#### **BAB IV**

#### RANCANGAN SISTEM DAN PROGRAM USULAN

#### 4.1. Analisa Kebutuhan Software

Analisa kebutuhan *software* menjelaskan tahapan perancangan sistem yang akan dibuat untuk mempersiapkan proses implementasi sistem dan untuk menggambarkan secara jelas proses-proses yang diinginkan didalam membangun perangkat lunak.

#### 4.1.1 Tahapan Analisis

Diajaryuk.com merupakan website e-learning bahasa inggris yang penulis rancang untuk murid MTs Al-Barokah dengan maksud untuk memberi pelajaran tambahan dalam mata pelajaran bahasa inggris. Berikut ini spesifikasi website yang akan dibangun berdasarkan pemakaiannya;

Admin website:

- A1. Administrasi dapat mengelola data modules.
- A2. Administrasi dapat mengelola data materi.
- A3. Administrasi dapat mengelola data soal.
- A4. Administrasi dapat mengelola data quiz.
- A5. Administrasi dapat mengelola data try out.
- A6. Administrasi dapat login.
- A7. Administrasi dapat akses menu registrasi admin.

User website:

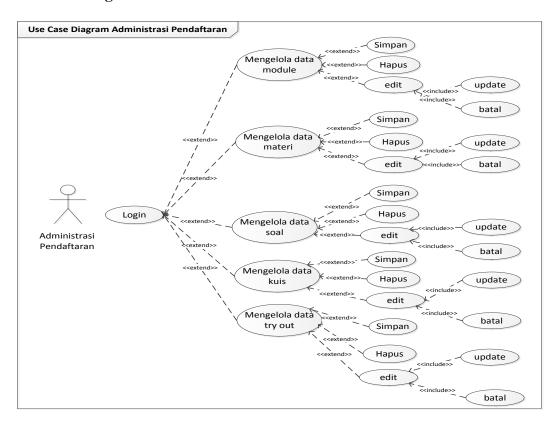
- B1. *User* dapat login sebelum akses materi di *website*.
- B2. *User* dapat akses data materi.

- B3. *User* dapat akses menu try out.
- B4. User dapat akses menu quiz.
- B5. User dapat akses menu registrasi user.

### 4.1.2 Use Case Diagram

Use Case menggambarkan sebuah interaksi antara satu atau lebih aktor dengan sistem infomasi yang akan dibuat. Use Case digunakan untuk mengetahui fungsi apa saja yang ada dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Berikut Use Case sistem usulan yang akan dibuat:

#### 1. Use Case Diagram Administrasi Website



Gambar IV.1. Use Case Diagram Administrasi Website

Tabel IV.1.
Deskripsi *Use Case* Diagram Pengelolaan Data Module

Use Case Name	Pengelolaan Data Module
Requirements	A1
Goal	Administrasi dapat mengelola data module, baik menambah, mengubah, menghapus, serta melihat list module.
Pre-conditions	Administrasi telah login sebagai admin dan mengakses menu data module.
Post-conditions	Data module berhasil tersimpan, terupdate, dan terhapus, dan melihat semua list module.
Failed end condition	Gagal menyimpan, gagal update, gagal hapus.
Primary Actor	Administrasi Website
Main Flow / Basic Path	<ol> <li>Administrasi mengakses data module.</li> <li>Administrasi menginput data module.</li> <li>Administrasi memilih tombol save.</li> </ol>
Alternate flow / Invariant 1	<ol> <li>Administrasi memilih tombol edit.</li> <li>Administrasi mengubah data module.</li> <li>Administrasi memilih tombol save.</li> </ol>
Invariant 2	<ol> <li>Administrasi memilih tombol         "Delete"</li> <li>System menghapus data module.</li> </ol>

Deskripsi Use Case Diagram Pengelolaan Data Materi:

Tabel IV.2. Deskripsi *Use Case* Diagram Pengelolaan Data Materi

Use Case Name	Pengelolaan Data Materi
Requirements	A2

Goal	Administrasi dapat mengelola data
	materi, baik menambah, mengubah,
	menghapus, serta melihat list materi.
Pre-conditions	Administrasi telah login sebagai
	admin dan mengakses menu data
	module.
Post-conditions	Data Materi berhasil tersimpan,
	terupdate, terhapus, dan melihat list
	materi.
Failed end condition	Gagal menyimpan, gagal update, gagal
	hapus.
Primary Actor	Administrasi Website
Main Flow / Basic Path	Administrasi mengakses data
	materi.
	2. Administrasi memilih tombol
	"create matery"
	3. Administrasi menginput data
	materi.
	4. Administrasi memilih tombol save.
Alternate flow / Invariant 1	1. Administrasi memilih tombol edit.
	2. Administrasi mengubah data
	materi.
	3. Administrasi memilih tombol save.
Invariant 2	Administrasi memilih tombol
	"Delete"
	2. <i>System</i> menghapus data materi.

Deskripsi *Use Case* Diagram Mengelola Data Soal:

Tabel IV.3. Deskripsi *Use Case* Diagram Mengelola Data Soal

Use Case Name	Mengelola Data Soal
Requirements	A3
Goal	Untuk mengelola data Soal
Pre-conditions	Administrasi website login terlebih dahulu

Post-conditions	Data soal dapat ditambahkan,
	disimpan, dan di <i>update</i>
Failed end condition	Gagal menyimpan, gagal <i>update</i> , gagal hapus
Primary Actor	Administrasi Website
Main Flow / Basic Path	<ol> <li>Administrasi mengakses data soal.</li> <li>Administrasi memilih tombol         <i>"create questions"</i></li> <li>Administrasi menginput data soal.</li> <li>Administrasi memilih tombol save.</li> </ol>
Alternate flow / Invariant 1	<ol> <li>Administrasi memilih tombol edit.</li> <li>Administrasi mengubah data soal.</li> <li>Administrasi memilih tombol save.</li> </ol>
Invariant 2	<ol> <li>Administrasi memilih tombol         "Delete"</li> <li>System menghapus data soal.</li> </ol>

Deskripsi *Use Case* Diagram Mengelola Data Kuis:

Tabel IV.4. Deskripsi *Use Case* Diagram Mengelola Data Kuis

Use Case Name	Mengelola Data Kuis
Requirements	A4
Goal	Untuk Mengelola data kuis
Pre-conditions	Administrasi website login terlebih dahulu
Post-conditions	Data kuis dapat ditambahkan, disimpan, dan di <i>update</i>
Failed end condition	Gagal menyimpan, gagal <i>update</i> , gagal hapus
Primary Actor	Administrasi Website
Main Flow / Basic Path	<ol> <li>Administrasi mengakses data kuis.</li> <li>Administrasi memilih tombol         <i>"create quiz"</i></li> <li>Administrasi menginput data kuis.</li> </ol>

	4. Administrasi memilih tombol save.
Alternate flow / Invariant 1	1. Administrasi memilih tombol edit.
	2. Administrasi mengubah data kuis.
	3. Administrasi memilih tombol save.
Invariant 2	1. Administrasi memilih tombol
	"Delete"
	2. System menghapus data kuis.

