BAB IV

HASIL DAN PEMBAHASAN

A. Hasil Penelitian

Penelitian tugas akhir skripsi ini bertujuan untuk mengembangkan Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* dan menjamin kualitas sistem yang dikembangkan dengan diuji berdasarkan Standar ISO 25010 yang meliputi aspek *functional suitability, usability,* dan *performance efficiency*. Berikut hasil pengembangan aplikasi dan hasil pengujian yang telah dilakukan:

1. Pengembangan Perangkat Lunak

Pengembangkan Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* menggunakan metode pengembangan *Agile Unified Process* (AUP) dengan 4 fase yaitu *inception, elaboration, construction,* dan *transition*, dimana pada setiap fase memiliki tahapan-tahapan *model, implementation, test, deployment, configuration management, project management,* dan *environment*. Berikut langkah-langkah yang dilakukan pada proses pengembangan sistem:

a. Fase *Inception*

Fase *inception* berfokus pada pembuatan model proses bisnis (*business modelling*) yang dibutuhkan oleh sistem. Hasil dari *business modeling* adalah

kebutuhan sistem yang akan dikembangkan (*system requirements*). Berikut hasil tahapan pada proses *inception* :

1) *Model*

Pada tahap ini untuk menganalisis kebutuhan *user* yang akan diimplementasikan ke dalam Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* menggunakan metode wawancara dan observasi. Metode wawancara dilakukan dengan Bapak Muslikhin, S.Pd., M.Pd selaku Koordinator Praktik Industri Jurusan Pendidikan Teknik Informatika serta mahasiswa urusan Pendidikan Teknik Elektronika dan Informatika yang belum maupun telah melaksanakan Praktik Industri. Berikut hasil wawancara yang telah dilakukan peneliti :

a) *Business Modelling*

Kegiatan *business modelling* dimulai pada tahap studi literatur yang bersumber dari jurnal penelitian, skripsi, buku-buku dan internet yang berkaitan dengan Sistem *Tracer*, Pengembangan *Website* berbasis *MEAN Stack* dan metode pengembangan perangkat lunak.

- Hasil Wawancara

➢ Dengan Koordinator Praktik Industri

- Jurusan Pendidikan Teknik Elektronika dan Informatika secara formal belum menjalin kerjasama dengan Perusahaan/Industri/Bengkel untuk penyaluran mahasiswa yang akan melaksanakan Praktik Industri, namun

ada kolega dosen yang meminta rekomendasi/masukan untuk menyalurkan mahasiswa untuk melaksanakan Praktik Industri di perusahaannya.

- Terdapat daftar riwayat pelaksanaan Prakik Industri pada Sistem Informasi PKL/PI milik LPPMP UNY, namun penulis ketika mencoba mengakses url address yang diberikan, tidak menemukan daftar riwayat pelaksanaan Prakik Industri pada sistem tersebut.
- Terdapat beberapa keluhan dari mahasiswa ketika menemukan Perusahaan/Industri/Bengkel untuk dijadikan sebagai tempat Praktik Industri.
- Mahasiswa mengeluhkan ketika mendapatkan tempat Praktik Industri ternyata tidak sesuai ekspektasi.
- Perusahaan/Industri/Bengkel mengeluhkan terkadang mahasiswa Praktik Industri kurang komunikatif, dan kurang serius ketika melaksanakan Praktik Industri.

➢ Mahasiswa angkatan 2017 yang akan melaksanakan PI

- Mahasiswa sudah tau ada kegiatan Praktik Industri.
- Sebagian mahasiswa belum menemukan tempat Praktik Industri.
- Sebagian mahasiswa mencari referensi tempat Praktik Industri dari kakak angkatan.
- Sebagian mahasiswa sudah menetukan tempat Praktik Industri yang sesuai dengan bidang keahlian yang dimiliki.

- Hasil Observasi

Metode observasi dilakukan ketika peneliti mencari referensi-referensi Sistem *Tracer*. Peneliti melakukan observasi *business process* pada Sistem *Tracer* yang pernah dikembangkan seperti *Tracer Study*, yaitu merupakan sistem yang dikembangkan dengan tujuan melacak rekam jejak alumni suatu Lembaga Pendidikan ketika telah selesai menempuh suatu jenjang pendidikan. Selain itu, observasi juga dilakukan di Jurusan Pendidikan Teknik Elektronika dan Informatika. Berikut adalah hasil dari observasi yang dilakukan, yaitu :

- Pada Jurusan Pendidikan Teknik Elektronika dan Informatika belum tersedia media informasi berbasis *website* secara terpusat yang menyediakan daftar riwayat industri/perusahaan/bengkel yang pernah dilaksanakan oleh mahasiswa angkatan sebelumnya.
- Sistem *Tracer Study* pada era saat ini dikembangkan dalam bentuk *website* agar mudah diakses melalui *browser mobile* maupun *desktop*, sehingga dapat memberikan kemudahan akses bagi alumni yang akan mengisi data pada Sistem *Tracer Study*.
- Aktor yang terlibat pada dalam Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* yaitu Mahasiswa dan Admin. Hal ini didasarkan karena sistem ini hanya ditujukan untuk menyalurkan informasi riwayat tempat Praktik Industri secara terpusat mahasiswa Jurusan Pendidikan Teknik Elektronika dan Informatika agar dapat memberikan referensi tempat Praktik Industri bagi mahasiswa yang akan melaksanakan Praktik Industri.

- Teknologi atau bahasa pemrograman yang digunakan untuk mengembangkan *website* bermacam-macam seperti *PHP*, *Phyton*, *JavaScript*, dan sebagainya.
- Peneliti memutuskan untuk menggunakan *JavaScript* sebagai perangkat pengembangan perangkat lunak yaitu menggunakan *NodeJS* sebagai bahasa *server*, *ExpressJS* sebagai *Framework* yang menangani sisi *back-end* sistem, dan menggunakan *Angular* sebagai sisi *front-end* atau bagian yang berinteraksi dengan pengguna, dan *MongoDB* sebagai *database* untuk menyimpan data.

b) *System Requirements*

- Sistem Informasi berbasis *website* dengan menggunakan teknologi *responsive website*, agar dapat digunakan pada *desktop* maupun *mobile* dengan berbagai macam *platform* maupun sistem operasi yang digunakan oleh pengguna.
- Pengembangan Sistem Informasi menggunakan 3 Framework *JavaScript* yaitu *NodeJS* sebagai bahasa *server*, *ExpressJS* sebagai *back-end*, *Angular* sebagai *front-end*, dan *MongoDB* sebagai *database*, dengan menggunakan standar arsitektur *web service REST API* sebagai perantara interaksi data pada sisi *back-end*.
- Menggunakan *Cloud Application Platform Heroku* sebagai media *Deployment*.
- Menggunakan *MLab* sebagai *MongoDB Database Hosting* atau sebagai *database* untuk *MongoDB.*
- Memberikan *custom domain* agar pengguna mudah dalam mengakses sistem informasi yang dikembangkan.
- Terdapat 2 level/peran pengguna dalam sistem yaitu Mahasiswa dan Admin.

- Mahasiswa diharuskan melakukan registrasi terlebih dahulu dengan memasukkan NIM yang valid (NIM Jurusan Pendidikan Teknik Elektronika dan Informatika), Nama Lengkap, Email dan Password agar dapat mengakses *website*.
- Fitur utama yaitu menampilkan daftar riwayat industri/perusahaan/bengkel yang pernah dilakukan oleh mahasiswa Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
- Pengguna mahasiswa yang telah terdaftar dapat menambah tempat Praktik Industri, atau dapat memilih tempat Praktik Industri yang sudah tersedia di dalam daftar tempat Praktik Industri.
- Pengguna mahasiswa hanya dapat melakukan 1 (satu) kali input dalam membuat tempat Praktik Industri dan memiliki akses mengubah data apabila data pada tempat Praktik Industri yang telah dibuat terdapat kesalahan penulisan.
- Setiap pengguna mahasiswa dapat melakukan pembuatan laporan perusahaan (*report*), apabila terdapat data yang salah, dan hanya admin yang dapat melihat laporan tersebut.
- Pengguna admin memiliki akses penuh terhadap sistem, admin dapat melakukan manajemen CRUD (*Create*, *Read*, *Update*, *Delete*) daftar perusahaan tempat Praktik Industri.
- Pengguna admin yang dapat melakukan manajemen pengguna/mahasiswa, dan admin dapat mengubah peran mahasiswa menjadi admin.

