

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

A. Analisis Kebutuhan

Analisis kebutuhan diawali dengan melakukan observasi terhadap proses administrasi dan wawancara dengan pemilik bimbingan belajar Kindy Educa. Berdasarkan hasil wawancara dan observasi, peneliti mengetahui sistem administrasi yang saat ini diterapkan.

Sistem tersebut hanya berupa pendaftaran dengan kuitansi, sistem ini menyebabkan tidak adanya data yang pasti yang bisa digunakan untuk merekap data-data siswa dan pengajar. Jika ada pengajar yang mendapatkan siswa baru, pengajar harus mencari rumah siswa dengan ditemani oleh pengurus bimbingan yang terkadang berhalangan untuk menemani sehingga pengajar harus mencari sendiri lokasi rumah siswa.

Peneliti menyimpulkan bahwa perlu adanya sistem yang lebih tertata sehingga pengurus dapat mendata pengajar dan siswanya sehingga dapat dilakukan pengecekan dengan akurat. Sistem yang baru ini juga diharapkan dapat memudahkan pengajar untuk menemukan lokasi rumah siswa sehingga ketika pihak bimbel tidak dapat mengantar, pengajar dapat dengan mandiri mencari rumah siswa. Dari penjabaran tersebut dapat ditentukan untuk sasaran pengguna dari sistem tersebut adalah admin, tentor, dan siswa dari Lembaga Bimbingan Belajar Kindy Educa.

1. Analisis Kebutuhan Fungsional

Kebutuhan fungsional dari sistem yang akan dikembangkan untuk Bimbingan Belajar Kindy Educa adalah:

a. Umum (Dimiliki oleh Semua User)

Fungsi umum dimiliki oleh semua user yang sudah terdaftar baik admin, siswa, maupun pengajar. Terkecuali fungsi registrasi yang bisa diakses oleh semua pengguna yang mengakses sistem.

- 1) Registrasi Pengajar
- 2) Registrasi Siswa
- 3) *Login dan Logout*
- 4) Ubah Data Profil
- 5) Ubah Lokasi
- 6) Ubah *Password*

b. Administrator

Administrator merupakan pengelola bimbel yang memiliki akses untuk melakukan verifikasi pada user yang mendaftar, dan juga menyetujui permintaan bimbingan oleh siswa.

- 1) Menampilkan daftar mata pelajaran yang tersedia
- 2) Menambah mata pelajaran
- 3) Mengubah mata pelajaran
- 4) Menampilkan daftar siswa
- 5) Menampilkan jumlah siswa yang terdaftar
- 6) Menampilkan daftar pengajar
- 7) Menampilkan jumlah pengajar yang terdaftar
- 8) Menampilkan detail profil pengguna
- 9) Melakukan verifikasi pengguna yang mendaftar
- 10) Menampilkan daftar sesi bimbingan
- 11) Menampilkan jumlah sesi bimbingan

- 12) Menampilkan detail sesi bimbingan
- 13) Memberikan persetujuan pada sesi bimbingan

c. Pengajar

Pengajar merupakan pengguna yang dapat menerima permintaan sesi bimbingan dan dapat menyetujui maupun menolak permintaan bimbingan tersebut.

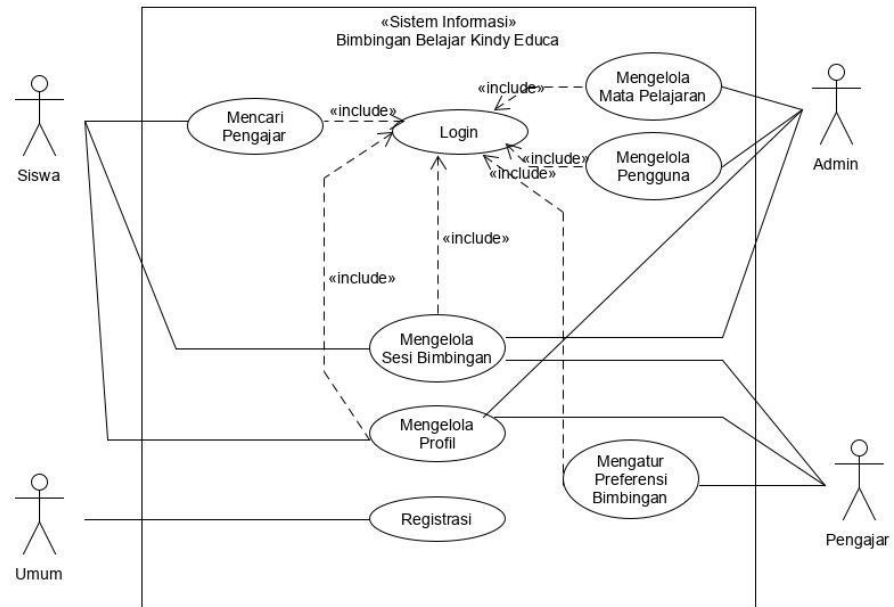
- 1) Melihat daftar sesi bimbingan
- 2) Melihat detail sesi bimbingan dari sisi pengajar
- 3) Menerima atau menolak permintaan sesi bimbingan
- 4) Mengubah preferensi sesi bimbingan
- 5) Menampilkan detail profil siswa setelah menerima permintaan bimbingan

d. Siswa

Siswa merupakan pengguna yang dapat mencari pengajar yang sesuai dengan permintaan, dan dapat mengajukan permintaan sesi bimbingan kepada pengajar tersebut.

