#### BAB 3

## PELAKSANAAN KERJA MAGANG

## 3.1 Kedudukan dan Organisasi

Dalam pelaksanaan kerja magang di ERV Software, penulis memegang jabatan sebagai *programmer* atau *web developer* di divisi *Programmer*, dengan supervisor dari Bapak Emi Sastra, selaku direktur ERV Software. Bapak Emi berperan dalam memberikan instruksi mengenai aplikasi yang perlu dibuat dan dirancang serta memberikan bimbingan dalam pembuatan *website* data siswa.

## **3.2** Tugas dan Uraian Kerja Magang

Selama pelaksanaan kerja magang, tugas dan tanggung jawab yang diberikan adalah membuat aplikasi data siswa untuk admin sekolah. Adapun uraian tugas yang dilakukan setiap minggu yang terdapat pada Tabel 3.1.

Minggu	Uraian Pekerjaan
ke-	
1	Pengenalan framework Laravel, mengerjakan proyek mini: Data
	Mahasiswa, membuat tabel mahasiswa di database, membuat
	fungsi import data mahasiswa, membuat formulir create dan edit
	mahasiswa, implementasi Daterangepicker dan button dropdown,
	membuat partial template blade pada formulir
2	Membuat tabel master jurusan, implementasi fitur sorting data
	tabel, searching dan filter data mahasiswa dengan JavaScript
3	Mengerjakan proyek utama: Website Data Siswa, merancang
	struktur database, membuat file migration dan file seeder,
	membuat halaman utama
4	Membuat fungsi import data siswa dan guru dari berkas CSV,
	implementasi pagination dan fitur rows per page dengan Ajax,
	implementasi fitur sorting data tabel dengan JavaScript, mengubah
	struktur <i>database</i> untuk nilai siswa
5	Membuat formulir <i>create</i> dan edit untuk data siswa dan data guru,
	membuat validasi formulir create dan edit untuk data siswa dan
	data guru
6	Membuat <i>flowchart</i> program, merancang ulang struktur <i>database</i> ,
	mengganti dan menambah file migration

Tabel 3.1 Jadwal kerja magang

Minggu ke-	Uraian Pekerjaan
7	Mengubah struktur <i>database</i> , menentukan <i>eloquent relationship</i> model, desain tampilan halaman utama, membuat menu <i>side navigation</i>
8	Memperbaiki fungsi <i>import</i> data guru dan menghapus fungsi <i>import</i> data siswa, memperbaiki fungsi <i>pagination</i> tabel data guru dengan Ajax, implementasi <i>live search</i> dengan Ajax
9	Memperbaiki formulir <i>create</i> dan edit guru, membuat fungsi hapus guru
10	Validasi ulang formulir <i>create</i> dan edit guru, membuat <i>alert box</i> pada halaman daftar guru dan daftar siswa ketika berhasil <i>create</i> atau <i>update</i> data guru atau data siswa, menambahkan desain halaman utama, memperbaiki fungsi <i>pagination</i> untuk tabel data siswa
11	Mengubah struktur database untuk data siswa, memperbaiki formulir create siswa dan validasinya
12	Integrasi back-end create siswa antar tabel di database, membuat fungsi yang mengambil data siswa dari database ke halaman daftar siswa
13	Melakukan konsultasi proyek, memperbaiki <i>bug</i> pada fungsi <i>pagination</i> tabel daftar guru dan siswa, membuat halaman penilaian siswa beserta <i>back-end</i> fitur penilaian siswa
14	Memperbaiki formulir edit siswa dan validasinya, integrasi antara data siswa setelah diedit dengan data mata pelajaran dan nilai siswa, membuat fungsi hapus siswa dan data siswa yang bersangkutan

## Tabel 3.2 Jadwal kerja magang (lanjutan)

### 3.3 Proses Pelaksanaan

Untuk pelaksanaan pembuatan proyek *website* data siswa, Bapak Emi selaku direktur ERV Software memberikan daftar *requirement* program yang harus dibuat serta teknologi yang akan digunakan dalam membuat proyek. Proses pelaksanaan pembuatan proyek hingga implementasinya akan dijabarkan secara terperinci dalam subbab-subbab berikut.

