaran = \App\Mapel::all();
        //dd($siswa);
        return view('siswa.profile',['siswa' => $siswa,'matapelajaran' =>
$matapelajaran]);
    }
    public function addnilai(Request $request,$idsiswa)
    {
        $siswa = siswa::find($idsiswa);
        if($siswa->mapel()->where('mapel_id',$request->mapel)-
>exists()){
            return redirect('/siswa/'.$idsiswa.'/profile')->with('error','Data
Nilai Sudah Ada');
        }
        $siswa->mapel()->attach($request->mapel,['nilai' => $request-
>nilai]);

        return redirect('/siswa/'.$idsiswa.'/profile')->with('sukses','Data
Berhasil Dimasukkan');
    }
    public function deletenilai($idsiswa,$idmapel)
    {
        $siswa = siswa::find($idsiswa);
        $siswa->mapel()->detach($idmapel);
        return redirect()->back()->with('sukses','Data Berhasil Dihapus');
    }
    public function getdatasiswa()
    {
        $siswa = siswa::select('siswa.*');
    }

```

```

        return \DataTables::eloquent($siswa)->toJson();
    }
    public function downloadLogbook($id)
    {
        $siswa = siswa::find($id);
        return response()->download(public_path('Logbook/'.$siswa-
>Logbook));
    }
    public function downloadProposal($id)
    {
        $siswa = siswa::find($id);
        return response()->download(public_path('Proposal/'.$siswa-
>Proposal));
    }
    public function downloadBuku1($id)
    {
        $siswa = siswa::find($id);
        return response()->download(public_path('Buku1/'.$siswa-
>Buku1));
    }
    public function downloadBuku2($id)
    {
        $siswa = siswa::find($id);
        return response()->download(public_path('Buku2/'.$siswa-
>Buku2));
    }
    public function downloadVideo_Magang($id)
    {
        $siswa = siswa::find($id);
        return response()-
>download(public_path('Video_Magang/'.$siswa->Video_Magang));
    }

```

```

}
public function profilsiswa()
{
    //$siswa = auth()->user()->siswa;
    return view('siswa.profilesiswa');
}
public function daftarbimbingan(Request $request,$dosen_id)
{
    $siswabimbingan = siswa::where('dosen_id',$dosen_id)->get();
    //dd($siswabimbingan);
    return view('siswa.daftarbimbingan',['siswabimbingan' =>
$siswabimbingan]);
}
public function index2()
{
    $siswaindex2 = siswa::all();
    return view('siswa.index2',['siswaindex2' => $siswaindex2]);
}
public function export()
{
    return Excel::download(new SiswaExport, 'Siswa.xlsx');
}
}

```

SiteController.php

```
<?php
```

```
namespace App\Http\Controllers;
```

```
use Illuminate\Http\Request;
```

```
use Illuminate\Support\Str;
```

```
class SiteController extends Controller
```

```

{
    public function register()
    {
        return view('sites.register');
    }
    public function postregister(Request $request)
    {
        $user = new \App\User;
        $user->role = 'siswa';
        $user->name = $request->Nama;
        $user->email = $request->email;
        $user->password = bcrypt($request->password);
        $user->remember_token = Str::random(60);
        $user->save();

        //Insert ke table siswa
        $request->request->add(['user_id'=> $user->id ]);
        $siswa = \App\siswa::create($request->all());

        return redirect('/login')->with('sukses','Berhasil Mendaftar');
    }
}

```

Dan masih banyak lagi *source code*

## RIWAYAT HIDUP PENULIS



Nama : Khabib Nur Budiono  
TTL : Surabaya, 2 September  
1997  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat Rumah : Jl. Kasatrian No 5 Kec.  
Sutojayan Blitar  
Nomor HP : 085791353447  
E-mail : khabibnb@gmail.com

### RIWAYAT PENDIDIKAN

- 2002-2004 : TK Kartika 37
- 2004-2010 : SDN Kalipang 01
- 2010-2013 : SMP Negeri 1 Blitar
- 2013-2016 : SMA Negeri 1 Blitar
- 2016 – sekarang : Program Departemen Teknik Elektro  
Otomasi, Fakultas Vokasi, ITS

### PENGALAMAN MAGANG

- Magang di PT Ternaknesia Farm Innovation

### PENGALAMAN KEPANITIAAN

- IARC 2017
- IARC 2018
- LKMM PRA-TD
- MRAC 2018

-----Halaman ini sengaja dikosongkan-----