

## PROPOSAL TUGAS AKHIR

### **ANALISIS PERANCANGAN SISTEM INFORMASI MANAJEMEN TUGAS AKHIR MENGGUNAKAN *FRAMEWORK CODEIGNITER* (STUDI KASUS: PRODI SISTEM INFORMASI UAJ)**



**Disusun oleh:  
JEVON ADELWIN HURA  
2018-0456-0013**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK  
UNIVERSITAS KATOLIK INDONESIA ATMA JAYA  
JAKARTA  
2021**

## LEMBAR PERSETUJUAN

### PROPOSAL TUGAS AKHIR ANALISIS PERANCANGAN SISTEM INFORMASI MANAJEMEN TUGAS AKHIR MENGGUNAKAN *FRAMEWORK CODEIGNITER* (STUDI KASUS: PRODI SISTEM INFORMASI UAJ)

diajukan oleh:  
Jevon Adelwin Hura  
2018-0456-0013

telah diperiksa dan disetujui oleh:

Pembimbing I



Ferdian Aditya Pratama S.Kom, M.Cs  
120171759 (NIP)  
Tanggal: 9 November 2021

**\*Catatan:**

Jika dosen pembimbing hanya satu orang maka penulisan Pembimbing I, Nama Dosen, NIP, dan tanggal menjadi rata tengah.

## DAFTAR ISI

LEMBAR PERSETUJUAN.....	II
DAFTAR ISI.....	III
DAFTAR TABEL.....	v
DAFTAR GAMBAR.....	vi
BAB 1` PENDAHULUAN.....	7
1.1. LATAR BELAKANG .....	7
1.2. RUMUSAN MASALAH.....	9
1.3. BATASAN MASALAH .....	9
1.4. TUJUAN DAN MANFAAT.....	9
BAB 2 KAJIAN PUSTAKA .....	11
2.1. KAJIAN PUSTAKA .....	11
2.2. LANDASAN TEORI.....	13
2.2.1 Sistem Informasi .....	13
2.2.2 <i>Unified Modeling Language (UML)</i> .....	15
1. <i>Use Case Diagram</i> .....	15
2. <i>Activity Diagram</i> .....	16
3. <i>Class Diagram</i> .....	17
2.2.3 <i>PHP: Hypertext Preprocessor</i> .....	17
2.2.4 <i>HTML: Hypertext Markup Language</i> .....	17
2.2.5 <i>CSS: Cascading Style Sheet</i> .....	19
2.2.6 <i>CodeIgniter</i> .....	20
2.2.7 <i>MySQL</i> .....	22
BAB 3 RANCANGAN PENELITIAN .....	25
3.1. METODOLOGI PENELITIAN.....	25
1. PERANCANGAN <i>USE CASE DIAGRAM</i> .....	27
2. <i>ACTICVITY DIAGRAM</i> MAHASISWA <i>REQUEST</i> BIMBINGAN DOSEN.....	29
3. <i>ACTICVITY DIAGRAM</i> MAHASISWA MELAKUKAN BIMBINGAN .....	30

4.	<i>ACTICVITY DIAGRAM</i> PENGUMPULAN <i>FILE</i> PROPOSAL FINAL .....	32
5.	<i>ACTICVITY DIAGRAM</i> MAHASISWA MENDAFTAR SIDANG .....	33
6.	<i>ACTICVITY DIAGRAM</i> KOORDINATOR TA MENGINPUT NILAI MAHASISWA .....	34
7.	<i>ACTICVITY DIAGRAM</i> KOORDINATOR TA MENGATUR JADWAL SIDANG/SEMINAR. ....	35
8.	PERANCANGAN <i>CLASS DIAGRAM</i> .....	36
3.3	RENCANA DAN JADWAL PENELITIAN.....	42
DAFTAR PUSTAKA .....		43

## DAFTAR TABEL

Tabel 1. Kajian pustaka sebelumnya .....	11
Tabel 2. Tipe data <i>Numeric</i> .....	22
Tabel 3. Tipe data <i>DATE</i> dan <i>TIME</i> .....	23
Tabel 4. Tipe data <i>String</i> .....	23
Tabel 5. Daftar pertanyaan kuesioner <i>usability testing</i> .....	41
Tabel 6. Rencana dan jadwal penelitian .....	42

## DAFTAR GAMBAR

Gambar 2. 1. Elemen <i>Use Case Diagram</i> .....	16
Gambar 2. 2. Elemen <i>Activity Diagram</i> .....	16
Gambar 2. 3 Bentuk <i>Class Diagram</i> .....	17
Gambar 2. 4 Script PHP.....	17
Gambar 2. 5 Hasil <i>Script PHP</i> .....	17
Gambar 2. 6 <i>Output</i> kode HTML .....	18
Gambar 2. 7 Contoh Struktur <i>Tag</i> .....	18
Gambar 2. 8 <i>Output</i> kode CSS.....	19
Gambar 2. 9 Struktur CSS .....	20
Gambar 2. 10 MVC.....	21
Gambar 3. 1 Alur Penelitian .....	25
Gambar 3. 2 Perancangan <i>Use Case Diagram</i> .....	27
Gambar 3. 3 <i>Activity Diagram</i> Mahasiswa <i>request</i> dosen pembimbing.....	29
Gambar 3. 4 <i>Activity Diagram</i> Mahasiswa melakukan bimbingan .....	30
Gambar 3. 5 <i>Activity Diagram</i> pengumpulan <i>file</i> proposal final .....	32
Gambar 3. 6 <i>Acticvity Diagram</i> Mahasiswa mendaftar Sidang .....	33
Gambar 3. 7 Koordinator TA menginput nilai Mahasiswa.....	34
Gambar 3. 8 Koordinator TA mengatur Jadwal Sidang/Seminar. ....	35
Gambar 3. 9 Rancangan <i>Class Diagram</i> .....	36
Gambar 3. 10 SUS <i>score</i> .....	39

# **BAB 1` PENDAHULUAN**

## **1.1. Latar Belakang**

Tugas akhir merupakan salah satu syarat wajib bagi mahasiswa program studi Sistem Informasi Universitas Katolik Indonesia Atma Jaya Jakarta untuk lulus dan mendapatkan gelar sarjana. Fakultas atau Program Studi mempercayakan dosen tertentu untuk membimbing mahasiswa dalam menyusun laporan tugas akhir. Ketika mengambil mata kuliah tugas akhir, mahasiswa memerlukan bimbingan dari dosen pembimbing untuk menulis dan menyelesaikan laporan tugas akhir mahasiswa.

Sebelum menulis laporan tugas akhir, mahasiswa terlebih dahulu menentukan judul yang akan diambil dan menemui dosen pembimbing untuk membicarakan penelitian yang akan di bahas dalam laporan tugas akhir. Ketika pengajuan judul diterima oleh dosen pembimbing mahasiswa dapat memulai mengerjakan proposal tugas akhir. Selama proses penulisan proposal ini, mahasiswa mengumpulkan laporan dan dosen pembimbing memberikan catatan pada laporan yang harus direvisi oleh mahasiswa. Tujuan dari bimbingan ini adalah memperbaiki penulisan pada laporan proposal atau tugas akhir mahasiswa. Setelah mahasiswa selesai sidang proposal, kemudian mahasiswa dapat melanjutkan penulisan laporan tugas akhir atau skripsi.

Proses yang terdapat dalam pengerjaan tugas akhir yang dimulai dari pembuatan proposal hingga skripsi masih dilakukan secara manual. Pengajuan judul oleh mahasiswa, bimbingan penulisan proposal dan skripsi, *upload* laporan serta proses pengumpulan dokumen-dokumen masih dilakukan dalam *platform* yang berbeda-beda

seperti *Microsoft Teams*, *MyAtma*, *Google Drive* hingga *WhatsApp*. Penggunaan *platform* yang berbeda – beda pada proses pengerjaan tugas akhir terjadi karena proses manajemen yang dilakukan secara manual dan belum adanya suatu sistem yang dapat melakukan manajemen proses pengerjaan tugas akhir. Menurut (Sunara & Candra, 2020) tanpa teknologi, proses-proses yang membutuhkan ketelitian dalam prosesnya dapat menyebabkan *human error*. Pembuatan sistem informasi dapat meminimalisir terjadinya *human error* dan meningkatkan efektivitas dan efisiensi kerja (Permana, Ariyadi, & Piriaselina, 2019) .Oleh karena itu, perlu untuk membuat Sistem Informasi Manajemen Tugas Akhir Prodi Sistem Informasi Unika Atma Jaya Jakarta.

CodeIgniter merupakan *framework* PHP yang memakai sistem *Model-View - Controller* (MVC) (Padmanaba, N, & Andayati, 2020). CodeIgniter mempunyai *library* yang lengkap untuk mengerjakan operasi-operasi yang umum, yang dibutuhkan dalam membangun aplikasi berbasis web, seperti mengakses database, validasi *form* sehingga sistem dapat dikembangkan dengan mudah (Sallaby & Kanedi, 2020). CodeIgniter mempermudah *developer* dalam membangun aplikasi *web* berbasis PHP karena sudah memiliki kerangka kerja, sehingga tidak perlu untuk menulis semua kode program. Penggunaan CodeIgniter sangat cocok dalam pengerjaan proyek skala menengah hingga yang besar. Pembuatan Sistem Informasi Manajemen Tugas Akhir Prodi Sistem Informasi Unika Atma Jaya akan menggunakan *framework* CodeIgniter.

Pembuatan Sistem Informasi Manajemen Tugas Akhir Prodi Sistem Informasi Unika Atma Jaya Jakarta berbasis web bertujuan untuk mempermudah mahasiswa maupun dosen dalam berbagai proses yang terlibat untuk mengerjakan tugas akhir yang dilakukan oleh mahasiswa. Sistem informasi berbasis web ini akan dibangun dengan menggunakan *framework* Codeigniter.