### 3.3.1 Requirement

Dalam membangun *website* data siswa, terdapat *requirement* yang harus dipenuhi, yaitu:

- A. Sistem dapat menambahkan, mengedit dan menghapus data guru yang meliputi identitas guru dan mata pelajaran yang ingin diampu.
- B. Sistem dapat menambahkan, mengedit dan menghapus data siswa yang meliputi identitas siswa, tingkat instansi, kelas dan ruangan kelas, serta mata pelajaran yang diambil terkait tingkat instansi dan jurusan yang diambil khusus siswa SMA.
- C. Sistem dapat memberikan penilaian terhadap mata pelajaran yang siswa terima meliputi nilai tugas, nilai UTS dan nilai UAS, serta nilai akhir yang terhitung secara otomatis dan merupakan hasil dari 30% nilai tugas, 30% nilai UTS dan 40% nilai UAS.

### **3.3.2** Framework dan Database yang Digunakan

Pembuatan *website* data siswa menggunakan bahasa markah standar HTML dan CSS serta bahasa pemrograman JavaScript untuk merancang desain *front-end website*. JavaScript, yang merupakan salah satu dari banyak bahasa pemrograman lain yang masih digunakan hingga saat ini, dikenal sebagai teknologi yang membantu halaman web menjadi bersifat dinamik atau interaktif (Toal and Dionisio, 2009). Alasan menggunakan JavaScript adalah dapat dijalankan di semua perangkat komputer tanpa perlu instalasi tambahan dan JavaScript terintegrasi dengan konten halaman web lebih baik dibandingkan bahasa pemrograman lain saat ini sehingga memungkinkan JavaScript untuk melakukan segala bentuk interaksi maupun animasi di halaman web murni tanpa *plug-ins* atau *extensions*.

Untuk menangani kerja sistem yang berhubungan dengan server atau sisi *back-end* program dari *website* data siswa menggunakan *framework* Laravel. Laravel adalah pengembangan *website* berbasis MVP (*minimum viable product*) yang ditulis dalam PHP yang dirancang untuk meningkatkan kualitas perangkat lunak dengan mengurangi biaya pengembangan awal dan biaya pemeliharaan, serta untuk meningkatkan pengalaman bekerja dengan aplikasi dengan menyediakan sintaksis yang ekspresif, jelas, dan menghemat waktu (Yudhanto and Prasetyo, 2019). Alasan menggunakan *framework* Laravel adalah penulisan *code* yang lebih simpel dan sederhana karena dirancang untuk memudahkan dan mempercepat proses pengembangan web.

Selain itu, terdapat *database* yang digunakan dalam pembuatan *website* data siswa yaitu MySQL. MySQL *database server* adalah RDBMS (Relational Database Management System) yang dapat menangani data yang bervolume besar dan mampu mengirim dan menerima data dengan sangat cepat tanpa menuntut *resource* yang besar (Wahana Komputer, 2010). Alasan menggunakan MySQL adalah performanya yang tinggi dan sederhana, kemampuannya untuk menyimpan data berkapasitas besar, serta bebas digunakan oleh perseorangan atau instansi tanpa harus membeli atau membayar kepada pembuatnya.

#### 3.3.3 Flowchart

Alur proses impor guru dimulai dari munculnya halaman modal di mana user dapat memilih berkas bertipe CSV yang berisi data guru sesuai dengan atribut tabel guru di *database*. Saat proses impor berjalan, sistem akan membaca berkas tersebut dan sebelum menyimpannya sistem akan memeriksa apakah *teacher id* dari baris data yang dibaca sudah terdapat di *database*. Apabila *teacher id* sudah terdapat di *database*, maka sistem akan melompati data tersebut dan lanjut membaca data berikutnya, dan apabila *teacher id* belum terdapat di *database*, maka sistem akan menyimpan data tersebut ke *database* sebagai data baru. Setelah selesai, maka halaman modal akan tertutup secara otomatis. *Flowchart* impor guru dapat dilihat pada Gambar 3.1.