- 1) Mencari pengajar yang sesuai
- 2) Mengajukan permintaan bimbingan kepada pengajar
- 3) Melihat detail sesi bimbingan dari sisi siswa
- 4) Menampilkan daftar sesi bimbingan yang diajukan
- 5) Menampilkan detail profil pengajar

Setelah dijabarkan kebutuhan yang diperlukan dari masing-masing kategori pengguna, dibuat model analisis menggunakan UML dengan bentuk diagram *use case* yang menggambarkan kebutuhan pengguna pada sistem. Diagram *use case* dapat dilihat di bawah ini:



Gambar 4. Use Case Diagram

Terdapat empat aktor yang dirancang untuk sistem ini, admin, pengajar, siswa, dan umum. Umum yang merupakan pengguna baru hanya bisa melakukan pendaftaran pada sistem, sedangkan aktor lain dapat melakukan aktifitas yang lain setelah melakukan *login* pada sistem.

2. Analisis Kebutuhan Perangkat Keras dan Perangkat Lunak

Untuk dapat menjalankan sistem ini di *server*, server harus memiliki fitur PHP, Apache, dan MySQL. Untuk mengakses sistem yang dikembangkan, pengguna hanya membutuhkan perangkat keras yang terkoneksi dengan internet yang di dalamnya terdapat *browser* yang bisa memuat *website* modern.

Sedangkan untuk mengembangkan sistem ini dibutuhkan perangkat berupa:

- a. Perangkat komputer (PC atau Laptop)

- b. XAMPP (Apache, PHP, dan MySQL)
- c. *Browser*
- d. *Text Editor/IDE*

B. Desain

Tahapan desain dilakukan setelah tahap analisis. Pada tahap desain dilakukan perancangan berdasarkan kebutuhan yang telah ditentukan pada tahap analisis. Perancangan dilakukan agar saat tahap *coding* atau implementasi, terdapat rancangan yang dapat dijadikan sebagai acuan sehingga sistem yang dikembangkan tetap sesuai dengan kebutuhan.

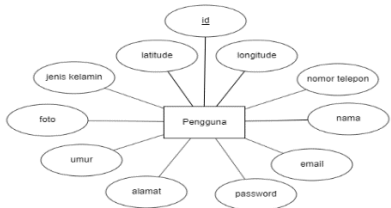
Perancangan dibagi menjadi beberapa tahap, yaitu:

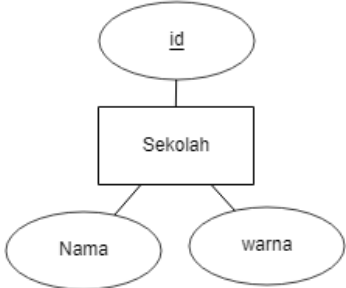
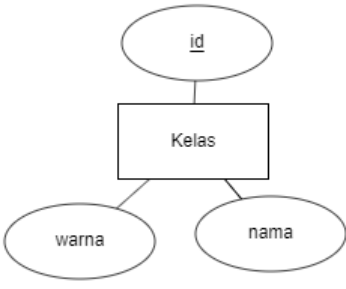

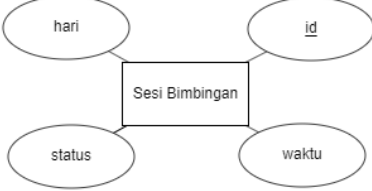
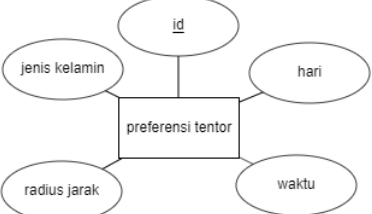
1. Desain Struktur Data

Dalam desain struktur data dilakukan dengan pembuatan *entity relationship diagram* untuk masing-masing kebutuhan. Diagram yang dibuat dibagi menjadi dua jenis, diagram atribut yang bisa dilihat pada Tabel 5, dan diagram relasi pada Tabel 6.

a. Diagram Atribut Entitas

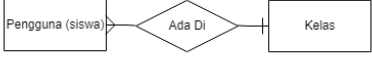


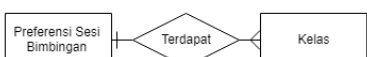


Tabel 5. ERD Atribut Entitas

No	Diagram	Keterangan
1		Merupakan diagram pengguna secara umum, dan pengguna memiliki peran yang berbeda-beda, seperti admin, siswa, dan pengajar.

2	 <pre> graph TD id((id)) --- Sekolah[<u>Sekolah</u>] Sekolah --- Nama((Nama)) Sekolah --- warna((warna)) </pre>	<p>Entitas sekolah memiliki atribut warna dan nama, warna digunakan untuk perbedaan dengan sekolah yang lain.</p>
3	 <pre> graph TD id((id)) --- Kelas[<u>Kelas</u>] Kelas --- warna((warna)) Kelas --- nama((nama)) </pre>	<p>Entitas kelas memiliki atribut yang sama dengan sekolah.</p>
4	 <pre> graph TD id((id)) --- MataPelajaran[<u>Mata pelajaran</u>] MataPelajaran --- warna((warna)) MataPelajaran --- nama((nama)) </pre>	<p>Entitas mata pelajaran juga memiliki atribut dengan fungsi yang sama dengan sekolah dan kelas.</p>
5	 <pre> graph TD id((id)) --- SesiBimbingan[<u>Sesi Bimbingan</u>] SesiBimbingan --- hari((hari)) SesiBimbingan --- status((status)) SesiBimbingan --- waktu((waktu)) </pre>	<p>Sesi bimbingan merupakan entitas tiap sesi yang diajukan oleh siswa, dan memiliki atribut status yang menggambarkan status bimbingan</p>
6	 <pre> graph TD id((id)) --- preferensiTentor[<u>preferensi tentor</u>] preferensiTentor --- jenisKelamin((jenis kelamin)) preferensiTentor --- hari((hari)) preferensiTentor --- radiusJarak((radius jarak)) preferensiTentor --- waktu((waktu)) </pre>	<p>Untuk entitas preferensi tentor, digunakan untuk menyimpan preferensi tentor saat menerima bimbingan, sehingga tentor tidak akan menerima bimbingan jika tidak sesuai dengan preferensi.</p>