- Pengguna admin dapat melakukan manajemen laporan, serta melakukan aksi untuk memberikan respon terhadap laporan tersebut.

2) *Configuration Management*

Pengembangan Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* pada penelitian kali ini, peneliti hanya mengerjakan tidak melibatkan tim.

3) *Project Management*

Membuat perencanaan jadwal (*planning schedule*) dalam pembuatan sistem. Tujuan dari *planning schedule* menjadi acuan bagi peneliti agar pelaksanaan penelitian berjalan dengan efektif dan diharapkan sesuai dengan estimasi watu yang telah direncanakan.

Tabel 10. *Schedule Planning*

	April	**Mei**	**Juni**	**Juli**
Inception				
Elaboration				
Construction				
Transition				

4) *Environment*

Hasil dari tahapan ini adalah menyiapkan perangkat keras yang digunakan untuk pembuatan desain sistem, pengembangan sistem, serta pengujiannya. Perangkat pengembangan yang digunakan adalah :

a) Perangkat Keras

- Laptop Asus X505ZA
- Handphone Samsung Galaxy On5 2016
- Modem Huawei HG8245A dengan Penyedia Jasa Layanan Internet IndiHome

b) Perangkat Lunak

- *Visual Studio Code*
- *StarUML*
- *MongoDB*
- *Angular CLI*
- *NodeJS*
- *NPM* (*Node Package Manager*)
- *ExpressJS*
- *Baslamics Mockup 3*
- *Postman*

b. Fase *Elaboration*

1) *Model*

Tahap *Model* pada Fase *Elaboration* berbeda dengan tahap *Model* pada Fase *Inception*. Tahap *Model* pada Fase *Elaboration* memiliki peran mengembangkan hasil dari Tahap *Model* pada Fase *Inception*. Berikut hasil dari *Model* pada Fase *Elaboration*:

a) *Business Modelling*

Tahap *Business Modelling* pada Fase *Elaboration* yaitu melakukan penjabaran deskripsi sistem atau perangkat lunak yang dikembangkan serta pemodelan *use case* untuk Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack*. Setelah diperoleh *use case*, langkah selanjutnya dijabarkan lagi menjadi langkah-langkah alur *use case* dalam bentuk skenario-skenario atau dapat disebut sebagai *expanded-use case*.

- Deskripsi Produk

Kegiatan ini memiliki tujuan untuk menjabarkan dan mengkaji produk yang dikembangkan lebih dalam lagi. Dan mengidentifikasi aktor, kata kerja, kata benda, dan masalah. Berikut merupakan penjabaran deskripsi sistem yang dikembangkan:

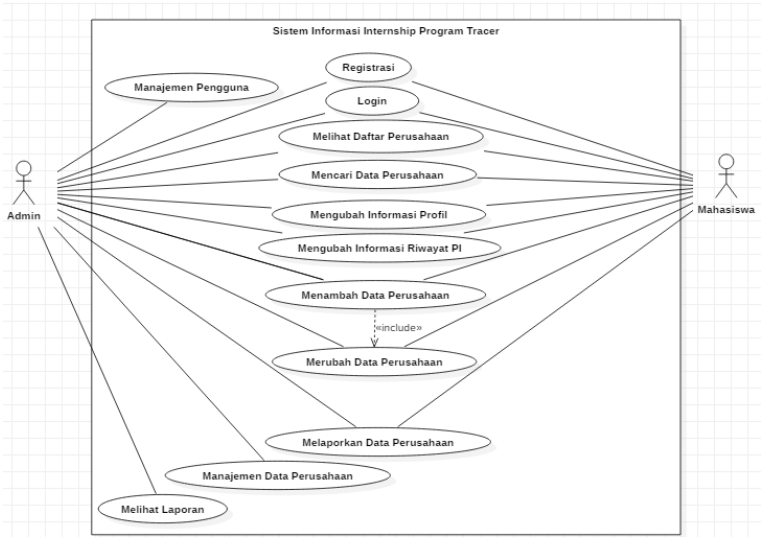
Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* merupakan sistem informasi yang dikembangkan untuk membantu mahasiswa mendapatkan referensi tempat Praktik Industri secara terpusat. Secara keseluruhan admin mengendalikan sistem ini, meliputi manajemen pengguna,

manajemen data perusahaan, manajemen laporan perusahaan. Mahasiswa melakukan registrasi dan login, agar dapat mengakses sistem. Kemudian mahasiswa yang telah melaksanakan praktik industri, masuk ke dalam halaman profil untuk menambahkan riwayat perusahaan. Apabila Perusahaan yang belum terdaftar pada pilihan yang disediakan, mahasiswa dapat membuat data perusahaan baru pada halaman perusahaan. Mahasiswa hanya dapat membuat data perusahaan sebanyak 1 (satu) kali, dan mendapatkan akses untuk merubah data perusahaan yang ditambahkan. Mahasiswa lain yang melaksanakan praktik industri yang sama, maka hanya perlu merubah informasi perusahaan praktik industri yang sudah tersedia pada halaman profil. Pada halaman profil, mahasiswa dapat merubah informasi profil pribadi. Mahasiswa yang belum melaksanakan, dapat melihat dan mencari daftar perusahaan pada halaman beranda, maupun halaman perusahaan. Mahasiswa dapat melaporkan data perusahaan, apabila terdapat kesalahan penulisan.

Biru = aktor, Kuning = kata kerja, Hijau = kata benda, Merah = masalah

- *Use Case Diagram*

Use Case Diagram dimodelkan dengan mengacu *automation scope* serta deskripsi sistem. Selain penjabaran *automation scope*, maka ditambahkan fungsi *login* dan *logout* sebagai proses autentikasi hak akses pengguna dengan pertimbangan keamanan sistem. Gambar 4 berikut menunjukkan *use case diagram* dari Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* :



Gambar 4. *Use Case Diagram*

Deskripsi aktor yang terlibat dapat disajikan pada Tabel 11 dan deskripsi *use case* dari masing-masing aktor disajikan pada Tabel 12.

Tabel 11. Deskripsi Aktor

No	**Nama Aktor**	**Deskripsi**
1	Admin	Aktor yang berperan sebagai admin dalam sistem ini adalah peneliti. Admin mengelola sebagian besar jalannya Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack*. Diantaranya adalah mengelola daftar perusahaan, pengguna, dan laporan.
2	Mahasiswa	Pengguna mahasiswa pada sistem ini dibagi menjadi 2 (dua), yaitu mahasiswa yang pernah melaksanakan praktik industri dan mahasiswa yang akan melaksanakan praktik industri. Mahasiswa

		yang telah melaksanakan praktik industri menyalurkan informasi tempat praktik industri mereka ke dalam sistem agar dapat diakses oleh mahasiswa yang akan melaksanakan praktik industri.

Tabel 12. Deskripsi *Use Case*

No	**Nama *Use Case***	**Deskripsi**
Umum		
1	Registrasi	Memasukkan data ke dalam sistem untuk mendapatkan akses login.
2	Login	Identifikasi hak akses pengguna.
3	Logout	Keluar dari sistem.
4	Merubah Informasi Akun	Merubah informasi akun seperti NIM, Nama, *Email*, dan *Password*.
Admin		
5	Manajemen Pengguna	Mengelola data pengguna, meliputi menghapus pengguna dan memberikan akses halaman admin kepada pengguna.
6	Manajemen Perusahaan	Mengelola data perusahaan, meliputi menambah, melihat, mengubah dan menghapus data.
7	Melihat Laporan	Mengelola data laporan kesalahan penulisan data perusahaan, meliputi menambah, melihat, dan menghapus data.
Mahasiswa		
8	Melihat Daftar Perusahaan	Melihat daftar perusahaan yang tersedia pada halaman beranda.