Gambar 3.1 Flowchart impor guru

Alur proses *create* guru dimulai dari munculnya formulir kosong data guru yang dapat diisi oleh *user*. Di akhir formulir terdapat tombol *submit* untuk mengirim isi formulir. Sistem akan mengecek isi formulir sebelum dikirim ke server, apabila semua formulir yang wajib diisi telah terisi maka sistem akan mengirim isi formulir ke server untuk menyimpan data guru yang baru. *Flowchart* dari proses *create* guru dapat dilihat pada Gambar 3.2.



Gambar 3.2 Flowchart create guru

Alur proses edit guru dimulai dari munculnya formulir yang telah berisikan data guru yang ingin diedit dan dapat diubah oleh *user*. Di akhir formulir terdapat tombol *save* untuk mengirim isi formulir. Sistem akan mengecek isi formulir sebelum dikirim ke server, apabila semua formulir yang wajib diisi telah terisi maka sistem akan mengirim isi formulir ke server untuk memperbarui data guru yang diubah. *Flowchart* dari proses edit guru dapat dilihat pada Gambar 3.3.



Gambar 3.3 Flowchart edit guru

Alur proses hapus guru dimulai dari mengirim *id* guru yang ingin dihapus ke server. Server akan mengambil data guru dengan *id* yang dikirim kemudian menghapus data yang ditemukan. *Flowchart* hapus guru dapat dilihat seperti pada Gambar 3.4.



Gambar 3.4 Flowchart hapus guru

Alur proses *create* siswa dimulai dari munculnya formulir kosong data siswa yang dapat diisi oleh *user*. Di akhir formulir terdapat tombol *submit* untuk mengirim isi formulir. Sistem akan mengecek isi formulir sebelum dikirim ke server, apabila semua formulir yang wajib diisi telah terisi maka sistem akan mengirim isi formulir ke server untuk menyimpan data siswa yang baru. *Flowchart* dari proses *create* siswa dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart create siswa

Alur proses edit siswa dimulai dari munculnya formulir yang telah berisikan data siswa yang ingin diedit dan dapat diubah oleh *user*. Di akhir formulir terdapat tombol *save* untuk mengirim isi formulir. Sistem akan mengecek isi formulir sebelum dikirim ke server, apabila semua formulir yang wajib diisi telah terisi maka sistem akan mengirim isi formulir ke server untuk memperbarui data siswa yang diubah. Apabila data instansi atau kelas siswa mengalami perubahan, maka data semua mata pelajaran siswa yang lama beserta nilainya akan dihapus dengan data mata pelajaran yang baru dan belum memiliki nilai. *Flowchart* dari proses edit siswa dapat dilihat pada Gambar 3.6.



Gambar 3.6 Flowchart edit siswa

Alur proses hapus siswa dimulai dari mengirim *id* siswa yang ingin dihapus ke server. Server akan mengambil data siswa dengan *id* yang dikirim. Kemudian server akan mengambil semua data yang berhubungan dengan data siswa seperti tingkat, kelas, ruangan kelas, mata pelajaran yang diterima dan nilai dari mata pelajaran tersebut. Setelah itu server akan menghapus semua data yang ditemukan. *Flowchart* hapus siswa dapat dilihat seperti pada Gambar 3.7.



Gambar 3.7 Flowchart hapus siswa

Alur proses penilaian siswa dapat dilakukan dengan mengisi atau mengubah nilai tiap mata pelajaran yang diterima oleh siswa di formulir penilaian yang berada di halaman detail siswa. Di akhir formulir terdapat tombol *save* yang secara langsung akan menyimpan perubahan sesuai isi formulir penilaian ke *database*. *Flowchart* dari proses penilaian siswa dapat dilihat pada Gambar 3.8.