Deskripsi Use Case Diagram Mencetak Laporan Data Try Out:

Tabel IV.5.
Deskripsi *Use Case* Diagram Mencetak Laporan Data *Try Out* 

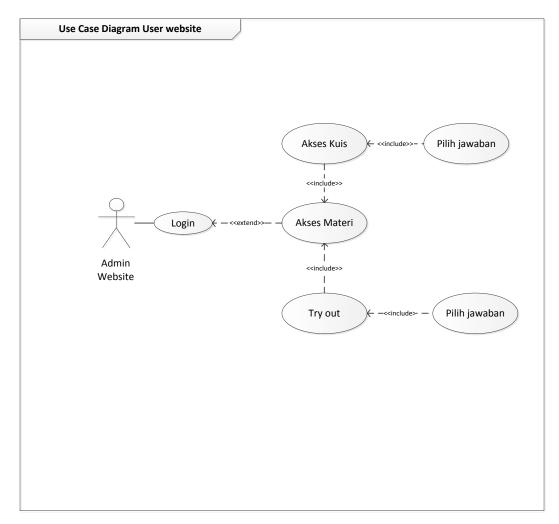
Use Case Name	Mengelola Data Try out
Requirements	A5
Goal	Untuk Mengelola data Try Out
Pre-conditions	Administrasi website login terlebih dahulu
Post-conditions	Data <i>try out</i> dapat ditambahkan, disimpan, dan di <i>update</i>
Failed end condition	Gagal menyimpan, gagal <i>update</i> , gagal hapus
Primary Actor	Administrasi Website
Main Flow / Basic Path	<ol> <li>Administrasi mengakses data kuis.</li> <li>Administrasi memilih tombol         "create try outs"</li> <li>Administrasi menginput data try         out.</li> <li>Administrasi memilih tombol save.</li> </ol>
Alternate flow / Invariant 1	<ol> <li>Administrasi memilih tombol edit.</li> <li>Administrasi mengubah data try out.</li> <li>Administrasi memilih tombol save.</li> </ol>
Invariant 2	<ol> <li>Administrasi memilih tombol         "Delete"</li> <li>System menghapus data try out.</li> </ol>

# Deskripsi *Use Case* Diagram *Login*:

# Tabel IV.6. Deskripsi *Use Case* Diagram *Login*

Use Case Name	Login
Requirements	A6
Goal	Untuk verivikasi pengguna
Pre-conditions	-
Post-conditions	Pengguna terverifikasi
Failed end condition	Gagal memverifikasi
Primary Actor	Administrasi & User
Main Flow / Basic Path	<ol> <li>Pengguna input <i>email</i> dan password</li> <li>Pengguna memilih tombol submit</li> <li>Pilih tombol submit</li> </ol>

# 2. Use Case Diagram User



Gambar IV.2. Use Case Diagram User

Deskripsi Use Case Diagram Login User:

Tabel IV.7.
Deskripsi *Use Case* Diagram Login *User* 

Use Case Name	Login
Requirements	B1
Goal	<i>User</i> terverifikasi
Pre-conditions	-
Post-conditions	User bisa masuk sistem dan terverifikasi

Failed end condition	User gagal terverifikasi
Primary Actor	Pengguna website
Main Flow / Basic Path	Pengguna isi data email &  paggunad  maggunad  magg
	password 2. Penggguna memilih tombol <i>submit</i>
	2. I chiggguna memilin tomoon suomu

Deskripsi *Use Case* Diagram Akses Data Materi:

Tabel IV.8. Deskripsi *Use Case* Diagram Akses Data Materi

Use Case Name	Akses Data Materi
Requirements	B2
Goal	<i>User</i> bisa mengakses data materi
Pre-conditions	User login terlebih dahulu
Post-conditions	Data materi bisa diakses
Failed end condition	Data materi gagal diakses
Primary Actor	User
Main Flow / Basic Path	1. User login
	2. <i>User</i> memilih menu module
	3. <i>User</i> memilih menu materi
	4. System menampilkan menu materi

Deskripsi Use Case Diagram Mengelola Akses Menu Try out:

Tabel IV.9. Deskripsi *Use Case* Diagram Akses Menu Try out

Use Case Name	Akses Menu Try out
Requirements	B3
Goal	User akses data try out
Pre-conditions	User login terlebih dahulu
Post-conditions	User bisa akses data try out

Failed end condition	User gagal akses data try out		
Primary Actor	User		
Main Flow / Basic Path	1. <i>User</i> memilih data materi		
	2. <i>User</i> memilih tombol "next"		
	3. <i>System</i> menampilkan menu try out		

Deskripsi Use Case Diagram Mengelola Akses Menu Kuis:

Tabel IV.10. Deskripsi *Use Case* Diagram Mengelola Akses Menu Kuis

Use Case Name	Akses Menu Kuis
Requirements	B4
Goal	User akses data kuis
Pre-conditions	User login dan menyelesaikan materi
	dan try out
Post-conditions	User mengakses data kuis
Failed end condition	User gagal mengakses data kuis
Primary Actor	User
Main Flow / Basic Path	1. User menyimak materi
	2. User mengerjakan try out
	3. <i>User</i> mengakses data kuis
	4. <i>System</i> menampilkan data kuis

Deskripsi Use Case Diagram Akses Menu Registrasi User:

Tabel IV.11.

Deskripsi *Use Case* Diagram Akses Menu Registrasi *User* 

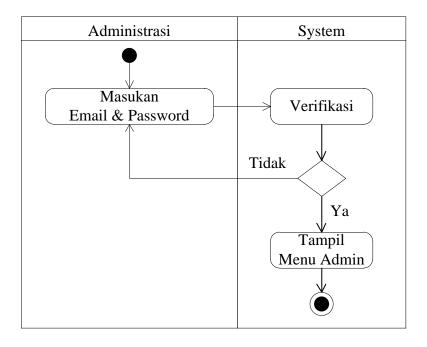
Use Case Name	Akses Menu Registrasi User
Requirements	B5
Goal	User akses menu registrasi User
Pre-conditions	-
Post-conditions	User bisa membuat data User baru

Failed end condition	User gagal menyimpan data user baru
Primary Actor	User / Pengguna website
Main Flow / Basic Path	<ol> <li>User memilih menu registrastion</li> <li>User mengisi data user</li> <li>User memilih tombol submit</li> <li>System menyimpan data user baru</li> </ol>

## 3. Activity Diagram

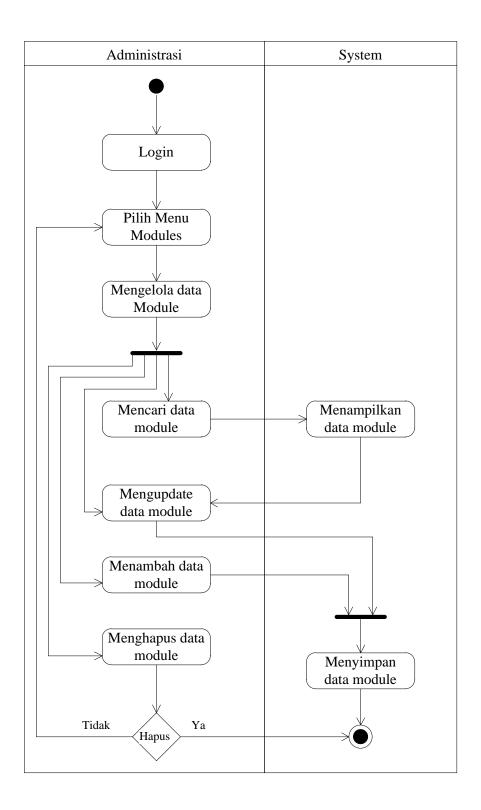
Diagram aktivitas atau activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak.