b. Diagram relasi

Tabel 6. ERD Relasi Entitas

No	Diagram	Keterangan
1		Pegguna dengan tipe siswa akan memiliki hubungan dengan entitas kelas dimana dia berada.
2		Pegguna dengan tipe tentor memiliki relasi dengan mata pelajaran yang dikuasai
3		Pegguna dengan tipe tentor memiliki relasi dengan preferensi tentor untuk menyimpan pengaturannya
4		Setiap entitas preferensi tentor (preferensi sesi bimbingan) terdapat pengaturan untuk kelas yang diterima oleh tentor untuk diajar, sehingga tentor SD tidak akan menerima permintaan dari anak SMA
5		Setiap entitas sekolah memiliki entitas kelas yang terikat, dan tiap-tiap kelas hanya bisa berada pada satu jenis tingkat sekolah
6		Setiap entitas sesi terhubung dengan pengguna bertipe siswa dan tentor, dan juga dengan mata pelajaran yang diajarkan pada sesi tersebut.

2. Desain Antarmuka

Desain antarmuka dilakukan dengan membuat *wireframe* untuk halaman-halaman yang akan digunakan pada sistem.

a. Halaman *Login*

Halaman *login* merupakan halaman pertama yang dapat diakses pengguna jika belum melakukan *login*. *Wireframe* untuk halaman *Login* dapat dilihat pada Gambar 4 di bawah ini:

KINDY EDUCA

Email

Password

Ingat Saya

[Lupa Password?](#)

[Daftar](#)

Gambar 5. Halaman *Login*

b. Halaman *Dashboard*

Halaman *dashboard* merupakan halaman utama yang terhubung dengan halaman lainnya. Tampilan halaman *dashboard* dapat dilihat pada Gambar 5 di bawah ini:

Kindy EducaLogout>

○ Nama Profil

Menu

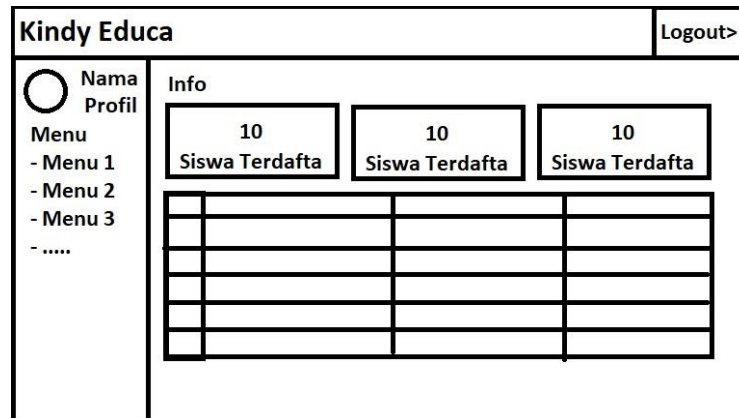
- Menu 1
- Menu 2
- Menu 3
-

Dashboard

Gambar 6. Halaman *Dashboard*

c. Halaman Info

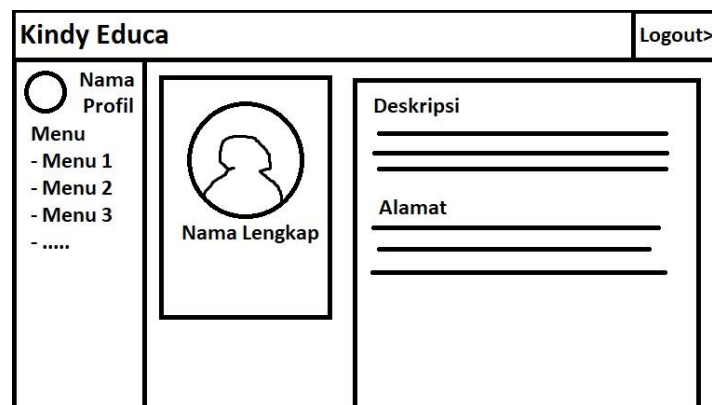
Halaman info dapat diakses dari halaman *dashboard* dan menampilkan informasi sesuai dengan yang diminta oleh pengguna. Susunan halaman info dapat dilihat pada Gambar 6 di bawah ini:



Gambar 7. Halaman Info

d. Halaman Profil

Halaman profil menampilkan profil pengguna. Desain halaman profil dapat dilihat pada Gambar 7:



Gambar 8. Halaman Profil

C. Implementasi

Pada tahap implementasi semua desain yang sudah dirancang disusun dan digabungkan dengan sedemikian rupa sehingga membentuk suatu sistem

yang berfungsi dengan baik. Pada tahap implementasi ini sistem dibuat dengan menggunakan bahasa pemrograman PHP dengan *framework* Laravel.

Pengembangan *web* menggunakan *framework* Laravel umumnya terbagi menjadi tiga tahap antara lain:

1. Pembuatan *Migration* (Basis Data)

Pembuatan migrasi merupakan tahap pertama dalam pengembangan *web* menggunakan Laravel, karena pada tahap ini merupakan bagian dimana basis data untuk semua sistem ditulis. Walaupun sebaiknya migrasi dari basis data sudah siap sejak awal pengembangan, Laravel memberikan solusi jika ada perubahan basis data di tengah proses pengembangan. Salah satu contoh potongan kode dari migrasi terdapat pada Gambar 8 di bawah ini:

```
class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->unsignedInteger('grade_id')->nullable();
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->string('avatar')->nullable();
            $table->decimal('latitude', 10, 8)->nullable();
            $table->decimal('longitude', 11, 8)->nullable();
            $table->unsignedTinyInteger('gender')->nullable();
            $table->unsignedMediumInteger('age')->nullable();
            $table->string('phone')->nullable();
            $table->string('address')->nullable();
            $table->string('description')->nullable();
            $table->unsignedTinyInteger('verified')->default(0);
            $table->rememberToken();
            $table->timestamps();
            $table->foreign('grade_id')->references('id')->on('grades');
        });
    }
}
```