Gambar 3.8 Flowchart penilaian siswa

#### 3.3.4 Entity Relationship Diagram

Dalam pembuatan proyek *website* data siswa, terjadi sedikitnya dua kali perubahan besar struktur *database* yang mengharuskan proses *development* terhenti untuk sementara waktu karena perlu dilakukan perubahan implementasi pada sisi *back-end* juga. Rancangan struktur *database* terakhir yang hingga saat ini masih digunakan direpresentasikan dalam bentuk Entity Relationship Diagram (ERD) yang dapat dilihat pada Gambar 3.9 dengan jumlah tabel adalah 12.



Gambar 3.9 Entity relationship diagram website data siswa

## 3.3.5 Implementasi

Dari semua poin-poin *requirement* yang telah dijabarkan, hanya tiga poin yang dapat diselesaikan hingga saat ini antara lain sistem CRUD guru, sistem CRUD siswa dan sistem penilaian terhadap mata pelajaran yang diterima oleh siswa. Berikut merupakan hasil implementasi *website* data siswa dari fitur-fitur yang sudah berhasil diselesaikan.

## A. Halaman Utama dan Menu Side Navigation

Gambar 3.10 merupakan tampilan dari halaman utama dan Gambar 3.11 merupakan bentuk tampilan menu *side navigation* ketika *button* menu di pojok kiri atas diklik.



Gambar 3.10 Halaman utama



Gambar 3.11 Menu side navigation

# B. Tampilan Impor Guru

Fitur impor guru hanya digunakan sebagai *shortcut* untuk mengimpor data *dummy* guru dari berkas CSV. Impor guru akan menampilkan sebuah modal seperti terlihat pada Gambar 3.12.



Gambar 3.12 Tampilan impor guru

## C. Halaman Daftar Guru

Halaman daftar guru akan menampilkan semua data guru yang tersimpan di *database*. Data yang ditampilkan meliputi nama, *teacher id*, alamat, kota, tanggal lahir, nomor telepon, tanggal masuk dan tanggal keluar. Apabila belum ada data guru yang tersimpan, maka halaman akan menampilkan pesan "Tidak ditemukan data guru" seperti terlihat pada Gambar 3.13, sedangkan Gambar 3.14 menampilkan semua data guru yang tersimpan di *database*.

DATA SISWA PROJECT		Import Teacher
Number of rows: 20 Tidak ditemukan data guru	DAFTAR GURU	Search Nama.