No	Nama *Use Case*	Deskripsi
9	Mencari Data Perusahaan	Mencari data perusahaan pada kolom pencarian pada beranda dan halaman perusahaan.
10	Manajemen Informasi Riwayat Praktik Industri	Mengelola informasi riwayat tempat praktik industri pada halaman profil meliputi menambah, mengubah, melihat, dan menghapus data.
11	Manajemen Data Perusahaan	Mengelola informasi perusahaan yang dibuat meliputi menambah, melihat, dan mengubah. Mahasiswa hanya dapat menambahkan data perusahaan sebanyak 1 (satu) kali, dan mendapatkan akses untuk merubah informasi yang ditambahkan.
12	Melaporkan Data Perusahaan	Membuat laporan kepada admin apabila terdapat kesalahan penulisan pada perusahaan agar mendapatkan tindakan oleh admin.

Setelah diperoleh *use case*, tahap berikutnya adalah menjabarkan skenario *use case*. Berikut skenario *use case* menambahkan informasi Riwayat Praktik Industri yang dijabarkan dari *use case* Manajemen Informasi Riwayat Praktik Industri. Skenario *use case* lebih lengkap terdapat pada Lampiran 7. Berikut salah satu Contoh Skenario *Use Case* Menambahkan Informasi Riwayat Praktik Industri :

Skenario *Use Case* Menambahkan Informasi Riwayat Praktik Industri

Primary Actor : Mahasiswa

Precondition : • Pengguna berada di “Halaman Profil”

Postcondition : • Informasi Riwayat PI berhasil disimpan

• Informasi jumlah mahasiswa PI pada suatu perusahaan bertambah

Main Flow : 1. Klik menu “Profil”

2. Klik sub-menu pada *scrolldown* “Informasi Riwayat Tempat Praktik Industri”
3. Pilih Perusahaan yang pernah menjadi tempat Praktik Industri
4. Klik tombol “Simpan”

Exception : 3a. Perusahaan belum terdaftar

1. Klik link “Tambah Perusahaan”
2. Klik icon + untuk menambahkan data perusahaan baru
3. **Skenario *Use Case* Menambahkan Perusahaan**

b) *System Requirements*

Proses ini menjelaskan lebih detail kebutuhan fungsionalitas Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack*. Proses ini dilakukan bertujuan agar kegiatan saat melakukan desain dan penulisan

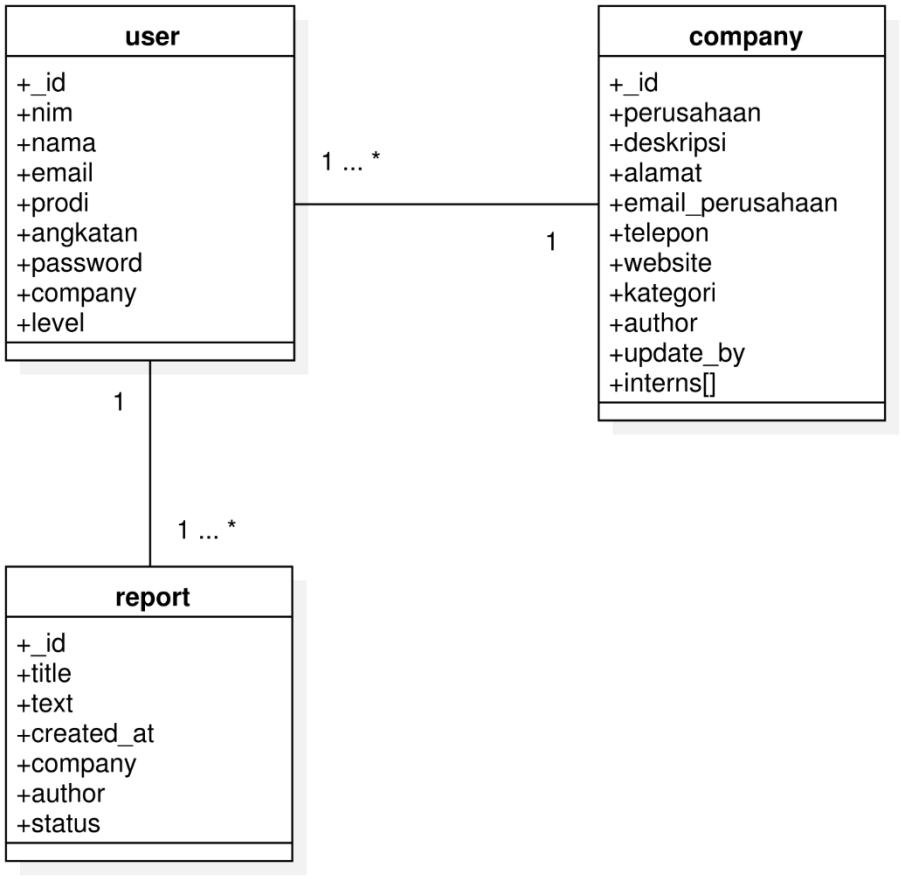
kode program lebih terfokus pada target. Berikut adalah hasil kebutuhan fungsionalitas sistem yang telah diperoleh:

- Terdapat pembagian peran level *user*/pengguna meliputi admin diwakili oleh peneliti dan mahasiswa. Setiap level pengguna mendapatkan hak akses dan fungsi yang berbeda.
- Mekanisme *login* dan pembatasan fungsi pada setiap level pengguna.
- Fungsi level pengguna Admin diantaranya mengelola data pengguna, mengelola data perusahaan, mengelola data laporan, mengelola data hak akses pengguna.
- Fungsi level pengguna mahaiswa diantaranya melihat daftar perusahaan, mencari data perusahaan, menambahkan data perusahaan, mengubah data perusahaan yang pernah ditambahkan. Mahasiswa hanya dapat menambahkan data perusahaan sebanyak 1 (satu) kali. Mahasiswa lain tidak dapat mengubah data perusahaan yang tidak dibuat oleh yang bersangkutan.

c) *Analysis & Design*

- *Class Diagram*

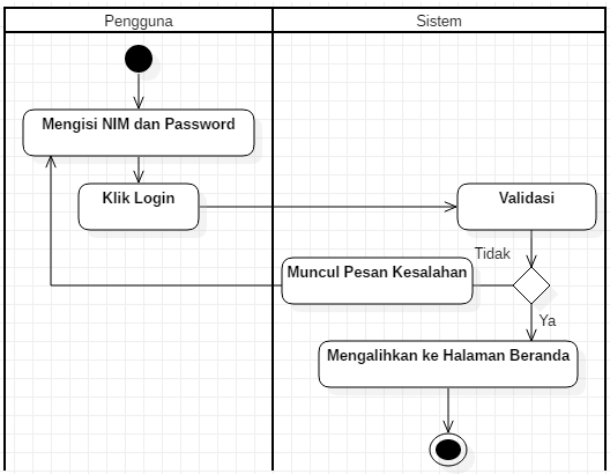
Pembuatan *class diagram* berpedoman terhadap *use case* yang telah dibentuk. Langkah pembuatan dimulai dari menentukan *class* berupa aktor yang terlibat di dalam *use case* dan *attribute* yang berupa identitas *class*, dan *behaviour* berupa *behavior* tingkah laku *class*. Lalu menentukan relasi antar *class* yang telah dibentuk. Hasil *class diagram* dapat dilihat pada Gambar 5 :



Gambar 5. *Class Diagram*

- *Activity Diagram*

Activity diagram atau Diagram Aktifitas berperan menggambarkan kegiatan yang terjadi antara aktor dan sistem. *Activity diagram* dibentuk dengan mengacu pada *use case* yang telah diperoleh. Salah satu *activity diagram* ditampilkan pada Gambar 6 :



Gambar 6. *LoginActivityDiagram*

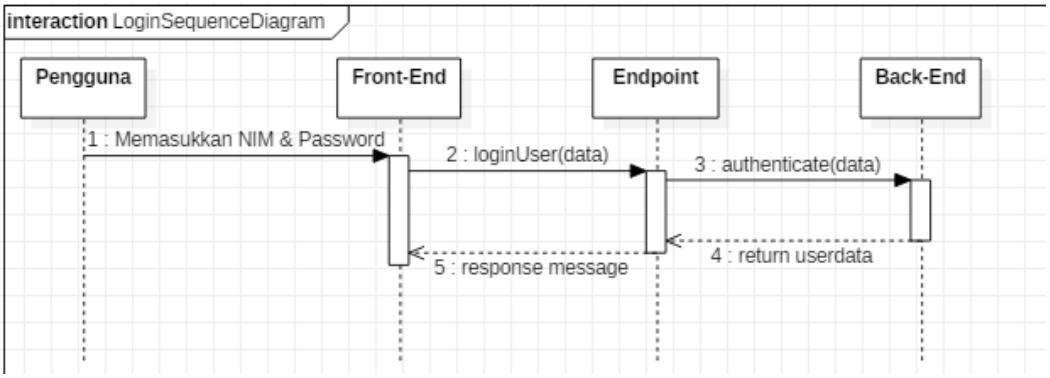
Activity Diagram selengkapnya dapat dilihat pada Lampiran 8. Daftar *activity diagram* dapat dilihat pada Tabel 13.