Gambar 9. Migrasi Tabel Pengguna

2. Pembuatan Model

Pembuatan *model* biasanya dilakukan setelah migrasi siap, sehingga ketika nanti *controller* membutuhkan akses ke *database*, *model* dapat menyediakan data yang dibutuhkan langsung tanpa perlu *controller* mengakses *database* sendiri yang keluar dari paradigma MVC. Karena *model* sangat identik dengan basis data Laravel memiliki fungsi yang memudahkan untuk membuat migrasi sekaligus menyiapkan model untuk migrasi yang sama. Salah satu contoh potongan kode dari *model* terdapat pada Gambar 9 di bawah ini:

```
1  <?php
2
3  namespace App;
4
5  use App\Models\Grade;
6  use App\Models\Rate;
7  use App\Models\Session;
8  use App\Models\Subject;
9  use App\Models\TeacherPreference;
10 use Backpack\CRUD\CrudTrait;
11 use Illuminate\Support\Facades\Storage;
12 use Spatie\Permission\Traits\HasRoles;
13 use Illuminate\Notifications\Notifiable;
14 use Illuminate\Foundation\Auth\User as Authenticatable;
15
16 class User extends Authenticatable
17 {
18     use CrudTrait;
19     use HasRoles;
20     use Notifiable;
21 }
```

Gambar 10. Potongan Kode Model Pengguna

3. Pembuatan Controller

Controller merupakan jembatan yang menghubungkan antara *model* dan *view*. Pada *controller* informasi yang dimasukkan pengguna, diteruskan ke *model* yang kemudian memberikan balasan yang diteruskan oleh *controller* kepada *view* agar bisa dikonsumsi pengguna. Salah satu contoh potongan kode dari migrasi terdapat pada gambar 10 di bawah ini:

```

1 |k?php
2
3 namespace App\Http\Controllers\Auth;
4
5 use Backpack\Base\app\Http\Controllers\Controller;
6 use Illuminate\Foundation\Auth\AuthenticatesUsers;
7 use Illuminate\Http\Request;
8
9 class LoginController extends Controller
10 {
11     protected $data = []; // the information we send t
12
13     /*
14     |-----
15     | Login Controller
16     |-----
17     |

```

Gambar 11. Potongan Kode *Login Controller*

4. Pembuatan *View*

View merupakan bagian dari sistem yang berhubungan langsung dengan pengguna, ketika migrasi, model, dan *controller* ditulis menggunakan PHP, *view* ditulis menggunakan *syntax* HTML. Salah satu contoh potongan kode dari *view* terdapat pada gambar 8 di bawah ini:

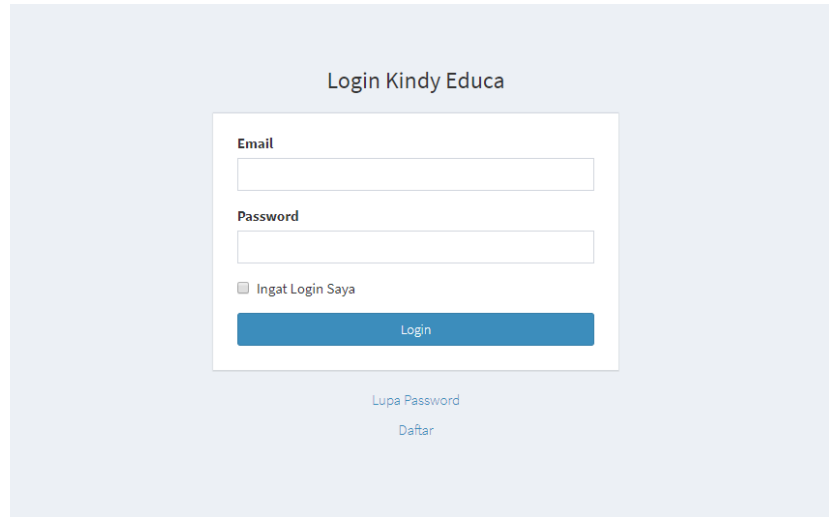
```

1 @extends('backpack::layout_guest')
2
3 @section('content')
4 <div class="row m-t-40">
5 <div class="col-md-4 col-md-offset-4">
6 <h3 class="text-center m-b-20">{{ trans('backpack::base.login') }} Kindy Educa</h3>
7 <div class="box">
8 <div class="box-body">
9 <form class="col-md-12 p-t-10" role="form" method="POST" action="{{ route('b
10 {!! csrf_field() !!}
11
12 <div class="form-group{{ $errors->has($username) ? ' has-error' : '' }}"
13 <label class="control-label">{{ config('backpack.base.authentication
14
15 <div>
16 <input type="text" class="form-control" name="{{ $username }}" v
17
18 @if ($errors->has($username))
19 <span class="help-block">
20 <strong>{{ $errors->first($username) }}</strong>
21 </span>
22 @endif
23 </div>
24 </div>
25
26 <div class="form-group{{ $errors->has('password') ? ' has-error' : '' }}"
27 <label class="control-label">{{ trans('backpack::base.password') }}<
28
29 <div>
30 <input type="password" class="form-control" name="password">
31
32 @if ($errors->has('password'))
33 <span class="help-block">

```

Gambar 12. Kode *Login View*

Setelah selesai ditulis dan dirender di *web browser*, *view* akan tertampil seperti pada Gambar 12.



Gambar 13. *Login View* pada *Browser*

D. Pengujian

Pengujian dilakukan menggunakan instrumen yang merujuk pada standar ISO 25010 yang terdiri dari *Functional Suitability*, *Performance Efficiency*, *Usability*, *Reliability*, *Security*, *Maintainability*, *Portability*. Untuk pengujian *Functional Suitability* dan *Usability* pengambilan data dilakukan dengan kuisioner dan untuk aspek yang lain menggunakan *software*.