### A. Activity Diagram Login Administrasi



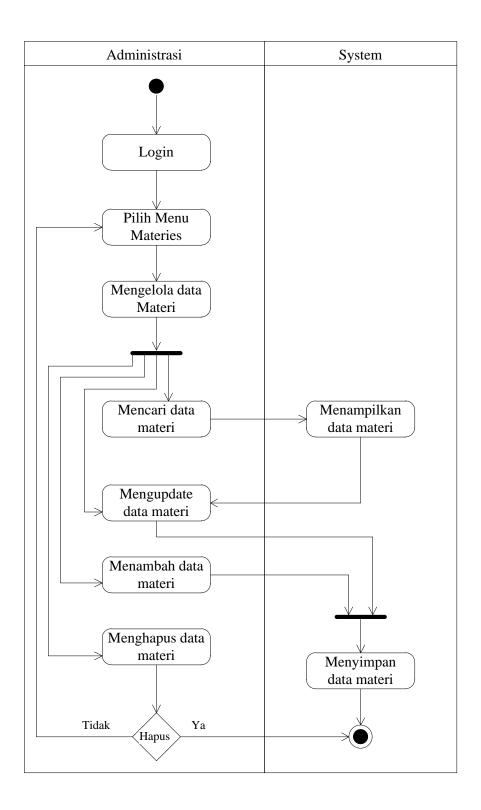
Gambar IV.3. Activity Diagram Login Administrasi

## B. Activity Diagram Mengelola Data Module



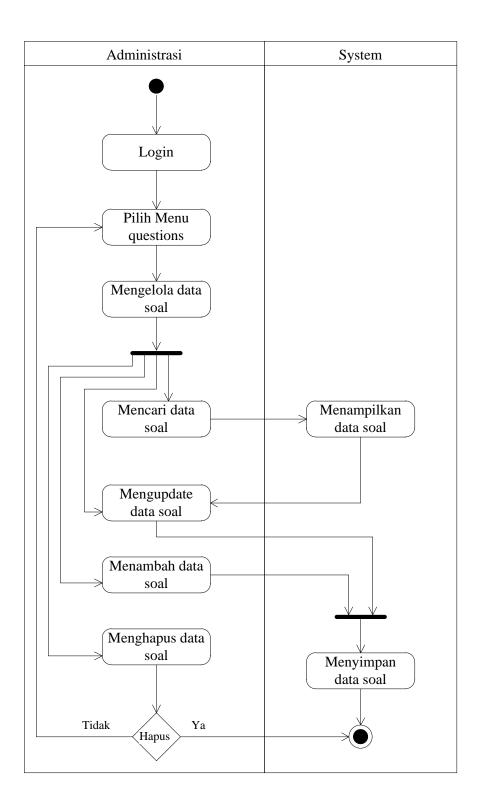
Gambar IV.4. Activity Diagram Mengelola Data Module

## C. Activity Diagram Mengelola Data Materi



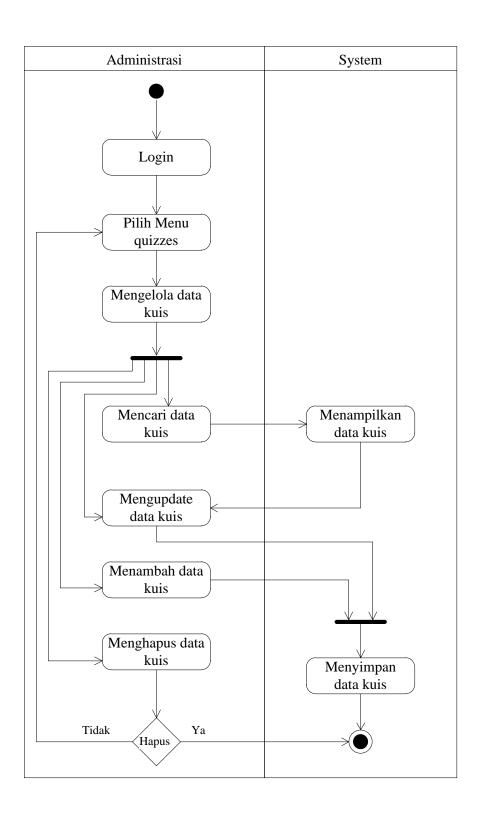
Gambar IV.5. Activity Diagram Mengelola Data Materi

## D. Activity Diagram Mengelola Data Soal



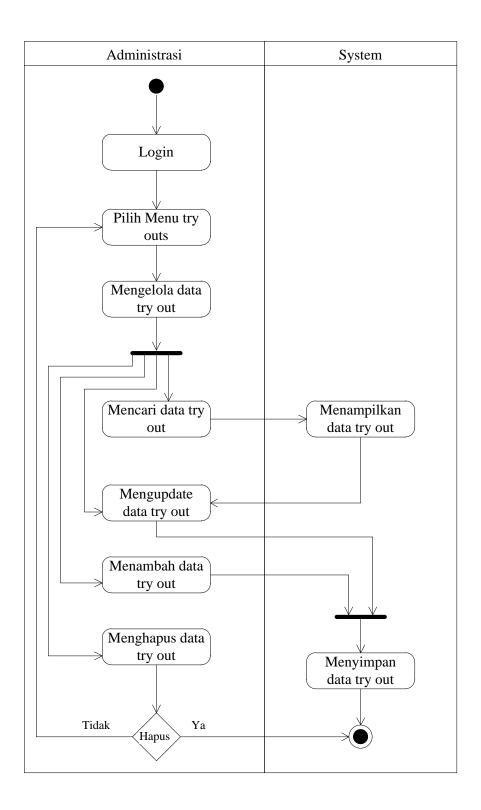
Gambar IV.6. Activity Diagram Mengelola Data Soal

## E. Activity Diagram Mengelola Kuis



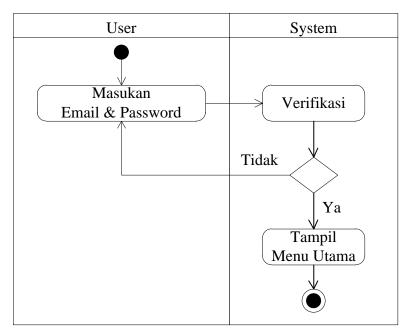
Gambar IV.7. Activity Diagram Mengelola Kuis

## F. Activity Diagram Mengelola Try out



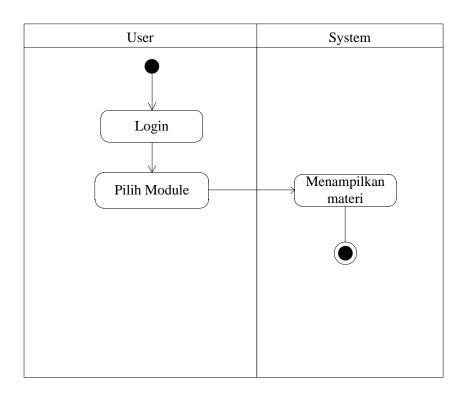
Gambar IV.8. Activity Diagram Mengelola Try out