Tabel 13. Daftar *Activity Diagram*

No.	**Nama *Activity Diagram***	**Keterangan**
1	*LoginActivityDiagram*	Menggambarkan aktivitas login
2	*RegisterActivityDiagram*	Menggambarkan aktivitas registrasi
3	*ReadActivityDiagram*	Menggambarkan aktivitas membaca/melihat (*read*) data
4	*CreateUpdateActivityDiagram*	Menggambarkan aktivitas membuat (*create*) dan mengubah (*update*) data
5	*DeleteActivityDiagram*	Menggambarkan aktivitas menghapus (*delete*) data
6	*UpdateHistoryActivityDiagram*	Menggambarkan aktivitas mengubah/menambah riwayat praktik industri

- *Sequence Diagram*

Sequence diagram menampilkan satu set objek dan pesan yang dikirim dan diterima oleh objek-objek itu. Salah satu *sequence diagram* dapat dilihat pada Gambar 7, *sequence diagram* selengkapnya terdapat pada Lampiran 9 :

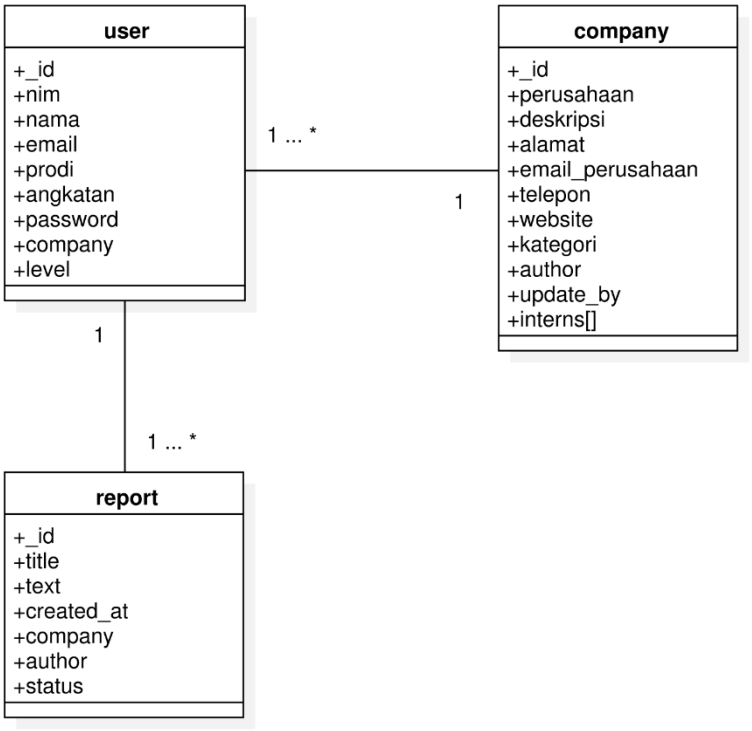


Gambar 7. *LoginSequenceDiagram*

- Desain *Database*

Kegiatan desain *database* ini memiliki tujuan untuk mendapatkan desain *collection* (istilah tabel dalam *nosql*) dengan relasi-relasinya yang diimplementasikan pada *Database Management System* bernama *MongoDB.*

Desain *database* yang dibuat mengacu pada *class*, *attribute*, dan relasi pada *class diagram* yang ditunjukkan pada Gambar 5. Desain *database* menggunakan model ERD (*Entity Relationship Diagram*) yang menggambarkan hubungan antar *class* dalam *database*. Hasil pemodelan ERD ditampilkan pada Gambar 8. Daftar *collection* pada *database* ditampilkan pada Tabel 14.



Gambar 8. *Entity Relationship Diagram* (ERD)

Tabel 14. Daftar *Collection* pada *Database*

No.	**Nama *Collection***	**Keterangan**
1.	*User*	*Collection* untuk menyimpan data pengguna
2.	*Company*	*Collection* untuk menyimpan data perusahaan
3.	*Report*	*Collection* untuk menyimpan data laporan

Keterangan *attribute* pada *Collection User* ditampilkan pada Tabel 15, keterangan *attribute* pada *Collection Company* ditampilkan pada Tabel 16, dan keterangan *attribute* pada *Collection Report* ditampilkan pada Tabel 17.

Tabel 15. Keterangan *Attribute* pada *Collection User*

No.	**Nama *Attribute***	**Tipe Data**	***Properties***
1.	_id	*Object_id*	*Primary Key*
2.	nim	*String*	*Unique*
3.	nama	*String*	
4.	email	*String*	*Unique*
5.	prodi	*String*	
6.	angkatan	*String*	
7.	password	*String*	
8.	company	*String*	
9.	status	*Number*	

Tabel 16. Keterangan *Attribute* pada *Collection Company*

No.	**Nama *Attribute***	**Tipe Data**	***Properties***
1.	_id	*Object_id*	*Primary Key*
2.	perusahaan	*String*	
3.	deskripsi	*String*	
4.	Email_perusahaan	*String*	*Unique*
5.	alamat	*String*	
6.	telepon	*String*	
7.	*website*	*String*	
8.	kategori	*String*	
9.	*author*	*Number*	
10.	*Update_by*	*String*	

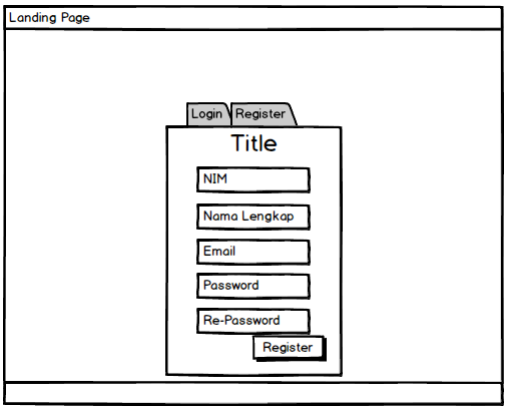
No.	Nama *Attribute*	Tipe Data	*Properties*
11.	*interns*	*Array*	

Tabel 17. Keterangan *Attribute* pada *Collection Report*

No.	Nama *Attribute*	Tipe Data	*Properties*
1.	_id	*Object_id*	*Primary Key*
2.	*title*	*String*	
3.	*text*	*String*	
4.	*author*	*String*	
5.	*company*	*String*	
6.	*Created_at*	*String*	

2) *Implementation*

Kegiatan *implementation* terdiri dari perancangan desain tampilan antarmuka (*user interface*) dalam bentuk *mockup* dan implementasi desain *database*. Kegiatan *implementation* ini menggunakan *software Balsamiq Mockups 3*. Sedangkan pembuatan desain *database* dilakukan dengan cara pembuatan model *database* pada *ExpressJS*. Gambar 9 berikut menunjukkan salah satu *Mockup Landing Page*, yaitu *Mockup* halaman *login* dan *register* :



Gambar 9. *Mockup Landing Page*

Desain *user interface* atau tampilan pengguna selengkapnya dapat dilihat pada Lampiran 10. Daftar desain tampilan pengguna Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* dapat dilihat pada Tabel 18.