## **1.2. Rumusan Masalah**

Berikut merupakan rumusan masalah dari penelitian ini:

1. Bagaimana memetakan proses pengerjaan tugas akhir pada program studi sistem informasi Unika Atma Jaya?
2. Bagaimana cara merancang dan membangun sistem informasi manajemen tugas akhir Program Studi Sistem Informasi Unika Atma Jaya Jakarta berbasis web?
3. Bagaimana mengukur tingkat *usability* pada sistem ini?

## **1.3. Batasan Masalah**

Berikut batasan-batasan yang digunakan dalam penelitian ini, yaitu:

1. Mempunyai aktor pengguna yang terdiri dari mahasiswa, dosen dan Koordinator TA.
2. Sistem yang dibuat berdasarkan proses pengerjaan tugas akhir pada Program Studi Sistem Informasi Unika Atma Jaya.
3. Sistem informasi manajemen tugas akhir dibuat menggunakan *framework* CodeIgniter.

## **1.4. Tujuan dan Manfaat**

Berdasarkan rumusan masalah dalam pembuatan Sistem Informasi Manajemen Tugas Akhir Prodi Sistem Informasi Unika Atma Jaya Jakarta berbasis web, maka penelitian ini bertujuan sebagai berikut:

1. Mengidentifikasi proses-proses yang terlibat dalam pengerjaan tugas akhir.

2. Membuat aplikasi berbasis web yang mampu menampung seluruh proses yang terlibat dalam pengerjaan tugas akhir.
3. Merancang dan membangun Sistem Informasi Manajemen Tugas Akhir Prodi Sistem Informasi Unika Atma Jaya Jakarta berbasis *web*.

## BAB 2 KAJIAN PUSTAKA

### 2.1. Kajian Pustaka

Terdapat penelitian terdahulu yang sudah dilakukan oleh peneliti lainnya, yang dapat menjadi kajian Pustaka untuk penelitian ini. Berikut merupakan kajian Pustaka penelitian terdahulu yang ditampilkan pada Tabel 1.

Tabel 1. Kajian pustaka sebelumnya

No.	Peneliti	Tahun	Judul	Pembahasan
1	Made Harry Dananjaya Adiartika <i>et al.</i>	2020	Pengembangan Sistem Informasi Manajemen Tugas Akhir Program Studi Teknik Informatika Universitas Udayana	Penelitian ini mengembangkan sebuah sistem informasi yang berfokus pada modul manajemen dokumen tugas akhir menggunakan MySQL dan CodeIgniter.
2	Bayu Wijaya Putra <i>et al.</i>	2019	Implementasi Framework CodeIgniter dan RESTful API pada Sistem Manajemen Tugas Akhir	Penelitian ini merancang dan membangun sebuah sistem manajemen tugas akhir menggunakan CodeIgniter, jQuery dan RESTful API

3	Teguh Dwi Julianto <i>et al</i>	2021	Sistem Informasi Manajemen Tugas Akhir di Program Studi Sistem Informasi Universitas Yapis Papua	Penelitian ini bertujuan untuk merancang sistem manajemen tugas akhir yang dirancang dengan metode UML, menganalisa sistem dengan metode <i>PIECES</i> dan melakukan analisis data kepuasan pengguna dan tingkat kepuasan pengguna menggunakan definisi Kaplan dan Norton.
---	---------------------------------------	------	---	--

Penelitian yang dilakukan Made Harry Dananjaya Diartika bertujuan untuk mengembangkan suatu sistem informasi yang berfokus pada modul manajemen tugas akhir. Latar belakang penelitian ini adalah meningkatnya kebutuhan akan fitur dalam mengelola atau manajemen proses administratif tugas akhir Program Studi Teknik Informatika Universitas Udayana. Sistem ini dikembangkan menggunakan *framework* CodeIgniter sebagai *backend* dan diuji dengan pengujian *blackbox* untuk melihat fungsionalitas sistem sudah berjalan dengan baik. Hasil dari penelitian ini adalah suatu sistem informasi manajemen tugas akhir yang dapat mempermudah proses administrasi *upload* dokumen tugas akhir.

Penelitian yang dilakukan oleh Bayu Wijaya Putra bertujuan untuk membangun sistem informasi manajemen tugas akhir dengan *framework* CodeIgniter dan Restful API. Latar belakang penelitian ini berdasarkan informasi dari penelitian sebelumnya pada Codeigniter, jQuery dan RESTful API dalam mengembangkan suatu sistem dengan performa terbaik sehingga berdasarkan studi literatur tersebut diimplementasikan suatu sistem manajemen proyek tugas akhir yang dilaksanakan di perguruan tinggi sehingga sistem ini dapat diukur dengan metode *System Usability Scale* (SUS). Kerangka kerja PHP dapat digunakan dalam pembuatan sistem informasi dari skala kecil hingga skala besar. Bahasa pemrograman PHP memiliki *framework* CodeIgniter yang mendukung teknik pemrograman *model-view-controller* (MVC) sehingga dapat mengurangi waktu pengembangan aplikasi. Hasil penelitian ini menunjukkan bahwa penggunaan *framework* CodeIgniter dan RESTful API terintegrasi dengan baik dan mendapatkan nilai rata-rata SUS mencapai 80,88 dari 100 responden.

Penelitian yang dilakukan Teguh Dwi Julianto bertujuan untuk merancang dan membangun sistem manajemen tugas akhir yang dirancang dengan metode UML, membuat sistem dengan PHP, HTML dan MySQL, menguji sistem dengan metode *PIECES* dan melakukan analisis data kepuasan pengguna dan tingkat kepuasan pengguna menggunakan definisi Kaplan dan Norton. Hasil dari penelitian ini berupa suatu sistem informasi manajemen tugas akhir yang mampu melakukan pengumpulan dokumen atau berkas terkait tugas akhir, seperti pengumpulan dokumen persyaratan tugas akhir, pengajuan judul, ujian proposal, ujian hasil dan ujian tutup.

Dari ketiga penelitian terdahulu memiliki kesamaan dalam beberapa aspek dengan penelitian ini yaitu aplikasi berbasis *web* yang dibangun dengan menggunakan

*framework* CodeIgniter dan MySQL sebagai sistem basis data. Beberapa tujuan penelitian terdahulu juga memiliki kesamaan dengan penelitian ini, yaitu merancang, menganalisa proses dan membangun sistem manajemen tugas akhir. Penelitian yang dilakukan Made Harry Dananjaya Diartika hanya berfokus pada sisi administratif yaitu *upload* dokumen, sedangkan sistem yang akan dibangun nantinya akan memiliki fitur lain seperti pengajuan judul, bimbingan, pendaftaran sidang hingga pemberian nilai pada laporan. Pada penelitian yang dilakukan oleh Bayu Wijaya Putra menggunakan CodeIgniter dan RESTful API, penelitian yang akan dilakukan nantinya hanya menggunakan CodeIgniter sebagai *back-end*. Penelitian yang dilakukan oleh Teguh Dwi Julianto masih menggunakan PHP *native* pada *back-end*-nya sehingga membuat proses pengerjaan lebih lama. Penelitian yang akan dilakukan akan menggunakan *framework* CodeIgniter karena sudah memiliki kerangka kerja yang baik dan mempermudah proses pembuatan sistem informasi.

## **2.2. Landasan Teori**

### **2.2.1 Sistem Informasi**

Sistem Informasi merupakan proses mengolah data menjadi suatu informasi yang dibutuhkan oleh suatu individu maupun organisasi dalam membuat suatu keputusan (Tulodo & Solichin, 2019). Menurut Simangunsong sistem informasi bertujuan untuk menampilkan informasi yang dikelola dalam bentuk sistem (Simangunsong, 2018). Berdasarkan definisi tersebut, sistem informasi dapat didefinisikan sebagai suatu proses pengolahan data untuk menampilkan informasi yang dikelola dalam bentuk sistem yang dibutuhkan untuk pengambilan keputusan

suatu individu maupun organisasi. Menurut Negara, et al., (2021) sistem informasi terdiri dari kumpulan komponen yang disebut sebagai blok dan melakukan interaksi dengan blok-blok tersebut untuk mencapai tujuan tertentu. Blok-blok tersebut sebagai berikut.

1. Blok Masukan(*Input block*)

Masukan merupakan segala sesuatu yang digunakan untuk memasukkan data ke dalam sistem, termasuk metode-metode serta media yang digunakan untuk mengambil data yang akan dimasukkan. Data tersebut dapat berupa dokumen dasar.

2. Blok Model (*model block*)

Blok model merupakan kombinasi dari prosedur, metode matematik , serta logika yang digunakan untuk memanipulasi data masukan dan data yang sebelumnya sudah tersimpan pada basis data dengan cara yang sudah ditentukan sehingga menghasilkan output yang sesuai.

3. Block Keluaran (*output block*)

Blok keluaran merupakan hasil akhir dari suatu sistem informasi yang berupa informasi yang berkualitas serta dokumentasi penting yang berguna untuk semua tingkatan manajemen dan juga para pemakai atau pengguna sistem.

4. Blok Teknologi (*technology block*)

Dalam teknologi sistem informasi dipakai untuk menjalankan prosedur-prosedur sistem, mulai dari menerima masukan, menjalankan *save data*, mengambil atau mengakses data, memproses serta mengirimkan *output*.

5. Blok Basis Data (*database block*)

Blok basis data adalah kumpulan data yang saling berhubungan satu sama lain, yang tersimpan pada *database* dan dapat digunakan untuk dimanipulasi oleh perangkat lunak dalam komputer.

#### 6. Blok Kendali (*control block*)

Blok kendali merupakan rencana yang dirancang dan diterapkan untuk meyakinkan bahwa hal-hal yang dapat merusak sistem bisa dicegah dan apabila terjadi kerusakan dapat langsung diatasi.