Gambar 3.13 Halaman daftar guru saat kosong

DAFTAR GURU								
1 2 3 4	•					•	<b>Q</b> Search Nama	8
NAMA IT	TEACHER ID 11	ALAMAT IT	κοτα μη	TANGGAL LAHIR J†	NO TELP J†	TANGGAL MASUK J†	TANGGAL KELUAR IT	ACTION
Amir Salim	10226	JI MH Thamrin Kav 28-30 Plaza Indonesia Lt 2/108	Jakarta	30-04-1970	0838-555-977	03-02-2000		<b>1</b>
Ratu Dian Yuwono	11844	JI TB Simatupang Kavling 89,Tanjung Barat	Papua Barat	26-07-1976	0855-555-927	11-04-2000		<b>1</b>
fansur Deddy Widjaja	11996	JI Kb Kacang Grand Indonesia Shopping Town East Mall Lt Ground 30	Jakarta	02-09-1980	0811-555-404	26-03-2007		<b>1</b>
Dewi Sasmita	14513	JI HR Rasuna Said Kav 2-3 BI X-5 Menara Kadin Indonesia BI X-5 Lt	Jakarta	18-08-1975	0819-555-965	26-05-2001		1
Yuliani Pranoto	15472	JI HR Rasuna Said Kav 2-3 BI X-5 Menara Kadin Indonesia Lt 12 Uni	Jakarta	17-04-1971	0814-555-461	08-10-2001		<b>1</b>
Surya Jayadi	18427	JI Ki Asnawi 18 RT 002/07,Sukarasa	Tangerang	19-01-1989	0898-555-124	01-11-2012		<b>1</b>
Nur Tanuwidjaja	14155	JI Ki Samaun 10-A,Sukasari	Tangerang	07-12-1987	0852-555-068	17-09-2008		<b>1</b>
Nurul Sudino	10112	JI Lauk Emas 4	Bandung	28-11-1971	0814-555-922	09-10-2000		1
Ali Sudirman	13636	JI Wastukencana 15	Bandung	11-11-1979	0838-555-929	05-10-2002		1
Rahman Hardja	16583	JI Kb Kacang Mall A Grand Indonesia BI B Lt 2/1	Jakarta	30-10-1989	0878-555-279	29-07-2014		<b>1</b>
Raja Iwan Sudomo	14032	JI Seturan 189-C RT 007/16,Condongcatur	Depok	17-12-1984	0818-555-948	24-10-2009		<b>1</b>
Sri Susanto	10115	JI Pringgodani 9,Caturnunggal	Depok	09-01-1987	0878-555-711	24-06-2010		1
Aditya Purnama	12419	JI Malioboro 52-58 Yogyakarta	Yogyakarta	16-05-1991	0897-555-066	08-01-2018		Ø 🖄
Iman Suryadi	18520	JI Dr Sutomo 16	Semarang	17-10-1994	0878-555-788	18-05-2020		<b>Ø</b>
Muhammad	14050	JI Ki Asnawi 79 96 RT 001/07,Sukarasa	Tangerang	20-03-1974	0853-555-621	12-10-2002		<b>1</b>
Abdul	16411	JI Mrican Baru Ruko Selatan B 11-12,Caturnunggal	Depok	05-06-1975	0856-555-591	21-03-2005		Ø 🖞
Tan Adi Kusuma	14038	JI Nusa Indah 37-A	Yogyakarta	11-04-1985	0878-555-560	30-06-2010		21
Hasan Putra	12547	Gg Purisari 1	Yogyakarta	06-07-1993	0878-555-036	13-07-2019		<b>1</b>
Ratu Dewi Indrajaya	14752	JI Puspogiwang I 40	Semarang	01-07-1981	0898-555-075	05-03-2011		<b>1</b>
Mahmud	14267	Ji Alun Alun Utr	Yogyakarta	01-02-1977	0878-555-629	24-02-2003		1

Gambar 3.14 Halaman daftar guru saat terdapat data

## **D.** Halaman Create Guru

Halaman *create* guru menampilkan sebuah *form* kosong yang dapat diisi untuk membuat data guru baru yang ingin ditambahkan. *Form field* yang wajib diisi memiliki simbol *asterisk* berwarna merah di ujung kanan *input*. Tampilan *form* dapat dilihat pada Gambar 3.15.

Tambah Guru	
Nama	
Masukan Nama	•
Teacher Id	
Masukan ID	•
Alamat	
Masukan Alamat Tempat Tinggal	•
Kota	
Masukan Kota Lahir	•
Tanggal Lahir	
Masukan Tanggal Lahir	•
Nomor Telepon	
Masukan No Telp	•
Tingkat Instansi	
Pilih Tingkat Instansi	\$
Mata Pelajaran yang Diampu	
Pilih Mata Pelajaran	\$
Tanggal Masuk	_
Masukan Tanggal Masuk	•
Tanggal Keluar	
Masukan Tanggal Keluar	
Submit	icel

Gambar 3.15 Halaman *create* guru

## E. Halaman edit guru

Halaman edit guru menggunakan *form* yang sama dengan yang digunakan untuk *create* guru. Letak perbedaannya adalah halaman ini mengambil *id* dari guru yang ingin diedit sehingga *form* tidak lagi kosong dan sudah berisi data guru tersebut yang dapat diubah-ubah. *Form field* yang memiliki simbol *asterisk* berwarna merah di ujung kanan *input* tidak boleh kosong. Tampilan dapat dilihat pada Gambar 3.16.