1. Pengujian *Functional Suitability*

Pengujian kesesuaian fungsi dilakukan dengan menggunakan ceklis pada lembar *test case* pada tiga orang ahli dalam bidang teknologi informasi dan web. Di bawah ini adalah Tabel 7 yang berisi daftar penguji:

Tabel 7. Daftar Penguji

No	Nama	Pekerjaan	Instansi
----	------	-----------	----------

1	Gathot Fajar	Programmer	Creacle
2	Fauzi Nur Shofa	Programmer	
3	Eko Nugraha	Web Developer	

Hasil pengujian aspek fungsional untuk sistem informasi bimbingan

belajar adalah seperti yang tercantum pada Tabel 8 sebagai berikut:

Tabel 8. Hasil Pengujian Aspek *Functional Suitability*

No	Fungsi	Berhasil	Gagal
1	Registrasi Murid	3	0
2	Registrasi Pengajar	3	0
3	Login	3	0
4	Logout	3	0
5	Ubah Profil	3	0
6	Ubah Foto Profil	3	0
7	Ubah Lokasi	3	0
8	Ubah Password	3	0
9	Menampilkan Daftar Mata Pelajaran	3	0
10	Menambah Mata Pelajaran	3	0
11	Mengubah Mata Pelajaran	2	1
12	Menampilkan Daftar Siswa Terdaftar	3	0
13	Menampilkan Jumlah Siswa Terdaftar	3	0
14	Menampilkan Daftar Pengajar Terdaftar	3	0
15	Menampilkan Jumlah Pengajar Terdaftar	3	0
16	Menampilkan Detail Profil Pengguna	3	0
17	Melakukan Verifikasi Pengguna Terdaftar	3	0
18	Menampilkan Daftar Sesi Bimbingan	3	0
19	Menampilkan Jumlah Sesi Bimbingan	3	0
20	Menampilkan Detail Sesi Bimbingan	3	0
21	Memberikan Persetujuan Pada Sesi Bimbingan	3	0
22	Mencari Pengajar Yang Sesuai	3	0
23	Mengajukan Permintaan Sesi Bimbingan Ke Pengajar	3	0
24	Melihat Detail Sesi Bimbingan Dari Sisi Siswa	3	0
25	Melihat Daftar Sesi Bimbingan	3	0
26	Menampilkan Detail Informasi Pengajar Setelah Permintaan Bimbingan Diterima	3	0
27	Melihat Daftar Sesi Bimbingan	3	0

No	Fungsi	Berhasil	Gagal
28	Melihat Detail Sesi Bimbingan Dari Sisi Pengajar	3	0
29	Mengubah Status Permintaan Sesi Bimbingan	3	0
30	Mengubah Preferensi Sesi Bimbingan	3	0
31	Menampilkan Detail Informasi Siswa Setelah Menerima Permintaan Bimbingan	3	0
	Total	92	1

Berdasarkan hasil pengujian di atas, dapat diketahui nilai aspek *functional*

suitability dengan rumus sebagai berikut:

$$X = \frac{A}{B} = \frac{92}{93} = 0,99$$

2. Pengujian Performance Efficiency

Performance efficiency diukur menggunakan *website* GTMetix dengan pengaturan *test-server* berada di Australia dan browser yang digunakan adalah Google Chrome *Desktop* dengan versi 75. Hasil dari pengujian performa terdapat pada Tabel 9 di bawah ini:

Tabel 9. Efisiensi Performa dari GTMetrix

No	Nama Halaman	Page Speed		YSlow	
		Grade	Skor	Grade	Skor
1	Daftar Siswa	B	89	D	68
2	Daftar Mapel	B	89	D	68
3	Daftar Pengajar	B	89	D	68
4	Daftar Sesi	B	89	D	68
5	Detil Sesi	B	82	D	65
6	Detil Pengguna	B	83	D	64
7	Profil	A	90	D	68
8	Login	B	89	C	70
9	Register	A	92	C	71
10	Daftar Sesi	B	87	D	68
11	Detail Sesi	B	80	D	65
12	Tambah Sesi	B	89	D	67
13	Daftar Sesi	B	81	D	64
14	Detail Sesi	C	74	D	65
15	Pengaturan Sesi	B	81	D	68

	Rata-rata	B	85,6	D	67,1
--	-----------	---	------	---	------

3. Pengujian *Usability*

Pengujian *usability* sistem dilakukan dengan menggunakan kuisiomer yang berisi pernyataan yang terdapat pada SUS atau *System Usability Scale*. Kuisiomer diberikan kepada responden yang berjumlah 21 orang yang terdiri dari murid, tentor, dan admin. Hasil dari kuisiomer tersebut tercantum pada Tabel 10 di bawah ini:

Tabel 10. Hasil Kuisiomer *Usability*

No.	Total	Konversi
1	47	26
2	47	58
3	51	30
4	50	55
5	55	34
6	50	55
7	51	30
8	45	60
9	50	29
10	58	47
Total Konversi		424

Nilai total dari masing-masing butir soal dijumlah, dan dikonversi sesuai dengan ketentuan perhitungan SUS. Kemudian dilakukan perhitungan di bawah ini:

$$X = \frac{T}{n} * 2.5 = \frac{424}{21} * 2.5 = 50.47$$

Keterangan:

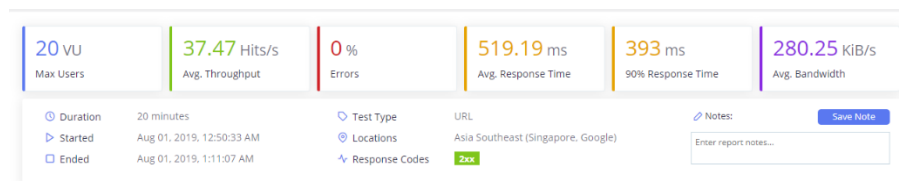
X= Skor akhir SUS

T = Total nilai yang sudah dikonversi

n = Jumlah responden

4. Pengujian *Reliability*

Pengujian reliabilitas sistem dilakukan dengan *website* BlazeMeter yang mengukur jumlah *hit*, dan *error* yang terjadi pada pengujian waktu tertentu. Gambar di bawah ini merupakan hasil pengujiannya, Gambar 14 menampilkan hasil pengujian, dan Gambar 15 menampilkan jumlah *hits* setelah diuji..