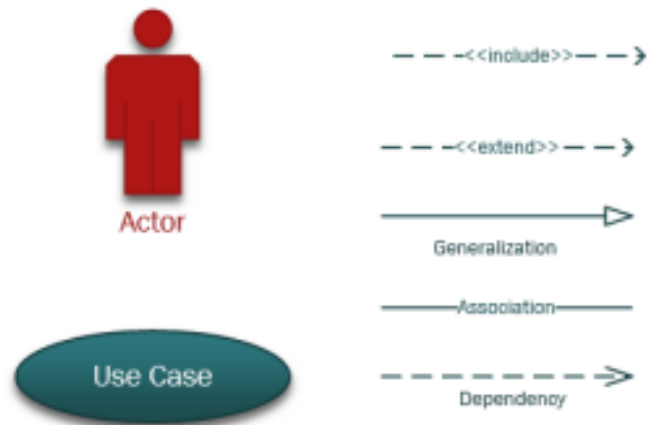
### 2.2.2 *Unified Modeling Language (UML)*

Menurut Prihandoyo (2018), *Unified Modeling Language (UML)* merupakan salah satu metode pemodelan visual yang digunakan dalam perancangan dan pembuatan sebuah *software* yang berorientasikan pada objek. UML merupakan standar penulisan yang didalamnya terdapat bisnis proses, penulisan kelas-kelas dalam sebuah bahasa yang spesifik. Berikut beberapa diagram UML yang digunakan dalam pengembangan sistem, yaitu:

#### 1. *Use Case Diagram*

*Use Case Diagram* merupakan salah satu jenis diagram UML yang dipakai dalam pengembangan sebuah sistem. Menurut Ionescu & Nica (2020) *Use Case* merupakan kumpulan skenario yang mewakili kemungkinan komunikasi yang terjadi antara sistem dan aktor eksternal untuk tujuan tertentu. Pada Gambar 2.1 terdapat elemen-elemen dalam merancang *use case diagram*.

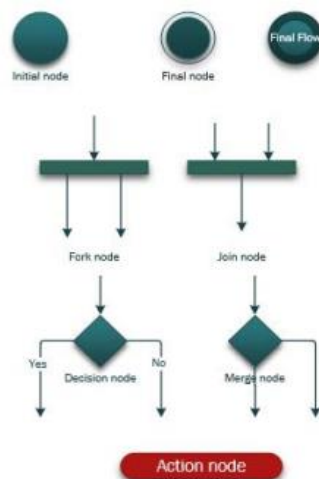




Gambar 2. 1. Elemen *Use Case Diagram*  
(Sumber: Nica & Ionescu, 2020)

## 2. *Activity Diagram*

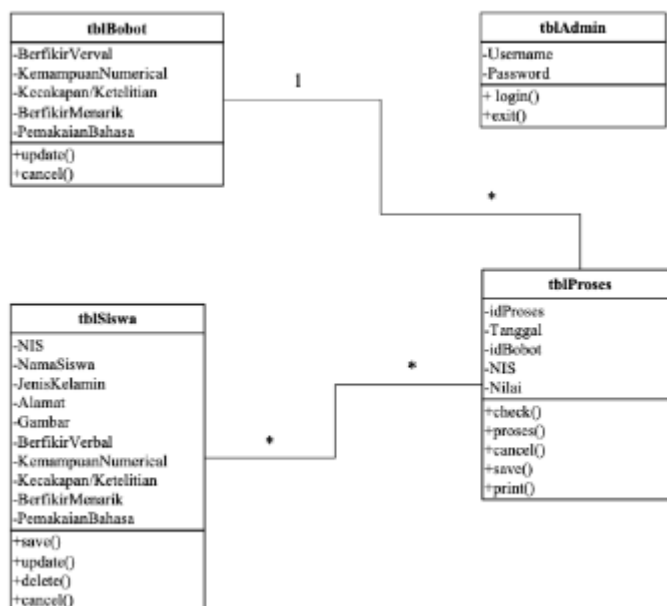
Menurut Ionescu & Nica (2020), *Activity Diagram* merupakan pemodelan urutan tindakan yang dijalankan oleh elemen sistem dan menunjukkan bisnis serta alur kegiatan. Pada Gambar 2.2 terdapat elemen-elemen dalam merancang *activity diagram*.



Gambar 2. 2. Elemen *Activity Diagram*  
(Sumber: Nica & Ionescu, 2020)

### 3. Class Diagram

*Class Diagram* merupakan gambaran struktur dan deskripsi dari *class*, *package* dan *object* yang saling berhubungan seperti pewarisan, asosiasi dan lainnya (Alfina & Harahap, 2019). Pada *class diagram* terdapat nama *class*, *attributes*, *operations* serta *association* (hubungan antar-*class*). Pada Gambar 2.3 dapat dilihat bentuk dari *class diagram*.

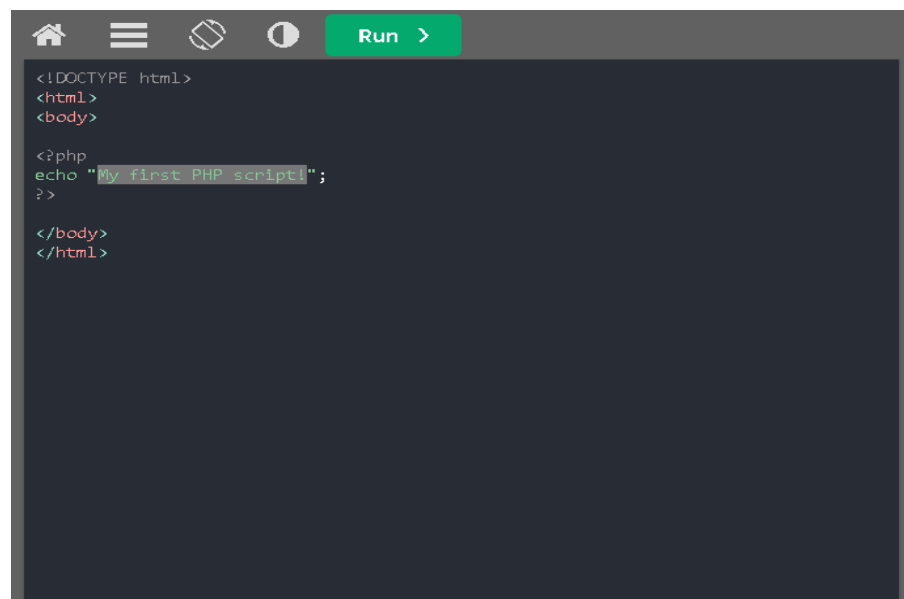


Gambar 2. 3 Bentuk *Class Diagram*

(Sumber: Alfina & Harahap, 20

### 2.2.3 PHP: *Hypertext Preprocessor*

*Hypertext Pre-Processor* (PHP) telah menjadi bahasa pemrograman penting bagi pengembang yang bekerja pada bidang *Software Development* untuk membuat *website*. PHP dimulai sebagai proyek *open source* kecil yang berkembang pesat dan banyak digunakan untuk mengembangkan dan membangun situs *web* dan proyek yang lebih besar. Pada umumnya, bahasa pemrograman PHP merupakan bahasa pemrograman yang dirancang untuk mengembangkan aplikasi web dan digunakan untuk menghasilkan HTML yang dinamis (Šimec, Lozić, & Golubić, 2017). PHP juga dipakai sebagai *server-side scripting* yang dilengkapi dengan HTML untuk mengelola konten yang dinamis, membangun situs seperti *e-commerce* dan terintegrasi dengan MySQL dan Oracle (Lamsal, 2020). Contoh penulisan *scripting* sederhana dengan bahasa PHP dapat dilihat pada Gambar 2.4 dan 2.5.

A screenshot of a code editor window with a dark theme. The editor contains a PHP script embedded within an HTML document. The code is as follows:

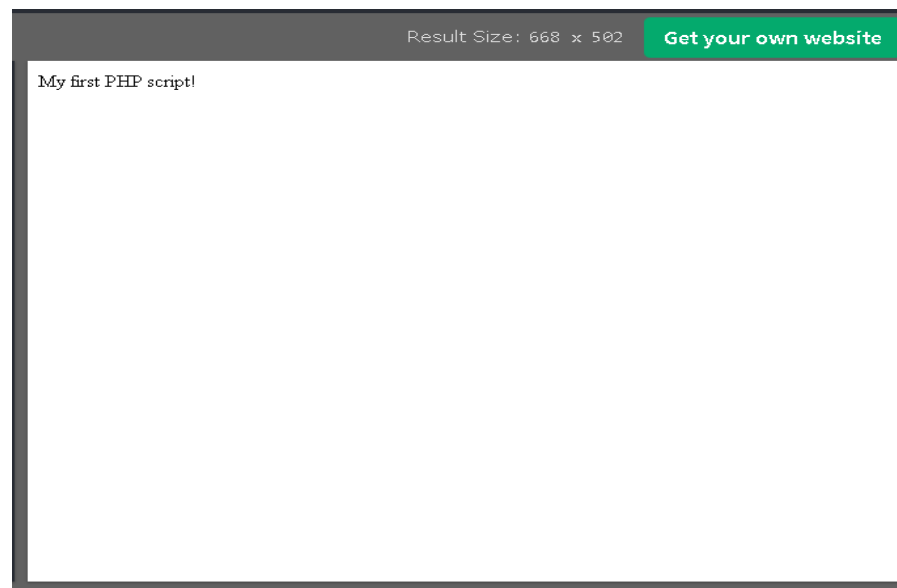
```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

The editor has a toolbar at the top with icons for home, list, undo, redo, and a green 'Run' button with a right arrow.

Gambar 2. 4 Script PHP



Gambar 2. 5 Hasil *Script* PHP

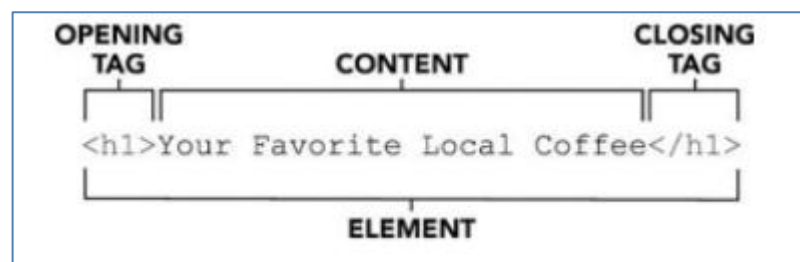
#### 2.2.4 HTML: *Hypertext Markup Language*

HTML atau *Hypertext Markup Language* merupakan bahasa *markup* yang membuat sebuah *web* berjalan. HTML mendefenisikan struktur dasar halaman *web*. Halaman *web* mengandung sebuah *text file* yang ditulis dengan HTML dan berisi kumpulan instruksi untuk mengatur format konten yang terdapat pada suatu halaman *web* (DuRocher, 2021). Pada Gambar 2.6 merupakan contoh penulisan di dalam *text file* yang mempunyai format HTML dan mengandung beberapa instruksi HTML yang menghasilkan sebuah konten di dalam halaman *web*.



Gambar 2. 6 *Output* kode HTML  
(Sumber: DuRocher, 2021)

*Tag* dalam HTML mengandung *tag* pembuka, konten dan *tag* penutup. Pada Gambar 2.7 merupakan sebuah *tag* yang mendefenisikan sebuah *Heading*. *Tag* `<h1>` merupakan *tag* pembuka dan `</h1>` merupakan tag penutup yang memiliki tanda garis miring sebelum penulisan h1. Di tengah kedua *tag* tersebut dapat diisikan suatu konten yang akan ditampilkan ketika diakses dengan *browser*.



Gambar 2. 7 Contoh Struktur *Tag*  
(Sumber: DuRocher, 2021)

### 2.2.5 CSS: *Cascading Style Sheet*

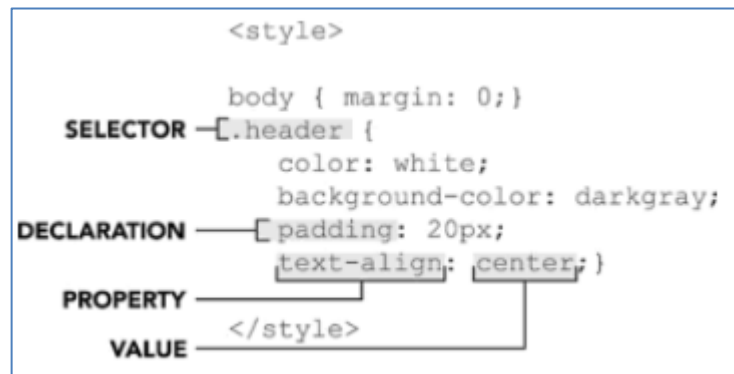
*Cascading Style Sheets* atau CSS merupakan bahasa *markup* yang berfungsi untuk memformat struktur dan tampilan *website* agar terlihat menarik secara visual dan dapat digunakan (DuRocher, 2021). Penggunaan CSS berfungsi pada kode HTML untuk merubah tampilam format dari HTML. Pada Gambar 2.8 merupakan *output* dari penulisan CSS dan HTML.



Gambar 2. 8 *Output* kode CSS

(Sumber: DuRocher, 2021)

CSS memiliki sebuah struktur yang mirip dengan HTML Pada Gambar 2.9 terlihat struktur dari CSS yang memiliki *selector*, *decalaration*, *property* dan *value*. *Selector* merupakan suatu kode untuk memilih elemen pada HTML yang akan dimodifikasi. Pada bagian *declaration* berisi kumpulan *property* untuk memodifikasi lemen HTML tersebut seperti *teks colour*, *centering*, *font style* dan lain-lain. Pada bagian *property* terdapat *value* atau nilai yang diberikan pada *property* seperti *bold*, *italic* atau *text alignment*.



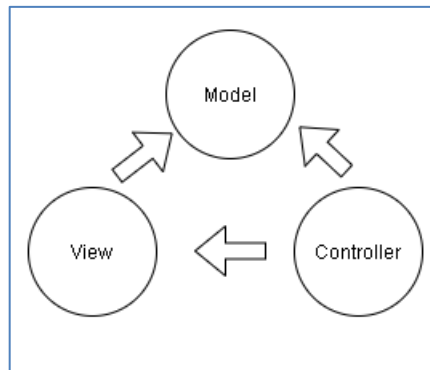
Gambar 2. 9 Struktur CSS

(Sumber: DuRocher, 2021)

### 2.2.6 CodeIgniter

CodeIgniter merupakan *framework* bahasa pemrograman PHP yang memiliki kerangka kerja yang kuat dan dibangun untuk pengembang yang membutuhkan *toolkit* sederhana dan elegan untuk membuat aplikasi berbasis web dengan fitur yang lengkap. (Codeigniter Web Framework, 2021). CodeIgniter dapat membangun aplikasi PHP dinamis dengan menggunakan konsep *Model-View-Controller* (MVC) dan menyediakan berbagai macam *library* yang mempermudah dan mempercepat pengembangan *web* (Erinton, Negara, & Sanjoyo, 2017).

#### 1. MVC(*Model-View-Controller*)



Gambar 2. 10 MVC

MVC merupakan sebuah arsitektur untuk membuat sebuah program (Kelen & Belalawe, 2018). Model MVC mampu mengembangkan *website* secara efektif dan efisien dilihat dari sudut pandang pengembangan, pembagian tugas, dokumentasi, pengujian dan pemeliharaan (Riana, Sanjaya, & Kalsoem, 2018). Gambar 2.10 merupakan gambar yang menunjukkan hubungan antara *Model*, *View* dan *Controller*. Menurut Supardi & Hermawan, 2018 terdapat 3 jenis komponen yang membangun suatu MVC yaitu sebagai berikut:

1. *View*

*View* merupakan bagian yang menangani *presentation logic*. Pada umumnya, *view* merupakan suatu *file template HTML* yang diatur oleh *Controller*. *View* berfungsi untuk menerima dan merepresentasikan data kepada *user*. *View* tidak mempunyai akses langsung terhadap *Model*.

2. *Model*

*Model* merupakan bagian yang berhubungan langsung dengan untuk melakukan manipulasi data, menangani validasi dari bagian controller tetapi tidak berhubungan langsung dengan *view*.



### 3. *Controller*

*Controller* merupakan bagian yang digunakan dalam menangani interaksi pengguna, bekerja dengan model dan memilih *view* yang akan digunakan untuk menampilkan data. *Controller* berfungsi untuk menerima dan *request* dan data dari *user* kemudian menentukan proses yang akan dilakukan aplikasi

#### 2.2.7 MySQL

MySQL merupakan salah satu sistem RDBMS (*Relational Database Management System*) yang menggunakan SQL sebagai standar bahasanya. SQL atau *Structured Query Language* merupakan bahasa yang digunakan untuk melakukan manipulasi, mengambil , *insert*, *update* dan *delete* data pada sistem RDBMS (Mehta, Bhavsar, Oza, & Shah, 2018). SQL memungkinkan *user* untuk mengakses data dari MySQL dengan perintah dan memanipulasi data. Penggunaan SQL dalam bahasa pemrograman lain dapat digunakan dengan memanfaatkan modul, *library* dan *precompiler* yang dapat membantu untuk melakukan *create/drop database* dan *table*, membuat *view* serta aktivitas lainnya dalam *database*. Menurut Mehta *et al* (2018) jumlah penyimpanan yang dibutuhkan untuk setiap tipe data berbeda – beda pada MySQL. Jumlah penyimpanan pada tiap tipe data dapat dilihat pada tabel berikut.

Tabel 2. Tipe data *Numeric*

<b>Tipe Data</b>	<b>Penyimpanan</b>