Edit Guru	
Nama	
Amir Salim	*
Teacher Id	
10226	*
Alamat	
JI MH Thamrin Kav 28-30 Plaza Indonesia Lt 2/108	*
Kota	
Jakarta	*
Tanggal Lahir	
1970-04-30	*
Nomor Telepon	
0838-555-977	*
Tingkat Instansi	
тк	\$
Mata Pelajaran yang Diampu	
Berhitung	\$
Tanggal Masuk	
2000-02-03	*
Tanggal Keluar	
Masukan Tanggal Keluar	
Saue	

Gambar 3.16 Halaman edit guru

## F. Halaman Daftar Siswa

Halaman daftar siswa akan menampilkan semua data siswa yang tersimpan di *database*. Data yang ditampilkan meliputi nama, *student id*, alamat, kota, tanggal lahir, nomor telepon, tanggal masuk dan tanggal keluar. Apabila belum ada data siswa yang tersimpan, maka halaman akan menampilkan pesan "Tidak ditemukan data murid" seperti terlihat pada Gambar 3.17, sedangkan Gambar 3.18 menampilkan semua data siswa yang tersimpan di *database*.



Gambar 3.17 Halaman daftar siswa saat kosong

<b>D</b> 9	NTA SISWA PROJECT								Import Teacher
	DAFTAR SISWA								
								Q Search Nama	<u>a</u>
	Eko Widya Atmadja	12345	JI Kwitang Raya 8 Ged Senatama Lt 3,Kwitang	Jakarta	31-01-2000	0212303641	14-12-2020	TANGGAL KELUAR IT	

Gambar 3.18 Halaman daftar siswa saat terdapat data

## G. Halaman Create Siswa

Halaman *create* siswa menampilkan sebuah *form* kosong yang dapat diisi untuk membuat data siswa baru yang ingin ditambahkan. *Form field* yang wajib diisi memiliki simbol *asterisk* berwarna merah di ujung kanan *input*. Untuk *form field* jurusan memiliki kewajiban diisi bersifat dinamik, yang artinya hanya akan wajib diisi dan simbol *asterisk* berwarna merah akan muncul hanya jika pilihan tingkat dipilih untuk SMA. Tampilan *form* dapat dilihat pada Gambar 3.19.

Tambah Siswa	
Nama	
Masukan Nama	•
Student Id	
Masukan NISN	
Alamat	
Masukan Alamat Tempat Tinggal	•
Kota	
Masukan Kota Lahir	*
Tanggal Lahir	
Masukan Tanggal Lahir	*
Nomor Telepon	
Masukan No Telp	
Tingkat	
Pilih Tingkat	\$
Kelas	
Pilih Kelas	\$
Jurusan	
Pilih Jurusan	*
Ruang Kelas	
Pilih Ruang Kelas	*
Tanggal Masuk	
Masukan Tanggal Masuk	•
Tanggal Keluar	
Masukan Tanggal Lulus	

Gambar 3.19 Halaman *create* siswa

## H. Halaman Detail Siswa dan Penilaian

Halaman detail siswa menampilkan informasi siswa yang belum ditampilkan pada halaman daftar siswa, yakni meliputi tingkat, kelas dan wali kelas, serta formulir penilaian terhadap semua mata pelajaran yang diterima oleh siswa tersebut. Gambar 3.20 merupakan tampilan dari halaman detail siswa.