## G. Activity Diagram login User

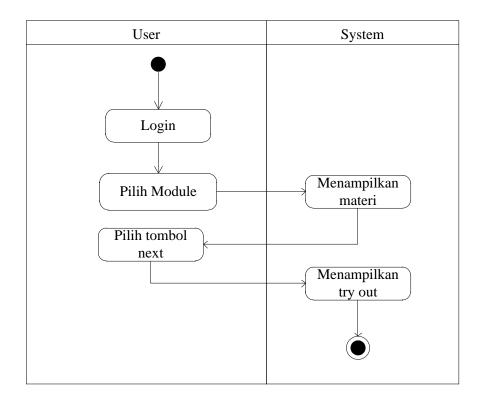


Gambar IV.9. Activity Diagram Login User

# H. Activity Diagram Akses Data Materi

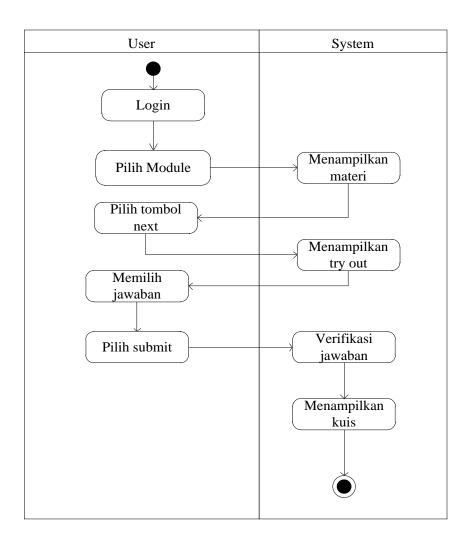


Gambar IV.10. Activity Diagram Akses Data Materi I. Activity Diagram Akses Menu Try Out



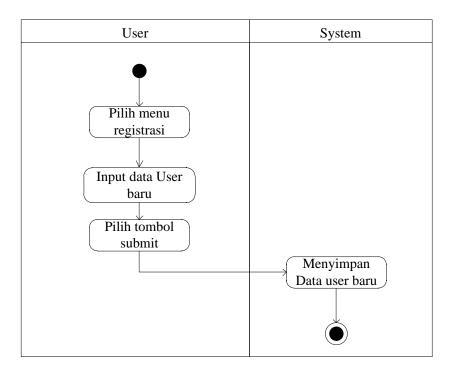
Gambar IV.11. Activity Diagram Akses Menu Try Out

# J. Activity Diagram Akses Menu kuis



Gambar IV.12. Activity Diagram Akses Menu Kuis

# K. Activity Diagram Akses Registrasi User



Gambar IV.13. Activity Diagram Registrasi User

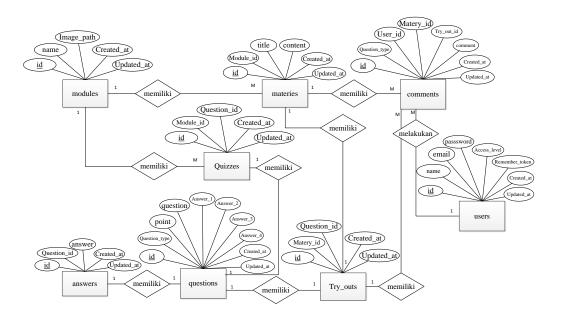
#### 4.2. Desain

Pada tahapan ini, penulis menterjemahkan kebutuhan-kebutuhan *software* yang sudah dianalisa pada tahapan sebelumnya ke sebuah perancangan perangkat lunak.

#### **4.2.1. Database**

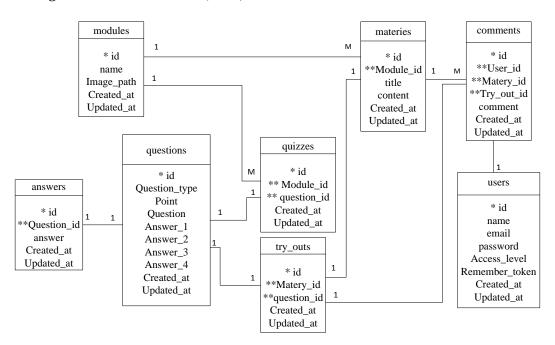
Menggambarkan hubungan antar table yang dibuat beserta relasi antar tabel. Dalam penggambarannya disini menggunakan Entity Relationship Diagram untuk pembuatan *website* sebagai berikut:

### 1. Entity Relationship Diagram (ERD)



Gambar IV.14. Entity Relationship Diagram (ERD)

#### 2. Logical Record Structure (LRS)



Gambar IV.15. Logical Record Structure (LRS)

### 3. Spesifikasi File

### A. Spesifikasi File Tabel Answers

Nama Database : elearning

Nama File : answers

Akronim : answers

Tipe File : File Master

Akses File : Random

Panjang Record : 256 Byte

Kunci Field : id

Tabel IV.12. Spesifikasi File Tabel Answers

No	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id	Id	Int	10	Primary Key
2	Id Pertanyaan	Question_id	int	11	Foreign key

3	Jawaban	Answer	varchar	191	
4	Waktu dibuat	Created_at	Timestamp	-	
5	Waktu diperbarui	Updated_at	Timestamp	-	

# **B.** Spesifikasi File Tabel Comments

Nama Database : elearning

Nama File : Comments

Akronim : comments

Tipe File : File Master

Akses File : Random

Panjang Record : 272 Byte

Kunci Field : id

Tabel IV.13. Spesifikasi File Tabel Comments

No	Elemen Data	Nama Field	Туре	Size	Keterangan
1	Id Komentar	Id	Int	10	Primary Key
2	Id user	<i>User</i> _id	Int	11	Foreign key
3	Id materi	Matery_id	Int	11	Foreign key
4	Id try out	Tyr_out_id	Int	11	Foreign key
5	Komentar	Comment	Text	-	
6	Waktu dibuat	Created_at	Timestamp	-	
7	Waktu diubah	Updated_at	Timpestamp	_	

# C. Spesifikasi File Tabel Materies

Nama Database : elearning

Nama File : Materies

Akronim : materies

Tipe File : File Master

Akses File : Random

Panjang Record : 70 Byte

Kunci Field : id

## Tabel IV.14. Spesifikasi File Tabel Materies

No	Elemen Data	Nama Field	Туре	Size	Keterangan
1	Id materi	Id	int	10	Primary Key
2	Id module	Module_id	int	11	Foreign key
3	Judul	Title	varchar	191	
4	konten	Content	Text	ı	
5	Waktu dibuat	Created_at	Timestamp	-	
6	Waktu diubah	Updated_at	Timpestamp	-	

# D. Spesifikasi File Modules

Nama Database : elearning

Nama File : Modules

Akronim : Modules

Tipe File : File Master

Akses File : Random

Panjang Record : 35 Byte

Kunci Field : id

## Tabel IV.15. Spesifikasi File Tabel Modules

No	Elemen Data	Nama Field	Type	Size	Keterangan	

1	Id modules	Id	Int	10	Primary Key
2	Nama module	Name	varchar	191	
3	Image path	Image_path	text	-	
	Waktu dibuat	Created_at	Timestamp	-	
	Waktu diubah	Updated_at	Timpestamp	1	

# E. Spesifikasi File Tabel Questions

Nama Database : elearning

Nama File : Questions

Akronim : questions

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 208 Byte

Kunci Field : id

# Tabel IV.16. Spesifikasi File Tabel *Questions*

No	Elemen Data	Nama Field	Туре	Size	Keterangan
1	Id pertanyaan	Id	Int	12	Primary Key
2	Tipe pertanyaan	Question_type	varchar	191	Foreign Key
3	Point	Point	Int	11	
4	Pertanyaan	Question	Text	1	
5	Jawaban 1	Answer_1	varchar	191	
6	Jabawan 2	Answer_2	varchar	191	
7	Jawaban 3	Answer_3	varchar	191	
8	Jawaban 4	Answer_4	varchar	191	
9	Waktu dibuat	Created_at	Timestamp	-	
10	Waktu diubah	Updated_at	Timpestamp	-	