TINYINT	1 byte
SMALLINT	2 bytes

MEDIUMINT	3 bytes
INT, INTEGER	4 bytes
BIGINT	8 bytes
FLOAT	4 bytes
DOUBLE	8 bytes
DECIMAL	bervariasi
BIT(M)	7-8 bytes

Pada Tabel 2. merupakan penyimpanan yang dibutuhkan pada tipe data *Numeric*

Tabel 3. Tipe data *DATE* dan *TIME*

<b>Tipe Data</b>	<b>Penyimpanan</b>
YEAR	1 byte
DATE	3 bytes
TIME	3 bytes + <i>fractional seconds storage</i>
DATETIME	5 bytes + <i>fractional seconds storage</i>
TIMESTAMPS	4 bytes + <i>fractional seconds storage</i>

Tabel 3 merupakan penyimpanan yang dibutuhkan untuk tipe data *DATE* dan *TIME*.

Tabel 4. Tipe data *String*

<b>Data Type</b>	<b>Penyimpanan</b>
CHAR (M)	$M \times w$ bytes , $0 \leq M \leq 255$ $w$ = jumlah ukuran byte

BINARY(M)	M bytes, $0 \leq M \leq 255$
VARCHAR (M), VARBINARY (M)	L + 1 bytes
TINYBLOB, TINYTEXT	L + 1 bytes, $L < 28$
MEDIUMBLOB, MEDIUMTEXT	L+3 bytes, $L < 224$
ENUM('value', 'value2', ...)	1 atau 2 bytes tergantung nilai <i>enumeration</i> (maksimal 65,535 nilai)
SET('value1', 'value2', ...)	1,2,3,4, atau 8 bytes tergantung jumlah <i>member</i> (maksimal 64 member)

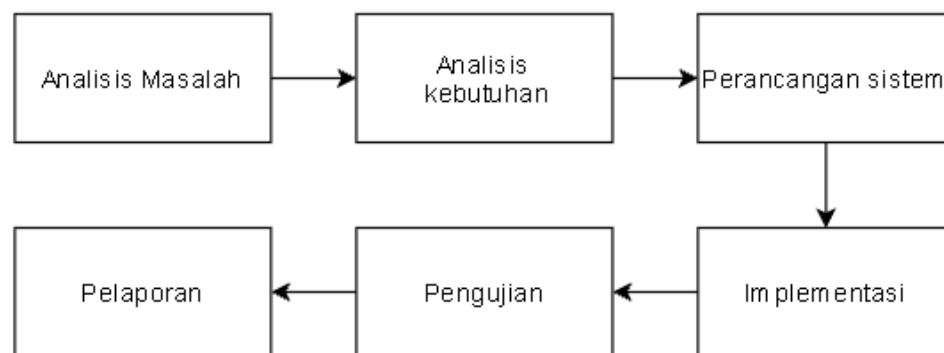
Tabel 4 merupakan penyimpanan yang dibutuhkan untuk tipe data *string*.

## BAB 3

### RANCANGAN PENELITIAN

#### 3.1. Metodologi Penelitian

Dalam proses penelitian yang akan dilakukan, terdapat beberapa tahapan untuk melakukan penelitian. Alur penelitian dapat dilihat pada Gambar 3.1.



Gambar 3. 1 Alur Penelitian

Berikut merupakan penjelasan dari setiap tahap penelitian.

##### 3.1.1 Analisis masalah

Pada tahap ini melakukan identifikasi masalah yang terdapat dalam pengerjaan tugas akhir. Masalah utama yang terdapat pada pengerjaan tugas akhir adalah proses pengerjaan yang masih dilakukan secara manual. Penggunaan *platform* yang berbeda – beda pada proses pengerjaan tugas akhir terjadi karena proses manajemen yang dilakukan secara manual dan belum adanya suatu sistem yang dapat melakukan manajemen proses pengerjaan tugas akhir. Menurut (Sunara & Candra, 2020) tanpa teknologi, proses-proses yang membutuhkan ketelitian dalam prosesnya dapat menyebabkan *human error*. Pembuatan sistem informasi dapat meminimalisir terjadinya *human error* dan meningkatkan efektivitas dan efisiensi kerja (Permana,

Ariyadi, & Piriaselina, 2019) .Oleh karena itu perlu untuk membangun sebuah sistem yang terintegrasi agar proses yang dilakukan secara manual dapat dilakukan dalam satu sistem manajemen.

### **3.1.2 Analisis kebutuhan**

Analisis kebutuhan dilakukan dengan cara menganalisis proses yang terjadi dalam pengerjaan tugas akhir melalui studi literatur dengan cara meneliti dan membaca panduan penyusunan tugas akhir, dokumen-dokumen yang ada atau sumber tertulis lainnya yang relevan dengan proses pengerjaan tugas akhir.

### **3.1.3 Perancangan sistem**

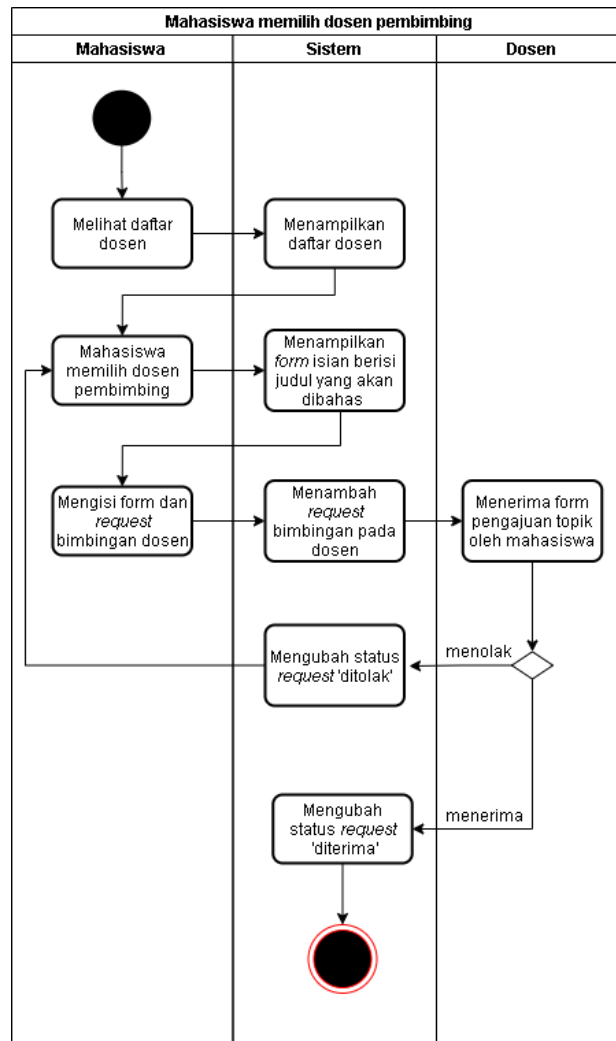
Berdasarkan hasil analisis masalah dan kebutuhan, selanjutnya memetakan proses-proses yang terjadi dan merancang sistem dengan membuat *use case diagram*, *class diagram* dan *activity diagram*.



seminar/ujian TA mahasiswa dapat melihat hasil nilai dari sidang yang telah dilakukan. Mahasiswa juga dapat melihat jadwal sidang proposal atau sidang TA secara keseluruhan dari fitur jadwal.

2. **Dosen Pembimbing:** Dosen pembimbing harus melakukan *login* terlebih dahulu agar dapat mengakses fitur yang terdapat di dalam sistem. Didalam sistem, Dosen pembimbing dapat memilih menerima atau menolak bimbingan dan membimbing mahasiswa dalam penulisan proposal atau laporan tugas akhir. Pada bagian bimbingan, Dosen pembimbing dapat mengunggah judul topik tugas akhir dan melakukan revisi pada bimbingan yang dilakukan oleh mahasiswa. Dosen pembimbing dapat melihat laporan yang di unggah oleh mahasiswa dan memberikan komentar untuk revisi dari laporan yang dikirimkan oleh mahasiswa.
3. **Koordinator TA:** Koordinator TA harus melakukan *login* terlebih dahulu agar dapat mengakses fitur yang terdapat di dalam sistem. Didalam sistem, Koordinator TA dapat mengelola pendaftaran sidang atau pengumpulan *file* proposal untuk sidang pada Pendaftaran Sidang TA/ Seminar Proposal. Pada bagian ini Koordinator TA dapat memeriksa jika dokumen atau persyaratan untuk pendaftaran telah dipenuhi atau lengkap dan dapat mengubah status pendaftaran. Pada Ujian seminar/ujian TA, Koordinator TA dapat menyusun jadwal Seminar proposal atau Sidang TA dan memasukkan nilai akhir bila mahasiswa telah selesai melakukan sidang.

## 2. Activity Diagram Mahasiswa Request bimbingan Dosen



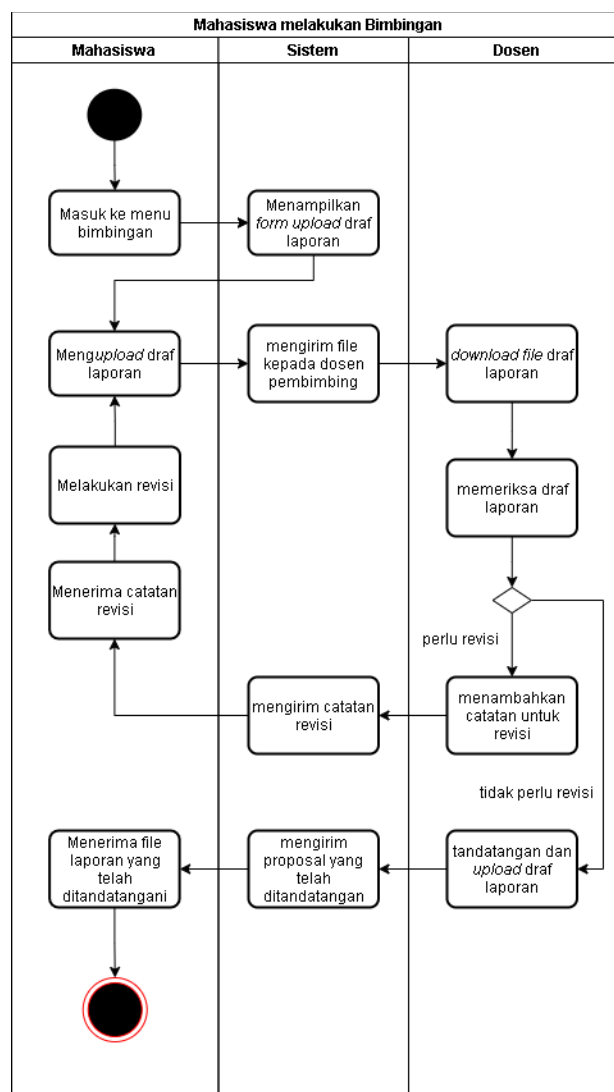
Gambar 3.3. Activity Diagram Mahasiswa request dosen pembimbing

Terlihat pada Gambar 3.3, merupakan alur mahasiswa melakukan *request* bimbingan kepada dosen yang dibuat menggunakan *activity diagram*. Pada proses mahasiswa *request* bimbingan kepada dosen, terdapat tiga aktor yang terlibat yaitu mahasiswa, sistem dan dosen. Alur proses ini diawali dengan mahasiswa masuk ke dalam sistem dan melihat daftar dosen. Sistem mengirimkan daftar dosen kepada mahasiswa dan mahasiswa dapat melihat daftar dosen pembimbing serta mengajukan untuk meminta dosen bersedia untuk menjadi dosen pembimbing. Di dalam sistem



terdapat *form* yang harus diisi oleh mahasiswa untuk mengajukan judul dan meminta dosen untuk bersedia menjadi dosen pembimbing. Jika dosen pembimbing bersedia, maka mahasiswa dapat melakukan bimbingan, jika tidak maka mahasiswa dapat mengganti judul atau memilih dosen pembimbing yang lain.