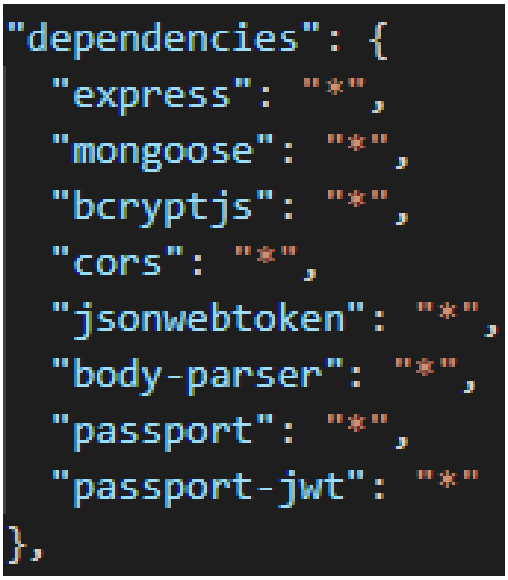
Tabel 18. Daftar Desain *Mockup*

No.	**Nama *Mockup***
1.	*Landing Page - Login Mockup*
2.	*Landing Page – Register Mockup*
3.	*Home Page Mockup*
4.	*Profile Page Mockup*
5.	*Company Page Mockup*
6.	*Admin-Dashboard Mockup*
7.	*Admin-Users Mockup*
8.	*Admin-Companies Mockup*
9.	*Admin-Reports Mockup*

c. Fase *Construction*

1) *Implementation*

Tahap *Implementation* pada Fase *Construction* dimulai dengan menyiapkan *Dependencies NodeJS* yang diperlukan untuk membangun sisi *back-end* dan *front-end* dari Sistem Informasi *Internship Program Tracer* berbasis *website* dengan *MEAN Stack*. Berikut adalah *Dependencies NodeJS* yang diperlukan untuk membangun sisi *back-end* ditampilkan dalam Gambar 10 :

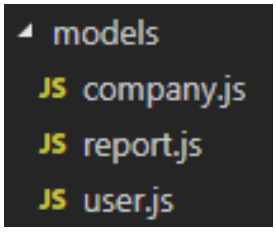


Gambar 10. Daftar *Dependencies NodeJS*

Implementasi pembuatan kode program menggunakan *Framework ExpressJS* dengan *text editor Visual Studio Code*. Penulisan kode program mengacu pada desain sistem yang telah dibuat pada tahap sebelumnya berdasarkan *business case* dan *use case* yang telah dibuat. *Source code* sepenuhnya terdapat pada link berikut : *https://github.com/fatihrizqon/mean*. Penulisan kode program dibagi menjadi tahap-tahap sebagai berikut :

a) *Model* (*Back-End*)

Model disini berbeda dengan istilah *model* pada tahapan-tahapan sebelumnya, *model* disini adalah istilah untuk menentukan *schema database* (istilah struktur *database* dalam *nosql*) dalam bentuk format data JSON (*JavaScript Object Notation*). Berikut adalah struktur *file* untuk menyusun *schema database* yang diperlukan :



Gambar 11. Struktur *Models*

Berikut merupakan salah satu *file models* untuk *schema user* :

```
const mongoose =
require('mongoose');const
bcrypt    =
require('bcryptjs');
const config    =
require('../config/database')
;
const UserSchema =
mongoose.Schema({
    nama:{
        type    : String
    },
    email:{
        type    : String,
        require : true,
        unique : true
    },
    nim:{
        type    : String,
        require : true,
        unique  : true
    },
    prodi:{
        type : String
    },
    angkatan:{
        type : String
    },
    password:{
        type    : String,
        require : true
    },
    company:{        type :
    String,
        ref  : 'company',
        unique  : false
    },
    level:{
        type : Number
    },
```

```
module.exports.addUser =
function(newUser, callback){
    bcrypt.genSalt(10, (err,
salt) => {

bcrypt.hash(newUser.password,
salt, (err, hash) => {
            if(err) throw err;
            newUser.password =
hash;

newUser.save(callback);
        });
    });
}
module.exports.getUserByNim =
function(nim, callback){
    const query = {nim: nim}
    User.findOne(query,
callback);
}

module.exports.getUserById =
function(id, callback){
    User.findById(id, callback);
}

module.exports.getUsers =
function(req, res, callback){
    User.find();
}

module.exports.addCompany =
function(req, res, user,
callback, err){
    console.log('Add User\'s
Company');
}
```

```
    status:{
        type    : String,
        require : false
    }
});
const User = module.exports =
mongoose.model('User',
UserSchema);
```

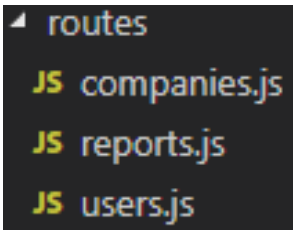
```
module.exports.getUserCompany =
function(nim, callback){
    res.send('Get User\'s
Company');
}

module.exports.comparePassword =
function(candidatePassword,
hash, callback){

bcrypt.compare(candidatePassword
, hash, (err, isMatch) => {
        if(err) throw err;
        callback(null, isMatch);
    });
}
```

b) *Routes* (*Back-End*)

Routes atau *routing* merupakan tahapan pembentukan *endpoints* pada sisi *back-end* yang berfungsi untuk membentuk operasi-operasi yang telah ditentukan pada *business case*. *Endpoints* dibuat dengan mengacu standar *RESTful API* dengan menggunakan metode HTTP (*HTTP Methods*) *GET*, *POST*, *PUT*, *DELETE*. Berikut adalah daftar *routes* yang diperlukan untuk membangun Sistem Informasi *Internship Program Tracer* berbasis *website* dengan *MEAN Stack* :



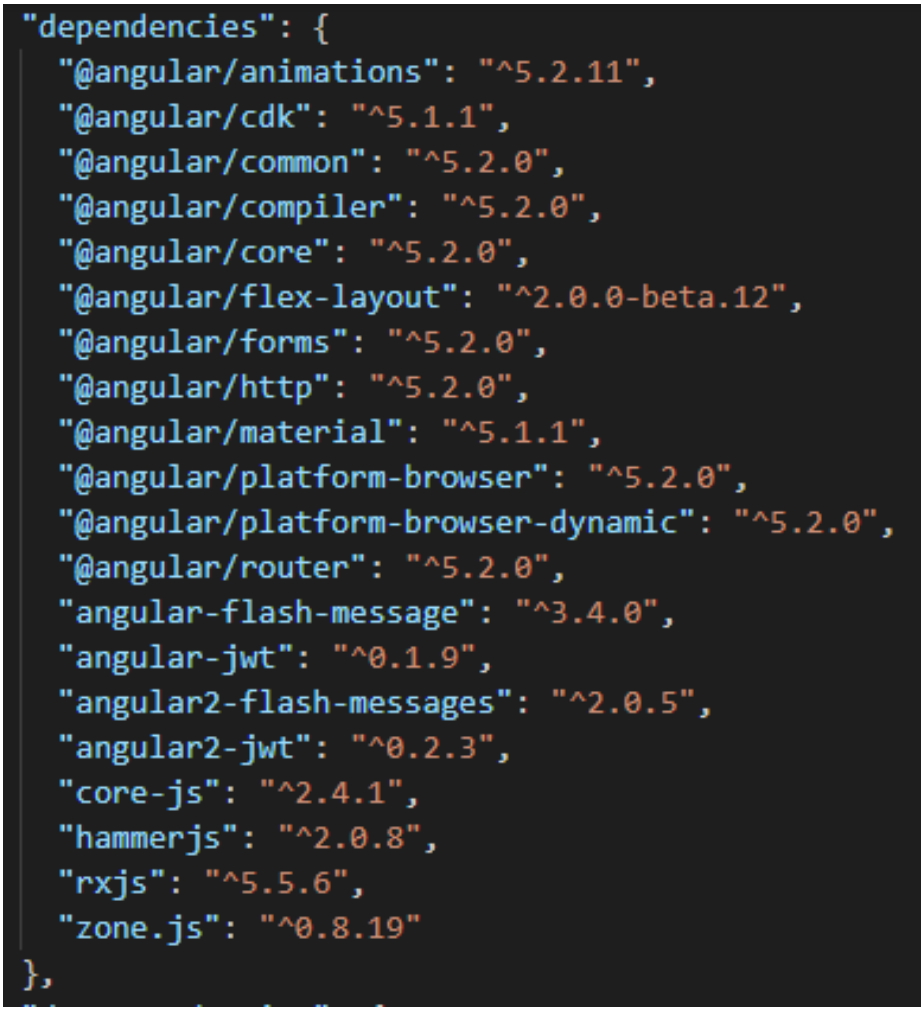
Gambar 12. Daftar *file routes*

Salah satu potongan kode *routes* untuk menampilkan daftar perusahaan :

```
router.get('/', function(req, res, next) {
    Company.find(function (err, companies) {
      if (err) return next(err);
      res.json(companies);
    });
});
```