# F. Spesifikasi File Tabel Quizzes

Nama Database : elearning

Nama File : Quizzes

Akronim : quizzes

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 428 Byte

Kunci Field : id

## Tabel IV.17. Spesifikasi File Tabel Quizzes

No	Elemen Data	Nama Field	Туре	Size	Keterangan
1	Id kuis	Id	Int	19	Primary Key
2	Id module	Module_id	Int	11	Foreign Key
3	Id pertanyaan	Question_id	int	11	Foreign Key
4	Waktu dibuat	Created_at	Timestamp	-	
5	Waktu diubah	Updated_at	Timpestamp	-	

# G. Spesifikasi File Tabel Try outs

Nama Database : elearning

Nama File : Try out

Akronim : try\_out

Tipe File : File Transaksi

Akses File : Random

Panjang Record : 24 Byte

Kunci Field : id

Tabel IV.18. Spesifikasi File Tabel Try out

No	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id try out	Id	Int	10	Primary Key
2	Id matery	Matery_id	Int	11	Foreign Key
3	Id pertanyaan	Question_id	Int	11	Foreign Key
4	Waktu dibuat	Created_at	Timestamp	-	
5	Waktu diubah	Updated_at	Timpestamp	-	

# H. Spesifikasi File Tabel *Users*

Nama Database : elearning

Nama File : *User*s

Akronim : users

Tipe File : File Master

Akses File : Random

Panjang Record : 13 Byte

Kunci Field : id

Tabel IV.19. Spesifikasi File Tabel Detail *Users* 

No	Elemen Data	Nama Field	Type	Size	Keterangan
1	Id user	Id	Int	6	Primary Key
2	Nama	Name	Varchar	191	
3	Email	Email	Varchar	191	
4	Password	Password	Varchar	191	
5	Akses level	Access_level	Varchar	191	
6	Token	Remember_token	Varchar	100	

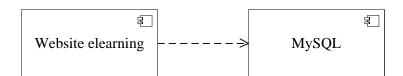
Waktu dibuat	Created_at	Timestamp	-	
Waktu diubah	Updated_at	Timpestamp	-	

### 4.2.2 Software Architecture

Menggambarkan struktur sistem dari program atau sistm komputer yang digambarkan dengan *Component diagram* dan *Deployment diagram*.

#### 1. Component Diagram

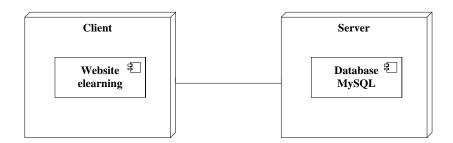
Menggambarkan struktur dan hubungan antar komponen piranti perangkat lunak, termasuk ketergantungan diantaranya. Component diagram juga dapat berupa interface yang berupa kumpulan layanan yang disediakan oleh komponen lainnya.



Gambar IV.16. Component Diagram

## 2. Deployment Diagram

Deployment Diagram menyediakan gambaran bagaimana sistem secara fisik akan terlihat. Sistem diwakili oleh node-node, dimana masing-masing node diwakili oleh sebuah kubus. Garis yang menghubungkan kedua kubus menunjukkan hubungan diantara kedua node tersebut. Berikut gambar Deployment Diagram

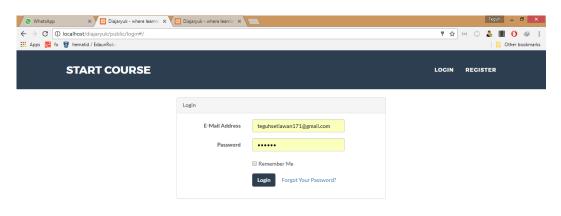


Gambar IV.17. Deployment Diagram

#### 4.2.3 User Interface

#### 1. Form Login

Form yang berfungsi untuk mengotentikasi user.

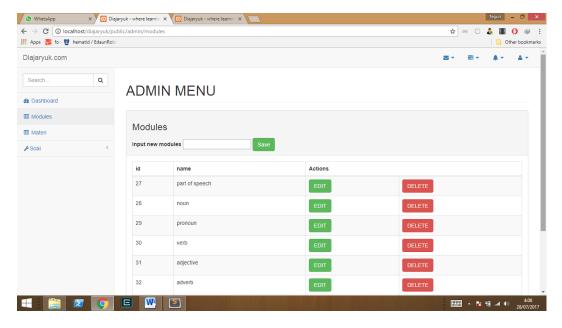




Gambar IV.18. Form Login

### 2. Form Menu Modules

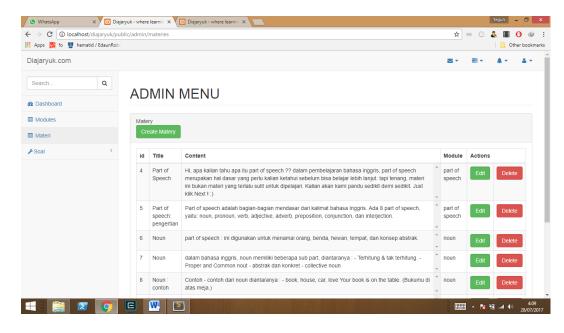
Form Untuk mengolah data modules.



Gambar IV.19. Form Menu Modules

#### 3. Form Data Materi

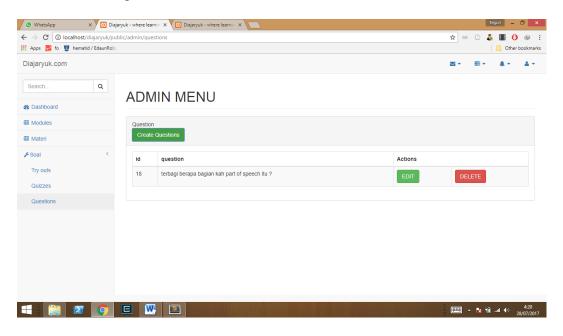
Form untuk mengolah data materi.



Gambar IV.20. Form Data Materi

#### 4. Form Data Soal

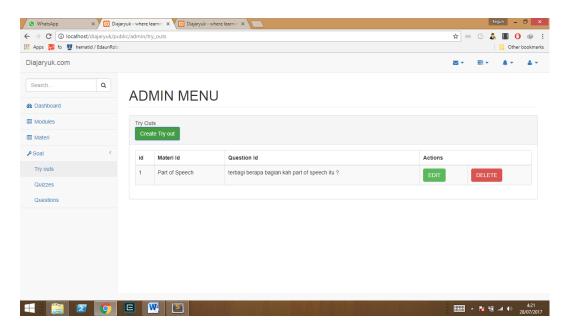
Form untuk mengolah data soal.