### 3. Activity Diagram Mahasiswa melakukan bimbingan

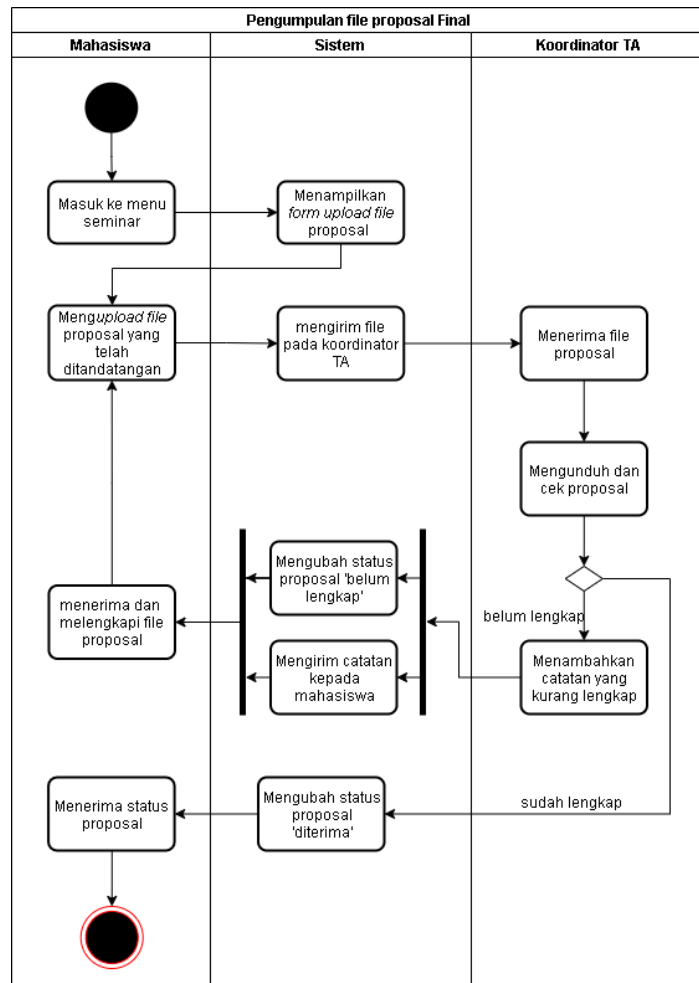


Gambar 3. 4. Activity Diagram Mahasiswa melakukan bimbingan

Jika mahasiswa telah berhasil mendapatkan dosen pembimbing, maka mahasiswa dapat melakukan bimbingan dengan dosen pembimbing. Alur proses

bimbingan dapat dilihat pada Gambar 3.4. Pada Gambar tersebut, diawali dengan mahasiswa masuk ke dalam menu bimbingan, di dalam menu bimbingan terdapat sebuah *form* untuk melakukan bimbingan. Di dalam sistem ini, mahasiswa dapat mengunggah *file* proposal yang telah dikerjakan dan nantinya dosen pembimbing akan memberikan komentar untuk revisi yang akan dilakukan oleh mahasiswa. Proses ini akan terjadi berulang kali jika penulisan proposal yang dilakukan oleh mahasiswa masih terdapat kesalahan penulisan atau belum lengkap. Jika proposal yang telah dibuat oleh mahasiswa sudah memenuhi persyaratan, maka dosen pembimbing dapat menandatangani *file* proposal tersebut secara *digital* dan mengunggahnya di dalam *form* tersebut. Mahasiswa dapat mengunduh *file* tersebut untuk mengumpulkannya ke koordinator tugas akhir.

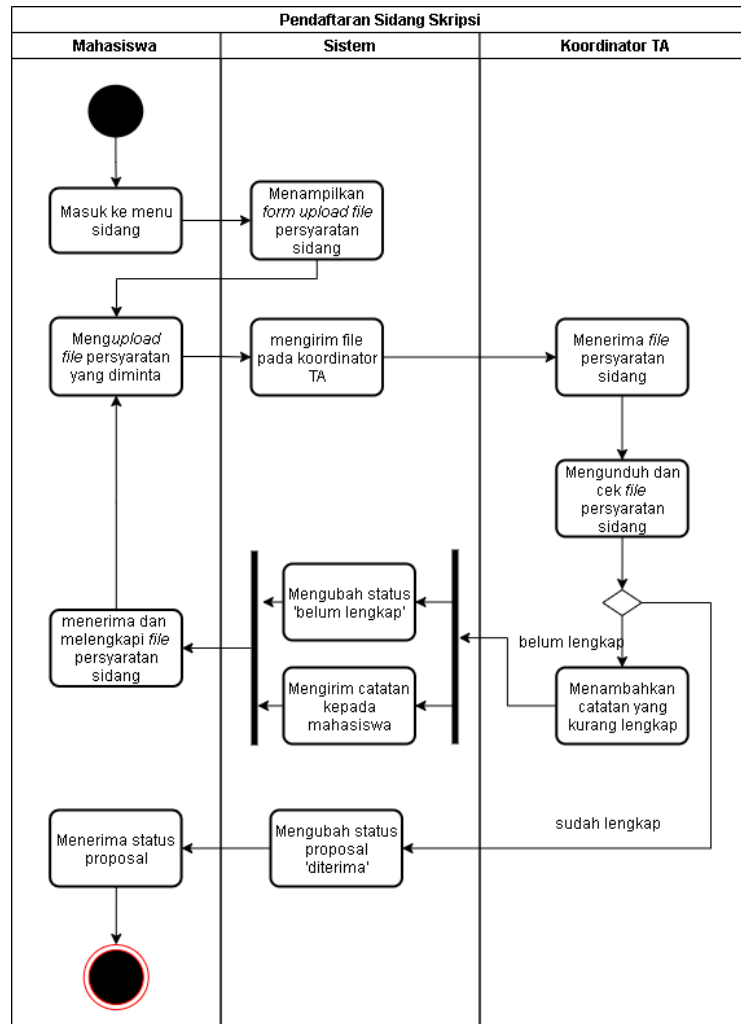
#### 4. Activity Diagram pengumpulan file proposal final



Gambar 3. 5. Activity Diagram pengumpulan file proposal final

Gambar 3.5 merupakan proses pengumpulan *file* proposal final. Proses ini diawali dengan mahasiswa masuk ke menu seminar, kemudian mengunggah *file* yang diminta oleh sistem. Sistem mengirim *file* tersebut kepada koordinator TA. Koordinator TA memeriksa jika mahasiswa telah mengunggah *file* yang diminta sudah memenuhi syarat untuk mendaftarkan. Jika *file* tersebut telah memenuhi syarat maka Koordinator TA dapat menerima *file* tersebut dan sistem mengubah status telah diterima. Jika belum maka mahasiswa akan diminta untuk kembali mengunggah *file* yang diminta

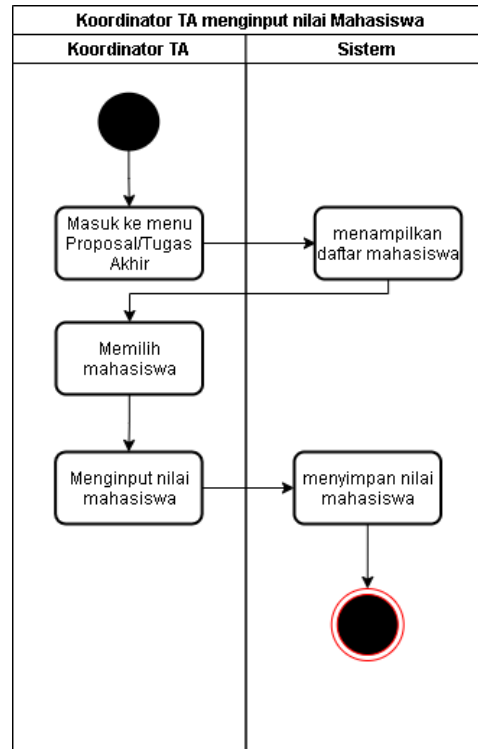
## 5. Activity Diagram Mahasiswa mendaftar sidang



Gambar 3. 6. Activity Diagram Mahasiswa mendaftar Sidang

Gambar 3.6 merupakan proses mahasiswa mendaftar sidang. Proses ini diawali dengan mahasiswa masuk ke menu sidang, kemudian mengunggah *file* yang diminta oleh sistem. Sistem mengirim *file* tersebut kepada koordinator TA. Koordinator TA memeriksa jika mahasiswa telah mengunggah *file* yang diminta sudah memenuhi syarat untuk mendaftarkan. Jika *file* tersebut telah memenuhi syarat maka Koordinator TA dapat menerima *file* tersebut dan sistem mengubah status telah diterima. Jika belum maka mahasiswa akan diminta untuk kembali mengunggah *file* yang diminta.

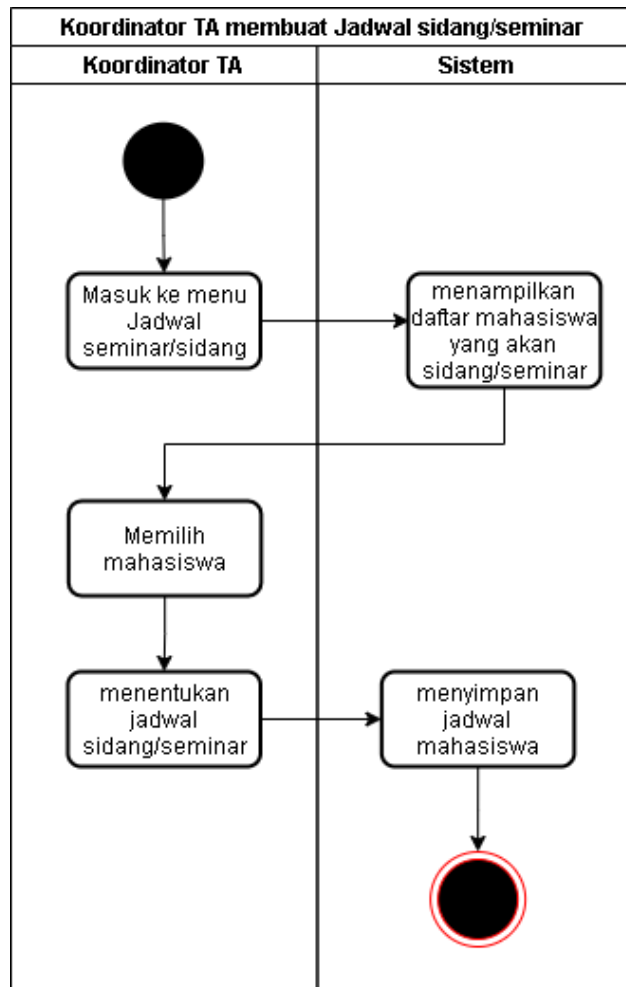
## 6. Activity Diagram Koordinator TA menginput nilai Mahasiswa



Gambar 3. 7. Koordinator TA menginput nilai Mahasiswa

Penginputan nilai dilakukan oleh Koordinator TA, terlihat pada Gambar 3.7, koordinator TA masuk ke dalam menu Proposal atau Tugas Akhir dan sistem akan menampilkan daftar mahasiswa. Selanjutnya, koordinator TA akan memilih mahasiswa dan melakukan *input* nilai mahasiswa. Sistem menyimpan nilai mahasiswa ke dalam *database*.

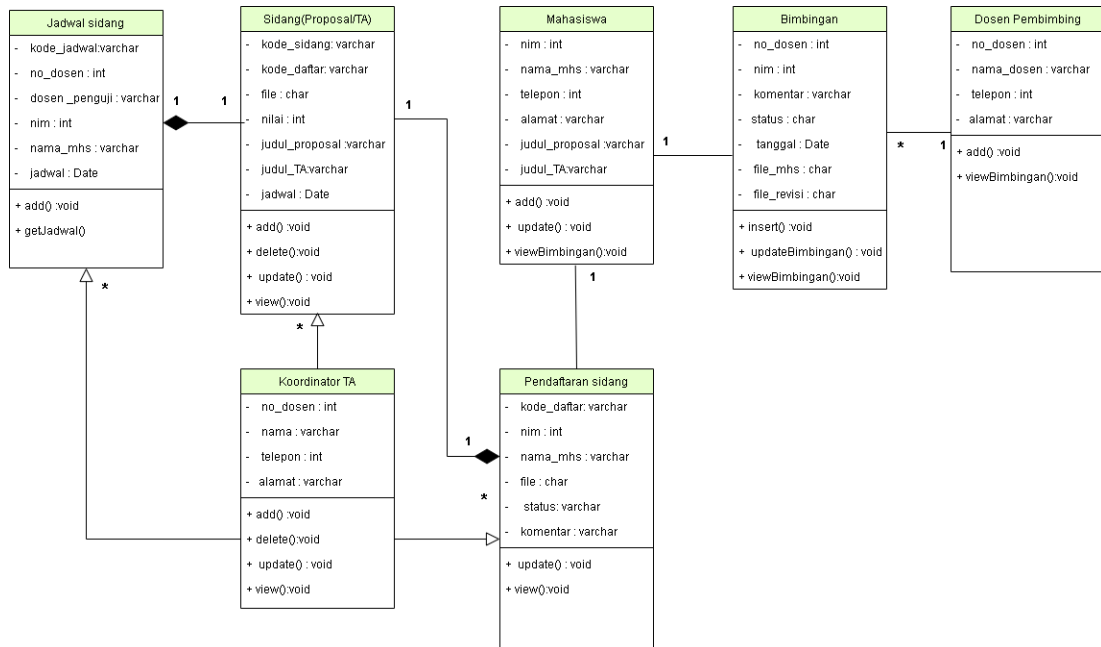
## 7. Activity Diagram Koordinator TA mengatur Jadwal Sidang/Seminar.



Gambar 3. 8 Koordinator TA mengatur Jadwal Sidang/Seminar.

Penyusunan jadwal sidang/seminar dilakukan oleh Koordinator TA. Terlihat pada Gambar 3.8 merupakan proses koordinator TA mengatur jadwal sidang atau seminar. Koordinator TA masuk pada menu jadwal seminar/sidang kemudian sistem akan menampilkan daftar mahasiswa yang akan melakukan sidang/seminar. Koordinator TA kemudian memilih mahasiswa dan meentukan jadwal seminar/sidangnya. Sistem akan menyimpan data jadwal di dalam *database*.

## 8. Perancangan Class Diagram



Gambar 3. 9 Rancangan Class Diagram

Gambar 3.9 merupakan rancangan *class diagram* dari sistem yang akan dibuat.

Pada Gambar 3.9 terdapat beberapa kelas di dalam sistem ini yaitu:

1. **Mahasiswa:** Mahasiswa memiliki beberapa properti seperti nim, nama\_mhs, telepon, alamat, judul\_proposal, dan judul\_TA. Mahasiswa memiliki *method* seperti *add()*, *update()* dan *view()*. Pada *method add()* mahasiswa dapat melakukan bimbingan baru. Pada *Method viewBimbingan()*, mahasiswa dapat melihat tampilan pada bimbingan.
2. **Dosen Pembimbing:** Dosen pembimbing memiliki properti no\_dosen, nama\_dosen, telepon dan alamat. Pada properti dosen pembimbing mempunyai *method add ()* untuk menambahkan *file* atau komentar dan *method viewBimbingan()* untuk melihat bimbingan.

3. **Bimbingan:** Pada bimbingan terdapat beberapa properti seperti `no_dosen`, `nim`, `komentar`, `status`, `tanggal`, `file_mhs` dan `file_revisi`. Method yang dimiliki oleh bimbingan adalah `insert()` untuk menambahkan bimbingan, `updateBimbingan()` untuk memperbaharui `file` atau `komentar` pada bimbingan dan `viewBimbingan()`, untuk melihat isi bimbingan.
4. **Koordinator TA:** Pada Koordinator TA terdapat property seperti `no_dosen`, `nama`, `telepon` dan `alamat`. Koordinator TA memiliki *method* `add()`, `delete()`, `update()` dan `view()`. *Method* yang terdapat pada Koordinator TA ini berfungsi untuk melakukan interaksi dengan relasi pada kelas Pendaftaran sidang dan Sidang (Proposal/TA).