c) *View* (*Front-End)*

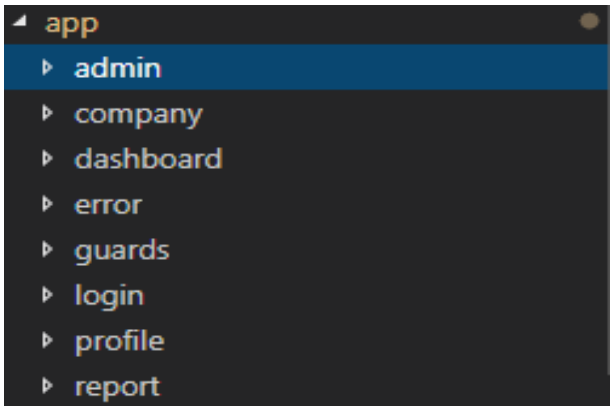
Sisi *view* (*front-end*) atau yang bagian berinteraksi dengan pengguna, dikembangkan menggunakan *framework javascript angular*. Instalasi *angular* yang dilakukan berbeda dengan sisi *back-end*. *Dependencies* yang diperlukan pada *angular* dapat dilihat pada Gambar 13 berikut :



Gambar 13. Daftar *Dependencies Angular*

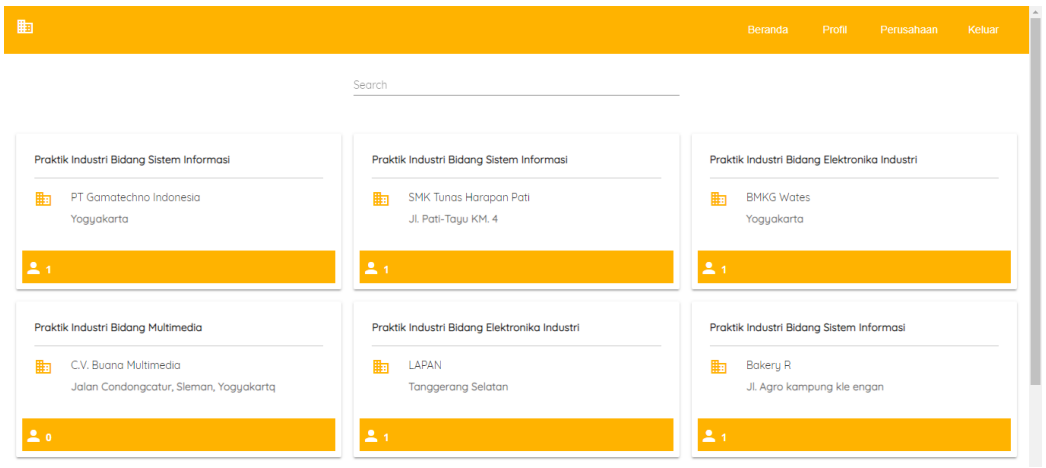
Pada sisi *front-end* diperlukan pembuatan daftar halaman-halaman *website* yang diperlukan dalam mengembangkan sistem. Halaman tersebut dibuat dalam bentuk

component (istilah dari *Angular*). Daftar halaman yang dibuat berdasarkan desain *mockups* yang telah didesain pada fase *elaboration* ditampilkan pada Gambar 14 berikut :



Gambar 14. Daftar Halaman Sistem

Hasil salah satu implementasi pengkodean dari desain *mockups* pada Halaman Dashboard ditampilkan pada Gambar 15 berikut :

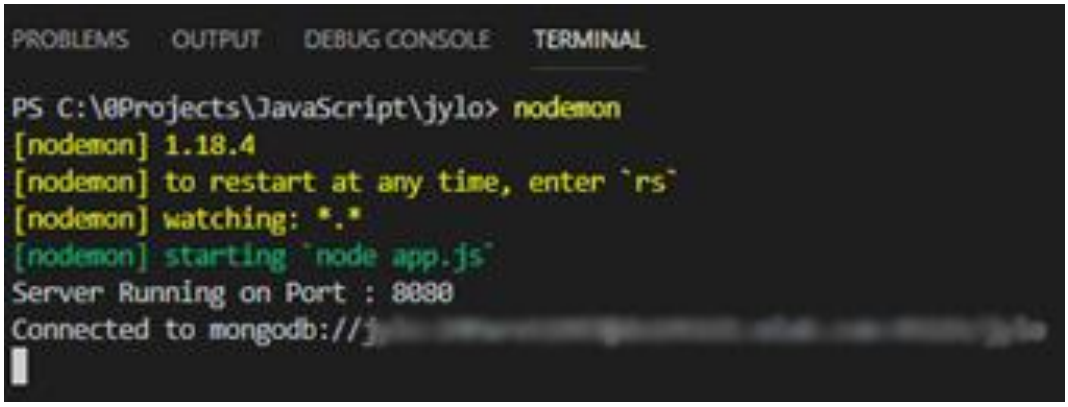


Gambar 15. Halaman Dashboard

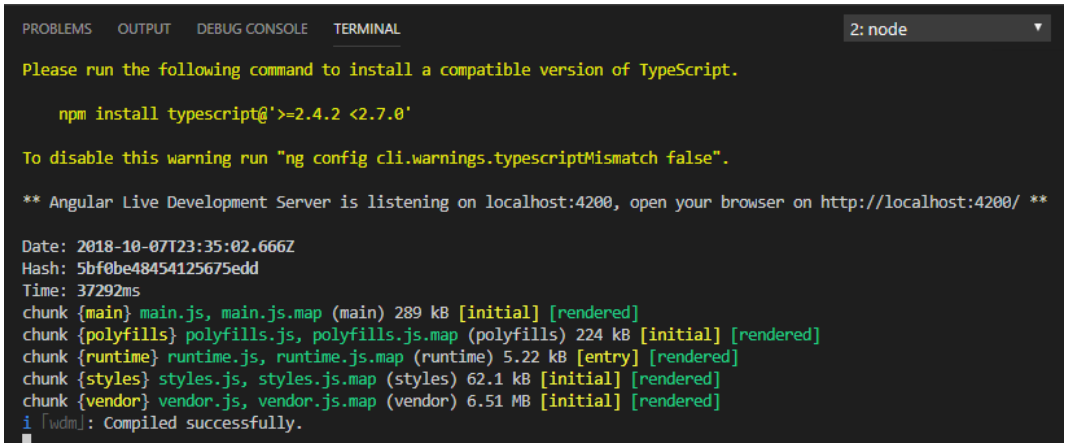
Hasil tampilan halaman *website* selengkapnya terdapat pada Lampiran 11.

2) *Test*

Tahap *test* atau pengujian pada fase *construction* sebatas pengujian-pengujian fungsional sistem dan fitur-fitur apakah terdapat *bug* atau cacat atau tidak. Pengujian otomatis dijalankan pada ketika menyimpan *file* atau kode program yang telah ditulis.