Gambar IV.21. Form Data Soal

#### 5. Form Data Try Out

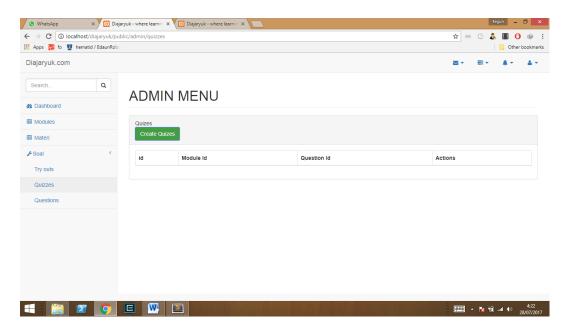
Form untuk mengolah data try out



Gambar IV.22. Form Try Out

#### 6. Form Data Kuis

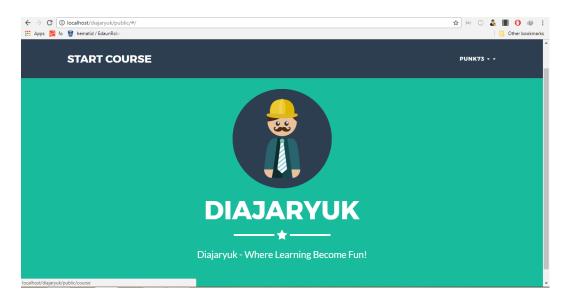
Form untuk mengolah data kuis



Gambar IV.23. Form Data Kuis

#### 7. Form *Home*

Halaman utama website.



Gambar IV.24. Form Home

#### 8. Form Course

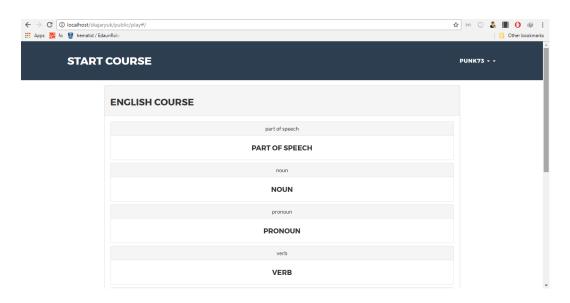
Halaman yang menampilkan overview pembelajaran.



Gambar IV.25. Form Course

## 9. Form Play

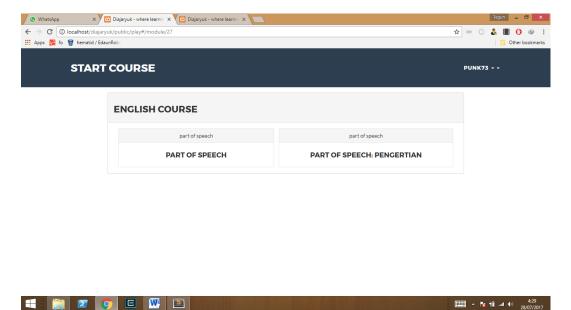
Halaman pembelajaran yang diakses user / siswa.



Gambar IV.26. Form Play

#### 10. Form Module

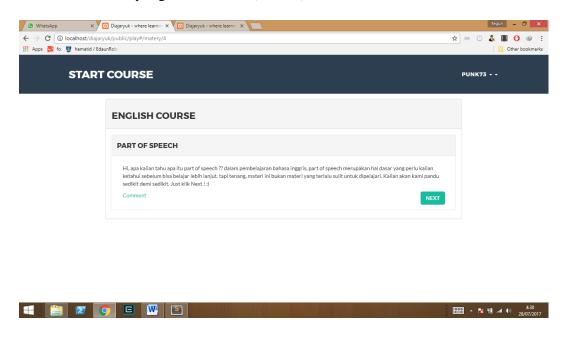
Halaman materi yang terdapat dalam satu module.



Gambar IV.27. Form Module

#### 11. Form Materi

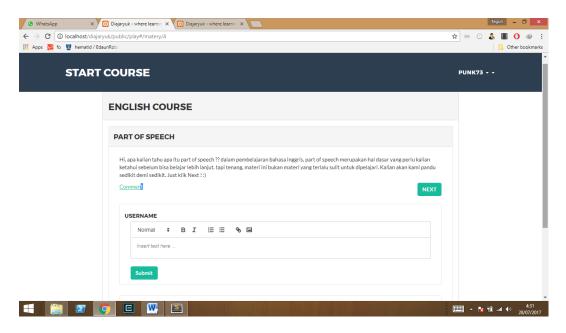
Halaman materi yang diakses user ( siswa )



Gambar IV.28. Form Materi

#### 12. Form Komentar

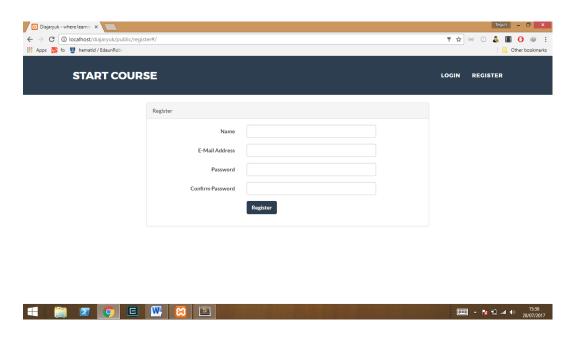
Form user mengisikan komentar.



Gambar IV.29. Form Komentar

## 13. Form Register User

Form registrasi user baru.



Gambar IV.30. Form Register User

#### 4.3 Code Generation

### 1. Form Login

```
<?php
use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;
class LoginController extends Controller
{
  | Login Controller
  | This controller handles authenticating users for the application and
  | redirecting them to your home screen. The controller uses a trait
  | to conveniently provide its functionality to your applications.
  use AuthenticatesUsers;
   * \textit{ Where to redirect users after login.}
   * @var string
   */
  protected $redirectTo = '/home';
   *\ Create\ a\ new\ controller\ instance.
   * @ \textit{return void}
  public function __construct()
     $this->middleware('guest')->except('logout');
```

#### 2. Form Modules

```
app.collections.modules.fetch({
       success: function(collection, response, opt){
       Vue.component('module-item',{
              template: '#module-item-template',
              props: ['id', 'name', 'module', 'modules'],
              methods: {
              editOnClick: function(module){
                      collection.fetch({
                      success(col, response, opt){
                              var tmp = col.get({id: module.id})
                             app.vue.modules.newModule = tmp.toJSON();
                                            },
                                            error(err){
                                                    console.log(err)
                                     })
                              },
                             deleteOnClick: function(module){
                                     var _this = this;
                                     //hapus di local
                                     app.vue.modules.modules = \\
app.vue.modules.modules.filter(function(obj){
                                            // console.log(obj.id !== module.id )
                                            return obj.id !== module.id
                                     })
                                     //hapus di server
                                     collection.fetch({
                                            success(col, response, opt){
                                                    col.get({id: module.id}).destroy({
                                                           success(data, response, opt){
       console.log(response._meta.userMessage)
                                                           },
                                                           error(){
                                                    });
                                            },
                                            error(err){
                                                    console.log(err)
                                     })
```