5. **Pendaftaran Sidang:** Pendaftaran sidang memiliki property `kode_daftar`, `nim`, `nama_mhs`, `file`, `status` dan `komentar`. Kelas Pendaftaran sidang memiliki *method* `update()` untuk melakukan *update* properti yang ada pada Pendaftaran sidang dan `view()` untuk melihat isi dari Pendaftaran sidang.
6. **Sidang (Proposal/TA):** Sidang(Proposal/TA) memiliki properti `kode_sidang`, `kode_daftar`, `file`, `nilai`, `judul_proposal`, `judul_TA` dan `jadwal`. *Method* yang terdapat pada kelas ini adalah `add()`, `delete()`, `update()` dan `view()`. *Method* yang terdapat pada kelas ini berfungsi untuk melakukan interaksi dengan relasi pada kelas Jadwal sidang, Koordinator TA dan pendaftaran Sidang.
7. **Jadwal Sidang:** Kelas Jadwal sidang memiliki property `kode_jadwal` `no_dose`, `dosen_penguji`, `nim`, `nama_mhs` dan `jadwal`. Kelas jadwal sidang memiliki *metdhod* `add()` untuk menambahkan jadwal dan `getJadwal()` untuk mendapatkan jadwal sidang.



### 3.1.4 Implementasi

Hasil dari perancangan sistem selanjutnya diimplementasikan dengan membuat program dengan bahasa pemrograman PHP dan menggunakan *framework* CodeIgniter versi 4. Basis data yang digunakan adalah MySql.

### 3.1.5 Pengujian

Pengujian sistem akan dilakukan dengan menggunakan metode *Usability Testing*. *Usability Testing* merupakan salah satu teknik dalam evaluasi *usability* untuk mengumpulkan data (Santoso, 2018) . Rumus untuk menghitung SUS dapat dilihat pada Rumus (3.1).

$$\bar{x} = \frac{\sum x}{n} \quad (3.1),$$

dengan,

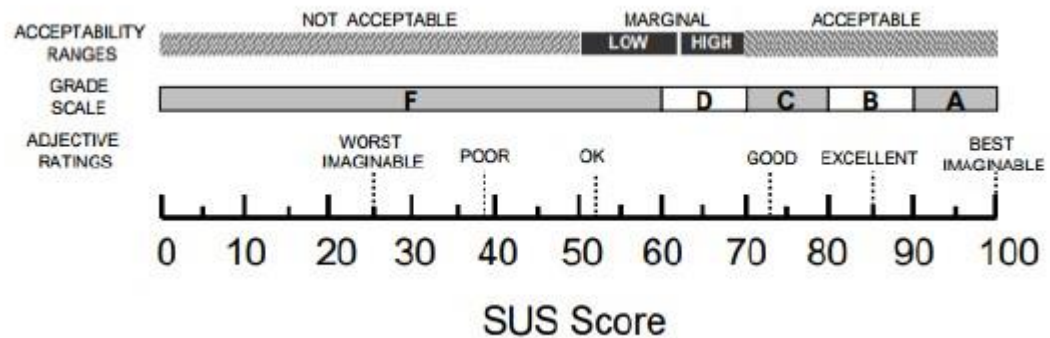
$\bar{x}$  = Skor rata-rata,

$\sum x$  = Jumlah Skor SUS,

$n$  = Jumlah Responden

Kuesioner terdiri dari 10 pertanyaan di mana setiap pertanyaan dijawab dengan menggunakan skala Likert sebanyak 5 buah skala. Penggunaan skala Likert pada penelitian ini menggunakan lima buah skala dengan ketentuan sebagai berikut, yaitu 1 = Sangat Tidak Setuju (STS), 2 = Tidak Setuju, 3 = Ragu-Ragu (RG), 4 = Setuju (S) dan 5 = Sangat Setuju (SS). Target sampel yang akan diuji merupakan mahasiswa Sistem Informasi UAJ angkatan 2017 dan 2018. Pengujian ini nantinya akan menggunakan *Google form* untuk membuat kuestioner dan membagikannya pada

responden. Hasil dari pengujian ini kemudian akan diinterpretasikan berdasarkan perbandingan skor pada Gambar 3.10.



Gambar 3. 10 SUS *score* (Santoso,2018)

Pada pengujian sistem, pengguna akan diberikan beberapa tugas pada saat menggunakan sistem sebelum mengisi kuesioner. Tugas tersebut antara lain:

1. Pengguna masuk ke dalam *website*.
2. Mahasiswa melakukan registrasi untuk menggunakan fitur yang terdapat pada sistem.
3. Setelah registrasi, mahasiswa melakukan *login* ke dalam aplikasi
4. Setelah *login*, mahasiswa masuk ke dalam halaman *dashboard*.
5. Pada halaman *dashboard* mahasiswa dapat melihat informasi tentang dosen pembimbing, judul topik yang diambil dan status pengerjaan tugas akhir serta proposal.
6. Pada bagian Bimbingan, terdapat menu daftar dosen yang mana mahasiswa dapat melihat daftar dosen dan melihat informasi tentang dosen. Pada bagian ini juga, mahasiswa dapat melihat dan mengambil judul topik yang di *post* oleh dosen. Mahasiswa juga dapat melakukan bimbingan setelah memilih dan

7. mendapatkan persetujuan dari dosen pembimbing. Pada menu bimbingan mahasiswa dapat melakukan bimbingan dengan cara mengunggah *file* laporan.
8. Pada bagian pendaftaran, mahasiswa dapat melihat menu pendaftaran seminar dan sidang TA. Pada tiap-tiap menu tersebut, mahasiswa dapat mengunggah dokumen dan laporan yang diminta sebagai persyaratan dalam melakukan sidang TA atau seminar.
9. Pada bagian sidang, mahasiswa dapat melihat nilai dari laporan yang telah disidangkan atau diseminarkan.
10. Pada bagian jadwal, terdapat menu jadwal sidang dan seminar. Mahasiswa dapat melihat jadwal sidang dan seminar mahasiswa itu sendiri dan mahasiswa lainnya.
11. Pada bagian jadwal, koordinator TA dapat menyusun jadwal sidang dan seminar mahasiswa.
12. Pada bagian sidang dan seminar, koordinator TA dapat menginput nilai akhir mahasiswa.
13. Pada bagian pendaftaran, koordinator TA dapat melihat isi dari form pendaftaran yang dikirim oleh mahasiswa dan mengirimkan komentar atau catatan.
14. Pada bagian bimbingan, dosen pembimbing dapat melihat daftar mahasiswa, mengunggah judul topik dan melakukan bimbingan dengan mahasiswa.

Kuesioner yang akan diberikan kepada mahasiswa berisi 10 pertanyaan yang sudah sesuai standar dalam melakukan *System Usability Scale* (SUS). Pada Tabel 2 merupakan daftar pertanyaan kuesioner yang akan diajukan kepada responden.

Tabel 5. Daftar pertanyaan kuesioner *usability testing*

No	Pertanyaan	STS	TS	RG	S	SS
1.	Saya berpikir akan menggunakan sistem ini lagi.					
2.	Saya merasa sistem ini rumit untuk digunakan.					
3.	Saya merasa sistem ini mudah digunaka.					
4.	Saya membutuhkan bantuan dari orang lain atau teknisi dalam menggunakan sistem ini.					
5.	Saya merasa fitur-fitur sistem ini berjalan dengan semestinya.					
6.	Saya merasa ada banyak hal yang tidak konsisten (tidak serasi pada sistem ini).					
7.	Saya merasa orang lain akan memahami cara menggunakan sistem ini dengan cepat					
8.	Saya merasa sistem ini membingungkan.					
9.	Saya merasa tidak ada hambatan dalam menggunakan sistem ini.					
10.	Saya perlu membiasakan diri terlebih dahulu sebelum menggunakan sistem ini.					