Gambar 16. Pengujian Fungsionalitas Sisi *Back-End*



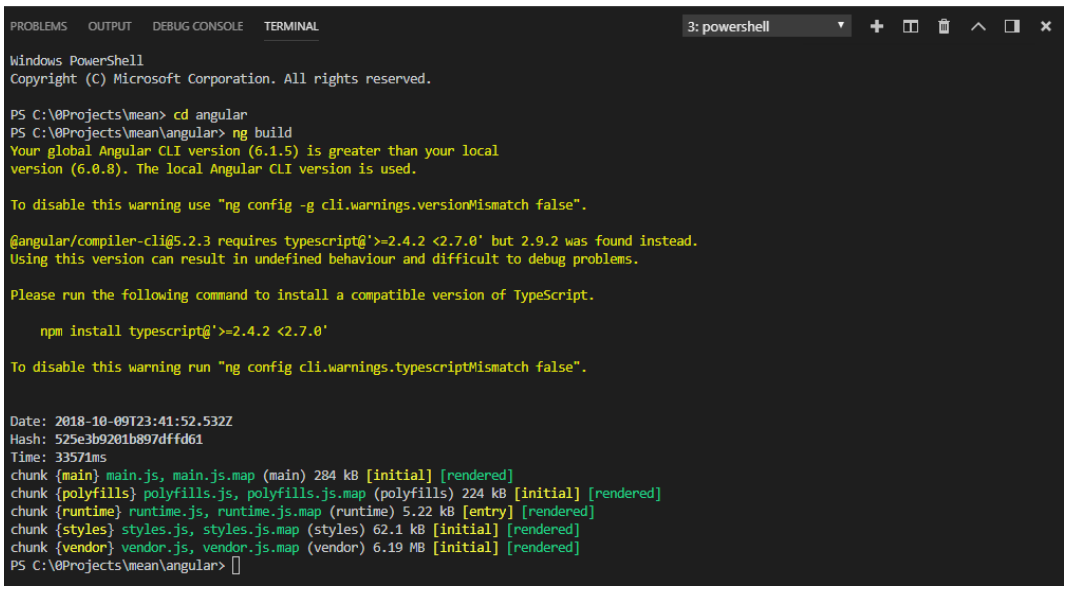
Gambar 17. Pengujian Fungsionalitas Sisi *Front-End*

d. Fase *Transition*

Fase *Transition* lebih berfokus pada tahap *deployment*, yaitu menyiapkan dan menyebarkan sistem yang telah dikembangkan serta pengujian dengan menggunakan Standar ISO/IEC 25010.

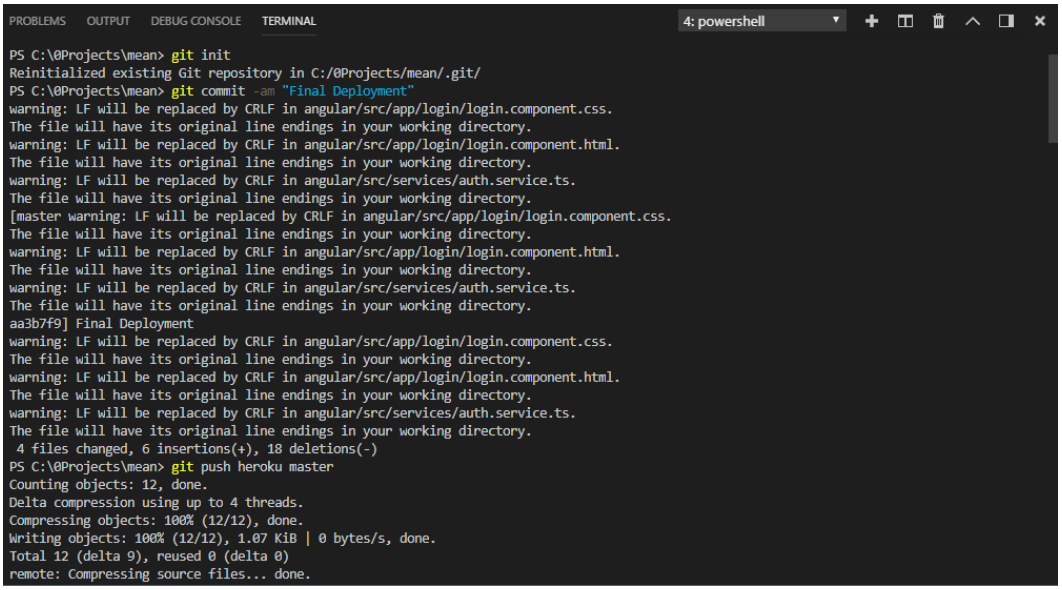
1) *Deployment*

Tahap *deployment* dimulai dengan melakukan *build* pada sisi *front-end* sistem. Proses *build* bagian *front-end* ditampilkan pada Gambar 18 berikut :



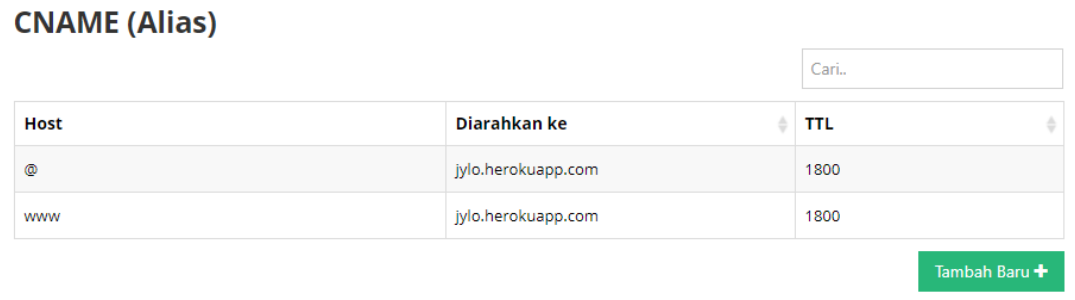
Gambar 18 . Proses *Build Angular*

Proses berikutnya yaitu mengunggah *source code* sistem yang telah dikembangkan ke *Cloud Platform Heroku*. Proses *upload* yang dilakukan menggunakan *Command Line Git*. Berikut proses mengunggah *source code* melalui *Git Command Line* :



Gambar 19. Proses Mengunggah *Source Code* ke *Cloud Platform Heroku*

Setelah *source code* berhasil diunggah, langkah selanjutnya yaitu mengkonfigurasi *domain website* agar pengguna mudah mengakses *website* dengan *domain* yang lebih singkat. Peneliti menggunakan salah satu jasa penyedia layanan *domain website* dan memberi nama *domain website* dengan nama : *www.fatih.online* . Konfigurasi *domain website* dapat dilihat pada Gambar 20 berikut :



Gambar 20. Konfigurasi *Domain Website*

2) *Test*

Tahap *test* atau pengujian pada fase *transition*, dilakukan dengan menggunakan Standar ISO 25010 yaitu pada aspek *functional suitability*, *usability*, dan *performance efficiency* yang akan dijabarkan pada sub bab bagian pengujian perangkat lunak.

2. Pengujian Perangkat Lunak

a. Pengujian *Functional Suitability*

Pengujian aspek *functional suitability* dilakukan oleh 2 (dua) orang ahli dalam bidang sistem informasi. Pengujian dilakukan untuk memvalidasi fungsional Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* berjalan dengan baik atau tidak. Berikut merupakan hasil pengujian *functional suitability* yang telah dilakukan oleh 2 (dua) orang ahli. Berikut adalah tabel hasil pengujian *functional suitability* :

Tabel 19. Hasil Uji *Functional Suitability*

No.	**Fitur**	**Skor Penguji**		**Jumlah**	**Skor Maksimal**
		1	**2**		
1.	Landing Page	1	1	2	2
2.	Halaman Beranda	1	1	2	2
3.	Halaman Profil	1	1	2	2
4.	Halaman Perusahaan	1	1	2	2
5.	Halaman Admin	1	1	2	2
6.	Login	1	1	2	2
7.	Logout	1	1	2	2

No.	Fitur	Skor Penguji		Jumlah	Skor Maksimal
		1	2		
8.	Identifikasi NIM	1	1	2	2
9.	Identifikasi Email	1	1	2	2
10.	Identifikasi Password	1	1	2	2
11.	*Update Privilege*	1	1	2	2
12.	Menampilkan Daftar Riwayat Tempat Praktik Industri	1	1	2	2
Total		**12**	**12**	**24**	**24**

Berdasarkan data diatas dapat diukur dengan menggunakan rumus *Feature Completeness* yang disajikan pada Rumus 1 sebagai berikut :

(Rumus 1. Rumus *Feature Completeness*)