#### 3. Form Materi

```
app.collections.materies.fetch({
success(materies, response, opt){
app.collections.modules.fetch({
success(modules,response, opt){
Vue.use(VueQuillEditor);
Vue.component('matery-item', {
template: '#matery-item-template',
props: [ 'id', 'title', 'content', 'module', 'matery', 'materies' ],
methods: {
editOnClick: function(matery){
materies.fetch({
success(col,response,opt){
var tmp = col.get({id: matery.id})
app.vue.materies.newMatery = tmp.toJSON();
$("#create-item").modal('toggle');
},
error(err){
console.log(err)
}
})
deleteOnClick: function(matery){
app.vue.materies.materies =
app.vue.materies.materies.filter(function(obj){
return obj.id !== matery.id
//fetch dulu untuk memastikan data bersangkutan sudah ada di materies
lokal.
materies.fetch({
success(col, response, opt){
col.get({id: matery.id}).destroy({
success(data, response, opt){
console.log(response._meta.userMessage)
},
error(err){
console.log(err)
});
},
error(err){
```

```
console.log(err)
})
},
});
app.vue.materies = new Vue({
el: "#materies_place",
data:{
materies: materies.toJSON(),
modules: modules.toJSON(),
newMatery:{
module_id:",
title:",
content:"
},
editorOption:{
theme: "snow",
modules: {
toolbar: [
[{ 'size': ['small', false, 'large'] }],
['bold', 'italic'],
[{ 'list': 'ordered'}, { 'list': 'bullet' }],
['link', 'image']
},
methods:{
getAll(){
var _this = this;
materies.fetch({
success(col, response, opt){
// var jumlah = col.models.length;
// var tmp = [];
/*col.each(function(model, index){
model.getModule(function(newModel){
tmp.push(newModel)
// console.log(tmp.length, col.length )
if( col.length == tmp.length ){
_this.materies = col.toJSON();
console.log(col.toJSON(), _this.materies );
```

```
/*col.models[jumlah-1].getModule(function(model){
console.log('yg terakhir nyampe', col.toJSON() )
_this.materies = col.toJSON();
});*//biar yang baru masuk duluan
console.log( "getAll",col.toJSON() )
_this.materies = col.toJSON();
},
error(err){
console.log(err)
})
},
store(){
var _this = this;
if(this.newMatery.module_id == "" || this.newMatery.title == "" ||
this.newMatery.content == ""){
alert("you need to fill all the data")
return false;
}
materies.create(this.newMatery,{
success(){
_this.getAll();
$('#create-item').modal('toggle');
},
error(err){
console.log(err)
})
modalOnHide(){
this.newMatery = {
module_id:",
title:",
content:"
}
onEditorBlur(editor) {
// console.log('editor blur!', editor)
onEditorFocus(editor) {
```

```
// console.log('editor focus!', editor)
},
onEditorReady(editor) {
//console.log('editor ready!', editor)
},
mounted(){
$("#create-item").on("hidden.bs.modal", this.modalOnHide);
});

},
error(err){
console.log(err)
}
})
},
error(err){
console.log(err)
}
})
```

### 4. Form Soal

```
//Questions
app.collections.questions.fetch({
success(collection, response, opt){
Vue.component('question-item', {
template: '#question-item-template',
props: ['id','name', 'answer', 'question', 'questions'],
methods: {
editOnClick: function(question){
var _this = this;
collection.fetch({
success(col, response, opt){
var tmp = col.get({id: question.id})
console.log(tmp)
app.vue.questions.newQuestion = tmp.toJSON();
tmp.getAnswer(function(jawaban){
       app.vue.questions.answer = jawaban.get('answer');
       $("#create-item").modal('toggle');
```

```
});
},
error(err){
console.log(err)
})
},
deleteOnClick: function(question){
var _this = this;
console.log(question)
//hapus di local
app.vue.questions.questions =
app.vue.questions.questions.filter(function(obj){
// console.log(obj.id !== module.id )
return obj.id !== question.id
})
//hapus di server
collection.fetch({
success(col, response, opt){
col.get({id: question.id}).destroy({
       success(data, response, opt){
               console.log(response._meta.userMessage)
        },
       error(err){
               console.log(err)
        }
});
},
error(err){
console.log(err)
})
},
});
app.vue.questions = new Vue({
el: "#question_place",
data: {
questions: collection.toJSON(),
answer: ",
newQuestion: {
question_type: 'PG', //pilihan Ganda
question: ",
```

```
answer_1:",
answer_2:",
answer_3:",
answer_4:",
point:10,
},
},
methods:{
getAll(){
var _this = this;
collection.fetch({
success(col, response, opt){
_this.questions = col.toJSON();
},
error(err){
console.log(err)
})
},
store(){
console.log(this.newQuestion)
var _this = this;
if( this.newQuestion.question == " || this.newQuestion.answer_1 == " ||
this.newQuestion.answer_2 == " || this.newQuestion.answer_3 == " ||
this.newQuestion.answer_4 == " ){
alert('you need to fulfill the whole data first!')
return false;
if(_this.answer == ""){
alert('you need to choose right answer first!')
return false;
}
collection.create(
this.newQuestion,
success(col, response, opt){
       console.log('success create',{
               col,response,opt
       });
       console.log(_this.answer)
       var id = response.result.id ;//col.get('id');
       col.getAnswer(function(jawaban){
               if(typeof(jawaban) == "undefined"){ //new
```

```
console.log({
                              question_id: id,
                              answer: _this.answer
                       })
                       //untuk bisa update postAnswer ini kita harus kirim
ID.
                       col.postAnswer({
                              question_id: id,
                              answer: _this.answer
                       });
               }else{ // edit
                       col.postAnswer({
                              id: jawaban.get('id'),
                              question_id: id,
                              answer: _this.answer
                       });
       })
        _this.getAll();
       $("#create-item").modal('toggle');
},
error(err){
       console.log(err);
       $("#create-item").modal('toggle');
}
}
);
this.newQuestion = {
question_type: ",
question: ",
answer_1:",
answer_2:",
answer_3:",
answer_4:",
point:0,
},
modalOnHide(){
```

```
console.log('modal hidden');
this.answer =";
this.newQuestion = {
question_type: 'PG', //pilihan Ganda
question: ",
answer_1:",
answer_2:",
answer_3:",
answer_4:",
point:0,
}
},
},
mounted(){
$("#create-item").on("hidden.bs.modal", this.modalOnHide);
});
},
error(err){
console.log(err)
})
```

### 5. Form Try Out

```
app.collections.try_outs.fetch({
success(try_outs,response,opt){
       // console.log(try_outs)
       Vue.component('try_out-item',{
               template: '#try_out-item-template',
               props: [ 'id', 'question', 'materi', 'try_out', 'try_outs' ],
               methods: {
                      editOnClick: function(try_out){
                              var model = try_outs.get({id: try_out.id})
                              app.vue.try_outs.newTryOut = model.toJSON();
                              $("#create-item").modal('toggle');
                      },
                      deleteOnClick: function(try_out){
                              var _this = this;
                              //hapus di local
                              app.vue.try_outs.try_outs =
app.vue.try_outs.try_outs.filter(function(obj){
                                     return obj.id !== try_out.id
                              })
                              //hapus di server
                              try_outs.fetch({
                                     success(col, response, opt){
                                             col.get({id: try_out.id}).destroy({
                                                    success(data, response, opt){
       console.log(response._meta.userMessage)
                                                     },
                                                    error(err){
                                                            console.log(err)
                                                    }
                                             });
                                      },
                                     error(err){
                                             console.log(err)
                              })
                      }
               },
       });
       app.vue.try_outs = new Vue({
```