Gambar 14. Hasil Pengujian Menggunakan BlazeMeter

Label	# Samples	Average	90% Line	99% Line	Error %	Hits/s
ALL	44754	523.06 ms	393 ms	10559 ms	0%	37.33
Kindy Educa Login	44754	523.06 ms	393 ms	10559 ms	0%	37.33

Gambar 15. Detil Jumlah *Hit* Setelah Diuji

Dari hasil pengujian di atas dapat dihitung nilai MTBF sebagai berikut

$$MTPBF = \frac{T}{F} = \frac{20}{0} = 0 \text{ Failure/Minute}$$

5. Pengujian *Security*

Pengujian aspek keamanan dilakukan menggunakan *website* Acunetix yang menyediakan layanan gratis untuk memindai keamanan sebuah sistem. Hasil pemindaian sistem informasi Kindy Educa tercantum pada Gambar 15 di bawah ini:

Scan of kindyeduca.fikry13.com

Scan details

Scan information	
Start time	29/07/2019, 14:35:19
Start url	kindyeduca.fikry13.com
Host	kindyeduca.fikry13.com
Scan time	19 minutes, 23 seconds
Profile	Network Scan

Threat level

Acunetix Threat Level 0

No vulnerabilities have been discovered by the scanner.

Alerts distribution

Total alerts found	11
High	0
Medium	0
Low	0
Informational	11

Gambar 16. Hasil *Scan* dengan Acunetix Web

6. Pengujian Maintainability

Pengujian aspek *maintainability* dilakukan dengan mengukur MI atau *Maintainability Index*. Pengukuran dilakukan menggunakan PHPMetric yang memindai keseluruhan kode sumber dan menampilkan hasil pemindaian dengan menyertakan nilai MI. Hasil pemindaian kode sumber sistem informasi tercantum pada Gambar 16 di bawah ini:

Name	loc	lloc	CommW	Length	Volume	Vocabulary	Eff.	MI	MlwC
.	17986	4130	10.47	24968	458.41	21.96	8468.24	91.11	80.64

Gambar 17. Hasil Pemindaian Kode Sumber dengan PHPMetric

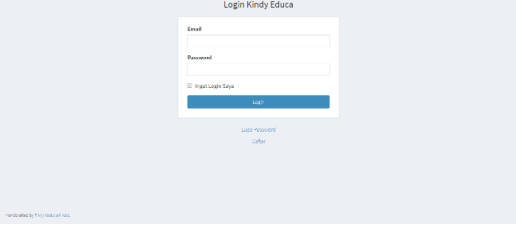
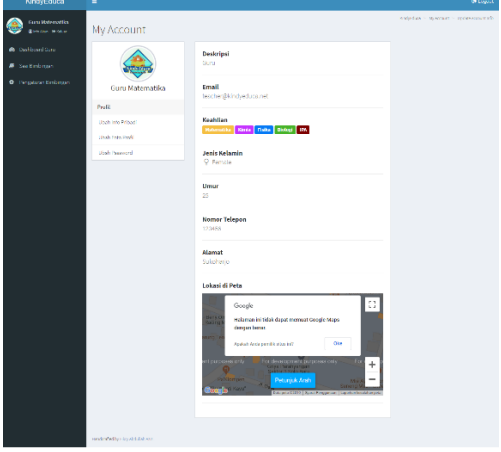
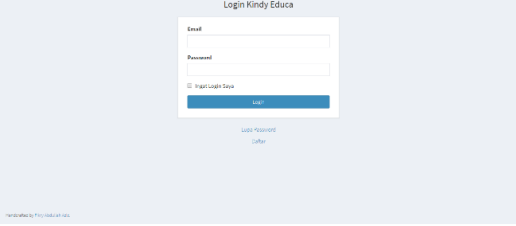
Nilai MI terdapat pada kolom dengan *header* MI.

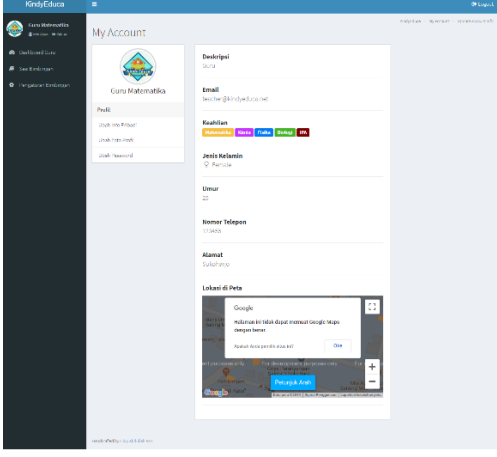
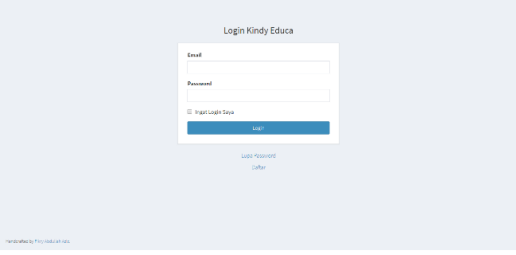
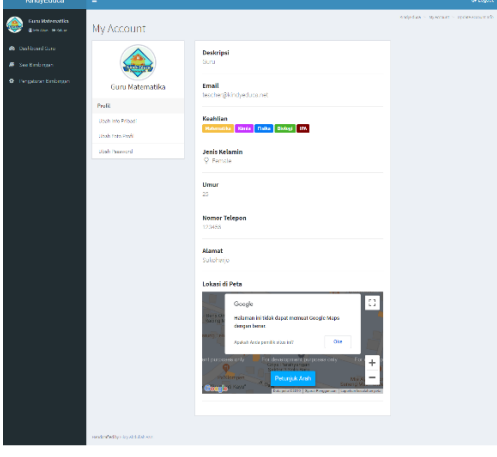
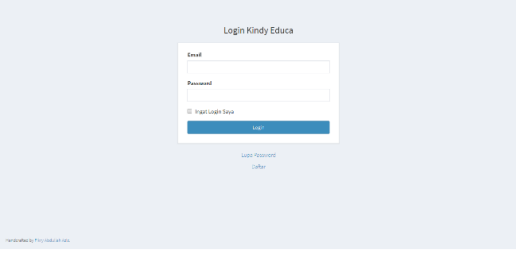
7. Pengujian Portability