### 3.1.6 Pelaporan

Tahap pelaporan dilakukan dengan penyusunan buku tugas akhir sesuai dengan panduan penulisan buku tugas akhir. Pada bagian ini, terdapat seluruh penjelasan secara teknis aplikasi yang dibuat dan juga hasil analisa *usability testing* yang dilakukan.

### 3.3 Rencana dan Jadwal Penelitian

Tabel 6. Rencana dan jadwal penelitian

No	Kegiatan	Agustus 2020				September 2020					Oktober 2020				November 2020					Desember 2020				Januari 2020			
		1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4	5	1	2	3	4	1	2	3	4
1.	Pengumpulan Data																										
	• Studi Literatur	√	√	√	√	√	√																				
	• Studi Lapangan	√	√	√	√	√	√																				
2.	Analisis Kebutuhan Sistem				√	√	√	√	√	√																	
3.	Perancangan Sistem							√	√	√	√	√	√	√	√	√	√	√	√								
4.	Implementasi Sistem												√	√	√	√	√	√	√	√	√						
5.	Pengujian Sistem																	√	√	√	√	√					
6.	Pelaporan																										
	• Publikasi															√	√	√	√	√	√	√	√	√	√	√	
	• Laporan TA															√	√	√	√	√	√	√	√	√	√	√	
7.	Ujian TA																								√	√	

## DAFTAR PUSTAKA

- Šimec, A., Lozić, D., & Golubić, L. T. (2017). Benchmarking PHP modules. *Informatologia*, 1.
- Adiartika, M. H., & Supriana, I. W. (2020). Pengembangan Sistem Informasi Manajemen Tugas Akhir Program Studi Teknik Informatika Universitas Udayana. *Jurnal Elektronik Ilmu Komputer Udayana*.
- Alfina, O., & Harahap, F. (2019). PEMODELAN UML SISTEMPENDUKUNG KEPUTUSAN DALAM PENENTUAN KELAS SISWA SISWA TUNAGRAHITA. *Jurnal Manajemen Informatika & Komputerisasi Akuntansi*, 147.
- Codeigniter Web Framework. (2021). Retrieved from CodeIgniter: <https://codeigniter.com/>
- DuRocher, D. (2021). *HTML & CSS QuickStart Guide: The Simplified Beginners Guide to Developing a Strong Coding Foundation, Building Responsive Websites, and Mastering the Fundamentals of Modern Web Design*. ClydeBank Media LLC.
- Erinton, R., Negara, R. M., & Sanjoyo, D. D. (2017). ANALISIS PERFORMASI FRAMEWORK CODEIGNITERDAN LARAVELMENGUNAKAN WEB SERVER APACHE. *e-Proceeding of Engineering* , 3567.
- Ionescu, Ș., & Nica, I. (2020). Business process modeling Using Unified Modeling Language to streamline. *Theoretical and Applied Economics*, 94-95.
- Iqbal, S. Z., & Idress, M. (2017). Z-SDLC Model: A New Model For Software . *International Journal of Engineering and Advanced Research Technology (IJEART)*, 1.
- Julianto, T. D., Alam, S. N., Robo, S., & Widiyantoro, M. R. (2021). Sistem Informasi Manajemen Tugas Akhir di Program Studi . *Jurnal TEKNO KOMPAK*.
- Kelen, Y. R., & Belalawe, B. J. (2018). IMPLEMENTASI MODEL-VIEW-CONTROLLER (MVC) PADA UJIAN ONLINEMELALUI PENERAPAN FRAMEWORKCODEIGNITER. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 11.
- Lamsal, K. (2020). Designing and Developing a dynamic website using PHP. 14-15.
- Mehta, C., Bhavsar, A. K., Oza, H., & Shah, S. (2018). *MySQL 8 Administrator's Guide: Effective guide to administering high-performance MySQL 8 solutions*. Packt Publishing Ltd.
- Negara, E. S., Romindo, R., Tanjung, R., Heriyani, N., Simarmata, J., Jamaludin, J., . . . Purba, B. (2021). *Sistem Informasi Manajemen Bisnis*. Yayasan Kita Menulis.
- Padmanaba, A., N, E. K., & Andayati, D. (2020). KOMPARASI PENGGUNAAN FRAMEWORK CODEIGNITER VS PHP NATIVE PADA SISTEM INFORMASI MANAJEMEN SURAT SEKRETARIAT DPRD PEMALANG. *Jurnal SCRIPT*, 1.

- Permana, H. J., Ariyadi, Y., & Piriaselina, I. (2019). IMPLEMENTASISISTEM INFORMASI UNIT LAUNDRY BERBASIS WEB DI RUMAHSAKIT KANKER DHARMAIS. *Jurnal Dinamika Informatika*, 1.
- Prihandoyo, M. T. (2018). Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis. *Jurnal Informatika: Jurnal Pengembangan IT (JPIT)*.
- Putra, B. W., Saputra, A., Sanjaya, R., & Kurniawan, D. (2019). Implementasi Framework CodeIgniter dan Restful API pada Sistem Informasi Manajemen Tugas Akhir. *Computer Science and ICT*.
- Ramadhan, R. F., & Mukhaiyar, R. (2020). Penggunaan DatabaseMysql dengan InterfacePhpMyAdmin sebagai Pengontrolan SmarthomeBerbasis Raspberry Pi. *JTEIN: Jurnal Teknik Elektro Indonesia*, 130.
- Riana, D., Sanjaya, R., & Kalsoem, O. (2018). Sistem Informasi Manajemen Laboratorium Patologi Anatomi Menggunakan Model MVC Berbasis Laravel. *Jurnal Atma Luhur*, 237.
- Risawandi. (2019). *Mudah Menguasai PHP & MySQL dalam 24 Jam*. Unimal Press.
- Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. *Jurnal Media Infotama*, 48.
- Santoso, H. B. (2018). Evaluasi Aplikasi Media Pembelajaran Statistika Dasar Menggunakan Metode Usability Testing. *Jurnal Teknologi Informasi dan Komunikasi*, 52-53.
- Simangunsong, A. (2018). Sistem Informasi Pengarsipan DokumenBerbasis Web. *Jurnal Mantik Penusa*, 13.
- Sunara, R. E., & Candra, F. (2020). Sistem Informasi Penggajian Anggota Koperasi Sumber Rezeki Kotalama Menggunakan Visual Basic .NET. *Computation Technology And its Application*, 1.
- Supardi, Y., & Hermawan, A. (2018). *Semua Bisa Menjadi Programmer CodeIgniter Basic*. Elex Media Komputindo.
- Tulodo, B. A., & Solichin, A. (2019). ANALISIS PENGARUH KUALITAS SISTEM, KUALITAS INFORMASI DAN PERCEIVED USEFULNESS TERHADAP KEPUASAN PENGGUNA APLIKASI CARE DALAM UPAYA PENINGKATAN KINERJA KARYAWAN(Studi Kasus PT. Malacca Trust Wuwungan Insurance, Tbk.). *Riset Manajemen Sains Indonesia (JRMSI)*, 27.