DATA SISWA				Import Teacher
DAFTAR SISWA>>DETAIL SISWA Nama : Eko Widya Atmadja [Edit] Student Id: 12345 Tingkat : SMA Kelas : XII IPA 1 Wali Kelas : Saif Sugiarto				
MATA PELAJARAN	TUGAS	UTS	UAS	NILAI AKHIR
Matematika				
Bahasa Indonesia				
Bahasa inggris				
Pendidikan Kewarganegaraan				
Agama				
Penjasorkes				
ТІК				
Muatan Lokal				
Seni Budaya				
Fisika				
Biologi				
Kimia				
Save				

Gambar 3.20 Halaman detail siswa dan penilaian

## I. Halaman Edit Siswa

Halaman edit siswa menggunakan *form* yang sama dengan yang digunakan untuk *create* siswa. Letak perbedaannya adalah halaman ini mengambil *id* dari siswa yang ingin diedit sehingga *form* tidak lagi kosong dan sudah berisi data siswa tersebut yang dapat diubah-ubah. *Form field* yang memiliki simbol *asterisk* berwarna merah di ujung kanan *input* tidak boleh kosong. Untuk *form field* jurusan memiliki kewajiban diisi bersifat dinamik, yang artinya hanya akan wajib diisi dan simbol *asterisk* berwarna merah akan muncul hanya jika pilihan tingkat dipilih untuk SMA. Tampilan dapat dilihat pada Gambar 3.21.

Edit Siswa		
Nama		
Eko Widya Atmadja		
Student Id		
12345		
Alamat		
Jl Kwitang Raya 8 Ged Senatama Lt 3,Kwitang	•	
Kota		
Jakarta	*	
Tanggal Lahir		
2000-01-31		
Nomor Telepon		
0212303641	•	
Tingkat		
SMA	*	
Kelas		
12	\$	
Jurusan		
Pilih Jurusan	*	
Ruang Kelas		
XII IPA 1	*	
Tanggal Masuk		
2020-07-28	•	
Tanggal Keluar		
Masukan Tanggal Lulus		

Gambar 3.21 Halaman edit siswa

## 3.3.6 Kendala yang Ditemukan

Selama bekerja di ERV Software, ditemukan adanya beberapa kendala yang menghambat proses pembuatan proyek *website* data siswa. Kendala-kendala yang ditemui secara umum sebagai berikut:

- a. Penggunaan *framework* Laravel yang belum fasih.
- Kurangnya kemahiran dalam penggunaan bahasa pemrograman JavaScript.
- c. Perubahan struktur *database* pada saat proses *development* sehingga perlu dilakukan penyesuaian *code* untuk *back-end* berulang kali.
- d. Komunikasi yang gagal dengan direktur sehingga instruksi dan *requirement* tidak diterima dengan jelas.
- e. Banyak *bug* pada proses pembuatan sisi *back-end* website.
- f. Proses *debugging* di sisi *back-end* yang sulit dilakukan.

## 3.3.7 Solusi atas Kendala yang Ditemukan

Dalam menghadapi kendala-kendala yang telah dijabarkan, terdapat beberapa solusi yang dapat digunakan untuk mengatasinya. Hal pertama yang dapat dilakukan ketika menghadapi *bug* atau eror dalam aplikasi yaitu mencari solusi di internet. Internet menyediakan banyak informasi mengenai masalah umum yang sering dialami oleh *programmer* lainnya.

Namun tidak selamanya internet memiliki semua jawaban atas masalah yang dihadapi. Ketika solusi tidak dapat ditemukan di internet, bertanya kepada rekan kerja atau teman yang memiliki pengalaman di *framework* yang digunakan dapat menjadi pilihan berikutnya. Apabila masih belum ditemukan solusinya, pilihan terakhir adalah untuk berkonsultasi langsung kepada supervisor, dalam hal ini yaitu Bapak Emi mengenai kendala yang dihadapi sehingga dapat dicari solusi bersama baik dengan cara mengubah logika pemrograman atau dengan mengubah *requirement* yang ada.