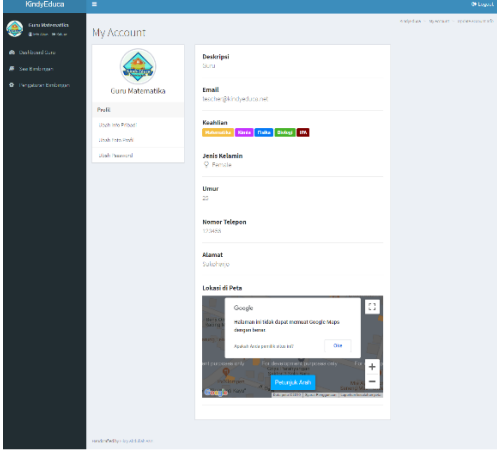
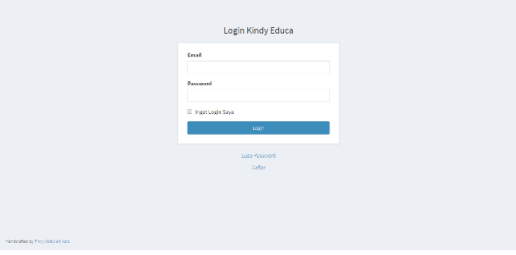
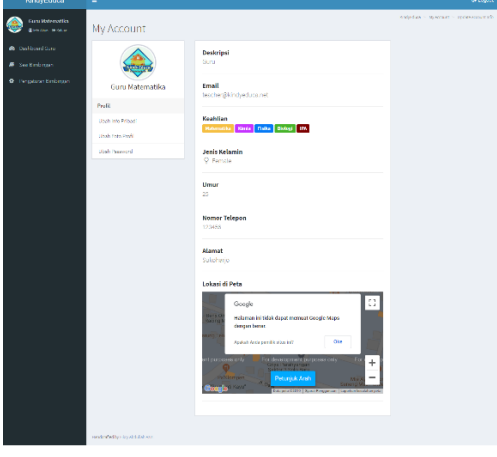
Pengujian portabilitas dilakukan dengan menjalankan sistem pada berbagai jenis *browser* dan *device* yang berbeda, *browser* yang menjadi target adalah Mozilla Firefox, Google Chrome, Microsoft Edge, Safari, dan Opera, sedangkan untuk *device* yang digunakan untuk menguji adalah *device* dengan

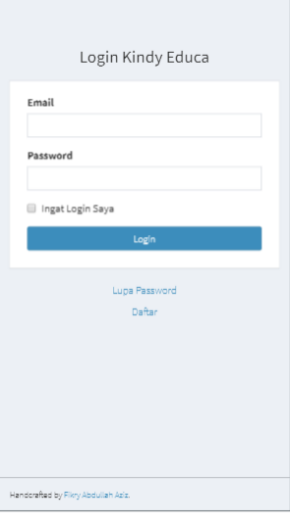
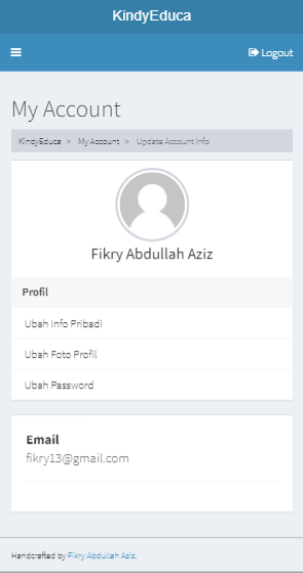
sistem operasi android dan iOS. Hasil pengujian dituliskan pada Tabel 11 di bawah ini:

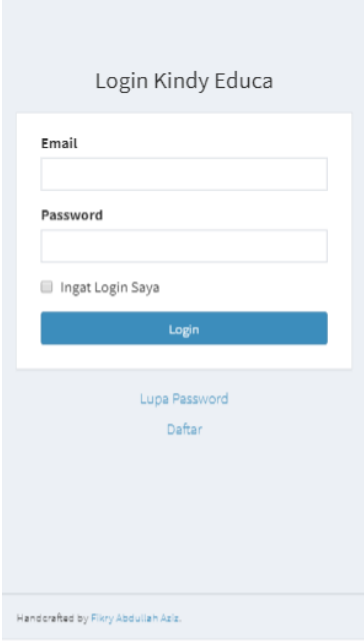
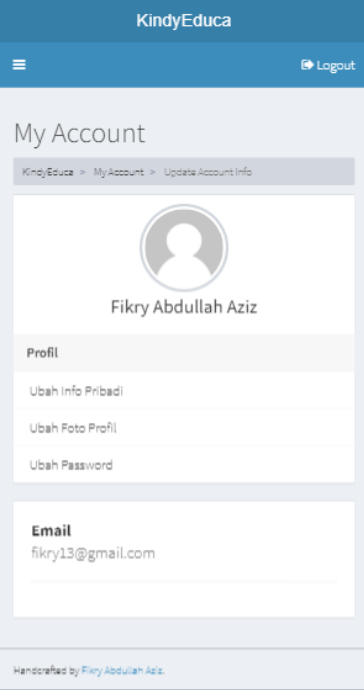
Tabel 11. Pengujian *Portability*