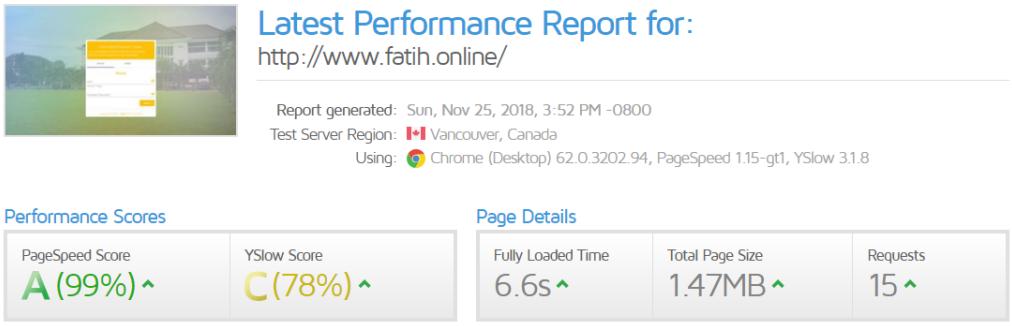
$$X = \frac{P}{I}$$

$$X = \frac{24}{24} \quad X = 1$$

Hasil dari pengujian *functional suitability* adalah X = 1 (satu) yang berarti bahwa seluruh fitur atau fungsi yang dikembangkan pada sistem dapat berjalan dengan baik. Berdasarkan hasil perhitungan dengan menggunakan rumus *feature completeness*, maka dapat disimpulkan bahwa pengujian *fungtional suitability* pada Sistem Informasi *Internship Program Tracer* berbasis *website* dengan *MEAN Stack* memiliki nilai “Baik”.

b. Pengujian *Performance Efficiency*

Pengujian *performance efficiency* Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* dilakukan menggunakan *software GTMetrix*. Hasil dari pengujian halaman login dapat dilihat pada Gambar berikut:



Gambar 21. Hasil Pengujian Halaman Login

Hasil pengujian *performance efficiency* selengkapnya dapat dilihat pada Lampiran 14. Daftar hasil pengujian *performance efficiency* ditampilkan pada Tabel 21.

Tabel 20. Hasil Pengujian *Performance Efficiency*

No	Halaman	Hasil Pengujian		
		Page Speed	YSlow	Waktu (detik)
1	Login & Register	99	78	5,5
2	Dashboard	98	78	5,0
3	Profile	99	78	7,2
4	Company	98	78	5,6
5	Admin	98	78	6,9
6	Admin/users	98	78	5,2
7	Admin/companies	98	78	6,9

No	Halaman	Hasil Pengujian		
		Page Speed	**YSlow**	**Waktu (detik)**
8	Admin/report	98	78	5,3
	Rata-rata	**98,25%**	**78%**	**5,95**

Berdasarkan pengujian *performance efficiency*, dapat diperoleh hasil *Page Speed* sebesar **98,25%** (***Grade* A**), *YSlow* sebesar **78%** (***Grade* C**), dan rata-rata waktu-load sebesar **5,95** detik.

c. Pengujian *Usability*

Pengujian aspek *usability* dilakukan oleh 15 mahasiswa yang telah melaksanakan Praktik Industri dan 5 (lima) mahasiswa yang akan melaksanakan praktik industri. Hasil pengujian *usability* dengan menggunakan *System Usability Scale* (SUS) sebagai berikut :

Tabel 21. Hasil Pengujian *Usability*

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai
1	4	2	3	1	4	2	3	3	3	2	**27**	**67,5**
2	2	0	3	1	3	3	4	3	4	3	**26**	**65**
3	3	4	4	2	3	3	4	4	1	1	**29**	**72,5**
4	3	3	3	3	3	3	3	3	3	3	**30**	**75**
5	3	2	4	3	1	2	3	3	3	2	**26**	**65**
6	2	2	4	1	4	3	3	3	1	2	**25**	**62,5**
7	2	3	4	4	3	3	4	4	4	3	**34**	**85**
8	3	3	3	1	3	3	4	3	3	3	**29**	**72,5**
9	4	3	3	2	3	3	4	3	3	3	**31**	**77,5**
10	3	2	2	1	3	2	2	3	3	2	**23**	**57,5**
11	3	3	4	4	3	3	4	3	4	3	**34**	**85**
12	3	3	3	3	4	3	4	3	4	3	**33**	**82,5**

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Jumlah	Nilai
13	3	3	3	3	4	3	4	3	4	4	**34**	**85**
14	3	1	2	2	2	3	3	2	3	2	**23**	**57,5**
15	3	3	4	4	3	3	3	3	4	3	**33**	**82,5**
16	3	3	4	3	3	3	3	4	4	2	**32**	**80**
17	3	3	1	1	3	2	3	3	3	1	**23**	**57,5**
18	3	3	3	1	3	3	3	3	3	3	**28**	**70**
19	3	2	3	3	2	3	3	3	4	2	**28**	**70**
20	3	3	3	1	3	3	3	3	3	3	**28**	**70**
Hasil												**72**

Berdasarkan hasil tabel di atas, maka diperoleh total skor rata-rata 72. Pengujian *usability* diperoleh dengan menggunakan rumus sebagai berikut:

System Usability Scale (SUS) memiliki 10 poin pertanyaan dengan 5 (lima) skala. Perhitungan *System Usability Scale* (SUS) sebagai berikut:

- Pada butir ganjil, skor responden dikurangi 1
- Pada butir genap, 5 dikurangi skor responden
- Jumlah 10 butir skor lalu hasilnya dikalikan 2,5

Jeff Sauro mengatakan “bahwa kualitas aspek *usability* dinyatakan baik jika memiliki skor di atas 68” (Sauro, 2011).

B. Pembahasan

Sistem Informasi *Internship Program Tracer* berbasis *Website* dengan *MEAN Stack* memberikan informasi tentang riwayat tempat Praktik Industri mahasiswa Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta. Sistem informasi ini dikembangkan menggunakan

ExpressJS sebagai *back-end* dan *Angular* sebagai *front-end* dengan menggunakan *MongoDB* sebagai *database*. Pengembangan sistem informasi ini dikembangkan menggunakan metode *Agile Unified Process* (AUP) dimana pada AUP ini terdapat 4 (empat) fase pengembangan *inception*, *elaboration*, *construction*, dan *transition*. Pada tiap fase terdapat tahapan-tahapan dimulai dari *model*, *implementation*, *test*, *deployment*, *configuration management*, *project management*, dan *environment*.

Fase *inception* berfokus pada pembuatan model proses bisnis (*business modelling*) yang dibutuhkan oleh sistem. Hasil dari *business modeling* adalah kebutuhan sistem yang dikembangkan (*system requirements*). Fase *elaboration* berfokus pada analisa dan desain sistem (*analysis & design*). Proses analisa dan desain sistem merupakan tindaklanjut hasil dari *business modeling* dan *system requirements*. Pada fase *construction* berfokus mentransformasikan tahapan *Model* UML yang dihasilkan dari fase *elaboration* ke dalam pembuatan kode program sistem. Pada fase *elaboration* berfokus pada tahapan *deployment*. Pada fase ini sistem yang dikembangkan siap didistribusikan kepada pengguna serta dilakukan pengujian Standar ISO/IEC 25010.

Berikut hasil pengujian terakhir aplikasi berdasarkan ISO/IEC 25010 yang disajikan pada Tabel 22:

Tabel 22. Hasil Pengujian Sistem

No.	Aspek	Hasil	Kategori
1.	*Fuctional Suitability*	Pengukuran dengan menggunakan rumus *feature completeness* mendapatkan skor	**Baik**

No.	Aspek	Hasil	Kategori
		X = 1, yang artinya seluruh fitur atau fungsi pada sistem berjalan dengan baik	
2.	*Usability*	Pengujian *Usability* menggunakan instrumen *System Usability Scale* (SUS) dengan skor akhir **72**	**Baik**
3.	*Performance Efficiency*	Hasil pengujian *PageSpeed* mendapatkan presentase sebesar **98,25%** dengan rata-rata waktu *load* sebesar **5,95 detik**.	**Baik**