#### 6. Form Kuis

```
//Quizzes
app.collections.quizzes.fetch({
success(quizzes, response, opt){
app.collections.modules.fetch({
success(modules, response, opt){
app.collections.questions.fetch({
success(questions, response,opt){
       Vue.component('quiz-item',{
               template: '#quiz-item-template',
               props: [ 'id', 'question', 'module', 'quiz', 'quizzes' ],
               methods: {
                      editOnClick: function(quiz){
                              var model = quizzes.get({id: quiz.id})
                              if(typeof(model) !== "undefined" ){
                                     app.vue.quizzes.newQuiz = model.toJSON();
                              }else{
                                     quizzes.fetch({
                                             success(col, response, opt){
                                                    var model = col.get({id: quiz.id})
                                                    app.vue.quizzes.newQuiz =
model.toJSON();
                                             },
                                             error(err){
                                                    console.log(err)
                                     })
                              $("#create-item").modal('toggle');
                      deleteOnClick: function(quiz){
                              var _this = this;
                             //hapus di local
                              app.vue.quizzes.quizzes =
app.vue.quizzes.quizzes.filter(function(obj){
                                     return obj.id !== quiz.id
                              })
                             //hapus di server
                              quizzes.fetch({
                                     success(col, response, opt){
                                             col.get({id: quiz.id}).destroy({
                                                    success(data, response, opt){
```

# **4.4.** *Testing*

# 1. Form Login Administrasi

Tabel IV.20. Hasil Pengujian *Black Box Testing* Form Login Administrasi

No	Sekenario	Test Case	Hasil yang	Hasil	Kesimpulan
	Pengujian		Diharapkan	Pengujian	
1	Email dan password tidak diisi kemudian klik tombol login	Email: (kosong) Password: (kosong)	Sistem menolak dan muncul pesan "please fill out this field"	Sesuai Harapan	Valid
2	Mengetikan Email dan password tidak diisi kemudian klik tombol login	Email: adm01 Password: (kosong)	Sistem menolak dan muncul pesan "please fill out this field"	Sesuai Harapan	Valid
3	Email tidak diisi dan password diisi kemudian klik tombol login	Email: (kosong) Password: 12345	Sistem menolak dan muncul pesan "please fill out this field"	Sesuai Harapan	Valid
4	Mengetikan salah satu kondisi yang salah kemudian klik tombol login	Emai: teguhsetia wan171@ gmail.com (benar) Password: 123 (salah)	Sistem menolak dan muncul pesan "This credential do not match our records"	Sesuai Harapan	Valid
5	Mengetikan username dan password dengan benar kemudian klik tombol login	Username: teguhsetia wan171@ gmail.com (benar) Password: 12345	Sistem menerima akses login kemudian muncul menu home	Sesuai Harapan	Valid

### 2. Form Modules

Tabel IV.21. Hasil Pengujian *Black Box Testing* Modules

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua <i>field</i> tidak diisi.	Field module name: (kosong)	Tampil pesan "you need to fill the module name"	Sesuai harapan	Valid

# 3. Form Materi

Tabel IV.22. Hasil Pengujian *Black Box Testing* Materi

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua <i>field</i> tidak diisi.	Title: (kosong) Content: (kosong) Module_id: (kosong)	Tampil pesan "you need to fill the all data"	Sesuai harapan	Valid
2.	Salah satu field diisi	Title: "Verb" Content: (kosong) Module_id: (kosong)	Tampil pesan "you need to fill the all data"	Sesuai harapan	Valid
3.	Semua field diisi	Title: "Verb" Content: "Verb adalah kata kerja" module_id: 1	Data tersimpan	Sesuai harapan	Valid

# 4. Form Pertanyaan

Tabel IV.23. Hasil Pengujian *Black Box Testing* Form Pertanyaan

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua field tidak diisi.	Question: (kosong) Jawaban A: (kosong) Jawaban B: (kosong) JawabanC: (kosong) Jawaban D: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
2.	Salah satu field diisi	Question: "apa yang dimaksud verb?" Jawaban A: (kosong) Jawaban B: (kosong) JawabanC: (kosong) Jawaban D: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
3.	Semua field diisi	Kode transaksi: KDR001 No RM: DAF160729001 Kode Dokter: D01 Tanggal: 2016- 07-29 Anamnesa: tenggorokan sakit Diagnosa: radang Kode admin: ADM01	Data tersimpan	Sesuai harapan	Valid
4.	Jawaban yang benar tidak dipilih	Radiobox uncheck	Tampil pesan "you need to choose the right answer first!"	Sesuai harapan	Valid

# 5. Form Try Out

Tabel IV.24. Hasil Pengujian *Black Box Testing Form Try Out* 

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua <i>field</i> tidak diisi.	Materi: (kosong) question: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
2.	Salah satu field diisi	Materi: "Part of Speech" question: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
3.	Semua <i>field</i> diisi	Materi: "Part of Speech" question: "apa yang dimaksud part of speech?"	Data tersimpan	Sesuai harapan	Valid

## 6. Form Kuis

Tabel IV.25. Hasil Pengujian *Black Box Testing* Form Kuis

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua <i>field</i> tidak diisi.	Module: (kosong) question: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
2.	Salah satu field diisi	Module: "Part of Speech" question: (kosong)	Tampil pesan "you need to fulfill the whole data first!"	Sesuai harapan	Valid
	Semua <i>field</i> diisi	Module: "Part of Speech" question: "apa yang dimaksud part of speech?"	Data tersimpan	Sesuai harapan	Valid

### 7. Form Registrasi User

Tabel IV.26. Hasil Pengujian *Black Box Testing* Form Registrasi *User* 

No	Skenario Pengujian	Test Case	Hasil Yang Diharapkan	Hasil Pengujian	Kesimpulan
1.	Semua <i>field</i> tidak diisi.	Name: (kosong) Email: (kosong) password: (kosong) Confirm password: (kosong)	Tampil pesan "Plase fill out this field"	Sesuai harapan	Valid
2.	Salah satu field diisi	Name: Teguh Email: (kosong) password: (kosong) Confirm password: (kosong)	Tampil pesan "Plase fill out this field"	Sesuai harapan	Valid
3.	Semua <i>field</i> diisi	Name: Teguh Email: teguh@gmail.co m password: 12345 Confirm password: 12345	Data tersimpan	Sesuai harapan	Valid

### 4.5. Support

#### 4.5.1. Publikasi Web

Perancangan website ini berbasis internet sehingga mengharuskan untuk hosting. Perancangan website ini menggunakan jasa hosting dan domain dari idhostinger.com dalam mempublikasikan website. Jasa yang digunakan dari idhostinger adalah free dengan domain mtsalbarokah.sch.id/diajaryuk/public/

### 4.5.2. Spesifikasi Hardware dan Software

Perangkat keras (*Hardware*) adalah seluruh komponen yang membentuk suatu sistem komputer dan peralatan lainnya yang minimum dan memungkinkan

komputer dapat melaksanakan tugasnya. Bagian penting lain yang mendukung program adalah perangkat lunak (*Software*) yang digunakan dalam mengeksekusi program aplikasi serta sistem aplikasi yang akan digunakan untuk menjalankan program tersebut.

Tabel IV.27. Spesifikasi *Hardware* dan *Software* 

Kebutuhan	Keterangan
Monitor	14"
Processor	Pentium Core 2 Duo, 2.4Ghz
RAM	1 GB
Hardisk	250 GB
Keyboard	108 Key
Printer	Ink Jet
Mouse	Standard
Sistem Operasi	Microsoft Windows 7
Software Pendukung	Browser, Xampp

## 4.6. Spesifikasi Dokumen Sistem Usulan

A. Nama Dokumen : Sertifikat Keikutsertaan

Fungsi : Sertifikasi Pengguna

Sumber : System

Tujuan : Pengguna

Media : PDF

Frekuensi : Setiap *User* menyelesaikan Pembelajaran

Format : Lampiran B-1