No	Target	Tampilan <i>Login</i> dan Profil	Hasil
1	Mozilla Firefox		Berjalan dengan Baik
			Berjalan dengan Baik
2	Google Chrome		Berjalan dengan Baik

No	Target	Tampilan <i>Login</i> dan Profil	Hasil
			Berjalan dengan Baik
3	Internet Explorer		Berjalan dengan Baik
			Berjalan dengan Baik
4	Opera		Berjalan dengan Baik

No	Target	Tampilan <i>Login</i> dan Profil	Hasil
			Berjalan dengan Baik
5	Safari		Berjalan dengan Baik
			Berjalan dengan Baik

No	Target	Tampilan <i>Login</i> dan Profil	Hasil
6	Android		Berjalan dengan Baik
			Berjalan dengan Baik

No	Target	Tampilan <i>Login</i> dan Profil	Hasil
7	iOS		Berjalan dengan Baik
			Berjalan dengan Baik

E. Pemeliharaan

Tahap pemeliharaan merupakan tahap yang dilakukan setelah implementasi. Pada tahap ini dilakukan beberapa perbaikan pada sistem yang sebelumnya terdapat kegagalan saat diuji oleh ahli.

F. Pembahasan

Pada tahap pembahasan hasil penelitian dikemukakan. Penelitian ini melewati beberapa tahap, tahap analisis, tahap desain, tahap implementasi, pengujian, dan perawatan.

Pada tahap analisis dilakukan analisa kebutuhan pengguna dan juga analisa hal-hal yang dibutuhkan untuk mengembangkan sistem. Saat melakukan observasi dan wawancara untuk kebutuhan sistem, peneliti tidak mewawancarai pihak siswa dan tutor, sehingga mengakibatkan adanya fitur yang belum terimplementasi pada sistem, dan juga alur penggunaan sistem yang rumit bagi siswa.

Pada tahap desain, hal-hal yang didapat pada tahap analisa diterjemahkan dalam bentuk desain yang berupa *ERD diagram*, dan *wireframe*. Tahap implementasi dilakukan dengan menggunakan *framework* Laravel sebagai basis sistem. Setelah implementasi, sistem diujikan berdasarkan standar ISO 25010 yang memiliki beberapa kriteria antara lain *functional suitability*, *performance efficiency*, *usability*, *reliability*, *security*, *maintainability*, dan *portability*. Hasil pengujian masing-masing kriteria adalah sebagai berikut:

1. Hasil Pengujian *Functional Suitability*

Dalam pengujian aspek *functional suitability* jika nilai yang didapat mendekati 1 maka sistem memiliki fungsi yang baik. Setelah di uji sistem yang dikembangkan memiliki nilai 0,99 dari 1 untuk nilai fungsionalitasnya yang berarti sangat baik.

2. Hasil Pengujian *Performance Efficiency*

Untuk aspek efisiensi performa, sistem mendapatkan nilai 89 dengan *grade* B untuk parameter PageSpeed dan 67,1 dengan *grade* D untuk parameter YSlow. Nilai yang didapat lalu dibandingkan dengan rata-rata yang ada di *webstie* GTMetrix sebagai pembanding, nilai rata-rata GTMetrix untuk kategori PageSpeed adalah 70 dengan *grade* C dan kategori YSlow rata-ratanya adalah 70 dengan *grade* C. Nilai PageSpeed sistem yang dikembangkan berada di atas rata-rata GTMetrix sedangkan YSlow berada di bawah rata-rata, sehingga *grade* yang didapatkan adalah C, atau cukup.

3. Hasil Pengujian *Usability*

Pada pengujian aspek *usability* nilai yang didapatkan diukur pada skala *usability* yang dikemukakan oleh Bangor, et. al. pada tahun 2009. Nilai yang didapat dari hasil pengujian *usability* adalah 50,4 yang jika mengacu pada skala Bangor maka masuk kategori cukup.

4. Hasil Pengujian *Reliability*

Untuk kriteria *reliability* perlu diukur nilai MTBF-nya, jika nilai yang didapat kecil, maka tingkat reliabilitasnya tinggi. Setelah diuji sistem mendapatkan

nilai MTBF yang skornya 0, yang berarti sistem sama sekali tidak memiliki *error* selama masa percobaan dan diprediksi tidak akan *error* dalam waktu dekat.

5. Hasil Pengujian *Security*

Pada pengujian keamanan, hasil pengujian diukur pada skala kemanan *Accunetix Web Scanner* yang memiliki empat tingkat dari 0 (nol) sampai 3 (tiga), semakin kecil angka yang didapat maka sistem semakin aman. Setelah diuji tingkat ancaman sistem berada pada *level 0* yang berarti aman.

6. Hasil Pengujian *Maintainability*

Pada pengujian *maintainability* digunakan skala *Maintainability Index* yang dikemukakan oleh Coleman pada tahun 1994 yang jika nilai diatas 85 maka sistem sangat mudah dirawat. Nilai *Maintainability Index* yang didapat dari PHPMetric adalah 91,11 yang nilainya lebih besar dari pada 85 yang berarti sistem ini mudah untuk dirawat.

7. Hasil Pengujian *Portability*

Pada pengujian *portability* untuk mengetahui sistem memenuhi standar adalah jika sistem dapat berjalan dengan lancar pada seluruh target uji. Setelah diuji sistem dapat berjalan tanpa gangguan pada seluruh target *browser* dan *device*.