**SISTEM INFORMASI TUGAS AKHIR (SISTA) BERBASIS WEB MENGGUNAKAN FRAMEWORK LARAVEL**

**Tugas Akhir**

diajukan untuk memenuhi sebagian persyaratan mencapai

gelar Ahli Madya pada jenjang Diploma III

Jurusan Teknik Informatika



Oleh

**AMELIA SYAMSA MAYANGSARI**

**NIM. 1603034**

**JURUSAN TEKNIK INFORMATIKA**

**POLITEKNIK NEGERI INDRAMAYU**

**2019**

# **HALAMAN PENGESAHAN**

Tugas Akhir ini diajukan oleh :

Nama : Amelia Syamsa Mayangsari

NIM : 1603034

Program Studi : Teknik Informatika

Judul : Sistem Informasi Tugas Akhir (SISTA) Berbasis Web

Menggunakan Framework Laravel

Pembimbing : 1. Eka Ismantohadi, S.Kom., M.Eng …………………

2. Muhamad Mustamiin, S.Pd., M.Kom …………………

Telah berhasil dipertahankan dihadapan dewan penguji pada tanggal 20 Agustus 2019 dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Ahli Madya Program Studi Teknik Informatika, Politeknik Negeri Indramayu.

DEWAN PENGUJI

|  |  |  |  |
| --- | --- | --- | --- |
| Nama | Jabatan | Tandatangan | Tanggal |
| 1. Darsih, S.Kom., M.Kom | Ketua Penguji | …………… | …………… |
| 1. Muhamad Mustamiin, S.Pd., M.Kom | Sekretaris Penguji | …………… | …………… |
| 1. Alifia Puspaningrum, S.Pd., M.Kom | Anggota | …………… | …………… |
|  | Indramayu, September 2019  Ketua Jurusan Teknik Informatika  **Iryanto, S.Si., M.Si.,**  **NIP 199008012019031014** | | | |

# **PERNYATAAN KEASLIAN**

Saya menyatakan dengan sesungguhnya bahwa Tugas Akhir ini adalah asli hasil karya saya sendiri, dan sepanjang pengetahuan saya tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Indramayu, 20 Agustus 2019

Yang menyatakan

AMELIA SYAMSA MAYANGSARI

NIM. 1603034

# **ABSTRAK**

Proses pelaksanaan tugas akhir yang didokumentasikan merupakan sebuah keharusan ditengah kemajuan teknologi seperti sekarang. Pengolahan administrasi tugas akhir pada D3 Teknik Informatika Politeknik Negeri Indramayu umumnya masih terkomputerisasi secara sederhana belum ada sistem khusus yang digunakan untuk mengelola tugas akhir. Hal ini tentunya akan membuat proses administrasi tugas akhir menjadi kurang efisien. berdasarkan hal tersebut maka penulis membuat SistemInformasi Tugas Akhir (SISTA) Berbasis Web Menggunakan Framework Laraveluntuk mempermudah proses pelaksanaan tugas akhir khususnya proses administrasi tugas akhir. Didalam Sistem Informasi Tugas Akhir (SISTA) terdapat beberapa fitur seperti pengajuan proposal, pembagian pembimbing, dan penjadwalan sidang. Dengan sistem informasi berbasis web ini diharapkan data tugas akhir pada jurusan tersebut bisa terstruktur dengan baik sehingga dapat mempermudah semua pihak yang mengelola tugas akhir dan memberikan pelayanan yang baik untuk mahasiswa. Penelitian ini menghasilkan sebuah aplikasi sistem informasi tugas akhir berbasis web yang digunakan untuk mengelola proses administrasi tugas akhir pada D3 Jurusan Teknik Informatika Politeknik Negeri Indramayu.

Kata kunci : Tugas Akhir, Sistem Informasi, Web, Laravel, Aplikasi.

# ***ABSTRACT***

*The process of implementing the documented final project is a necessity in the midst of technological progress as it is now. Processing the final assignment in D3 Informatics Engineering, Polytechnic of Indramayu State is generally still computerized in a simple way, there is no specific system used to manage the final project. This will certainly make the final project administration process less efficient. based on this, the authors make a Final Project Information System (SISTA) Web-Based Using the Laravel Framework to facilitate the process of implementing the final project, especially the final project administration process. In the Final Assignment Information System (SISTA) there are several features such as submission of proposals, the distribution of mentors, and scheduling of hearings. With this web-based information system, it is expected that the final project data in the department can be well structured so that it can facilitate all parties who manage the final project and provide good services for students. This research resulted in a web-based final project information system application that is used to manage the final project administration process at the D3 Department of Informatics, Polytechnic State of Indramayu.*

*Keywords : Division, Naive Bayes, Nature Lovers.*

# **MOTTO**

*“A good relationship isn’t found, it’s built.”* - @httpluahan\_

# **KATA PENGANTAR**

Puji syukur kita panjatkan kehadirat Allah SWT yang telah memberikan limpahan rahmat dan ridha-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini. Sholawat dan salam senantiasa tercurah atas Nabi Muhammda SAW.

Tugas akhir yang merupakan salah satu persaratan untuk memperoleh gelar Ahli Madya pada jurusan Teknik Informatika Politeknik Negeri Indramayu. Tugas Akhir yang penulis buat berjudul “Sistem Informasi Tugas Akhir (SISTA) berbasis Web Menggunakan Framework Laravel”.

Tugas Akhir ini kiranya tidak akan selesai tanpa bantuan dan dorongan dari berbagai pihak, oleh karena itu penulis ucapkan banyak terimakasih kepada :

1. Bapak Casiman Sukardi, ST., MT., selaku Direktur Politeknik Negeri Indramayu.
2. Bapak Iryanto, S.Si., M.Si., selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Indramayu.
3. Bapak Eka Ismantohadi S.Kom., M.Eng., selaku pembimbing utama.
4. Bapak Muhamad Mustamiin S.Pd.,M.Kom., selaku pembimbing pendamping sekaligus koordinator TA.
5. Kedua orang tua dan keluarga yang telah memberikan do’a serta dukungan kepada penulis.
6. Seluruh teman-teman kelas D3TI.B tahun angkatan 2016 yang telah banyak membantu dalam penyelesaian tugas akhir ini.

Akhir kata penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan pihak yang membutuhkan.

Indramayu, 20 Juni 2019

Penulis

# **DAFTAR ISI**

[HALAMAN PENGESAHAN ii](#_Toc19536766)

[PERNYATAAN KEASLIAN iii](#_Toc19536767)

[ABSTRAK iv](#_Toc19536768)

[*ABSTRACK* v](#_Toc19536769)

[MOTTO vi](#_Toc19536770)

[KATA PENGANTAR vii](#_Toc19536771)

[DAFTAR ISI viii](#_Toc19536772)

[DAFTAR TABEL xi](#_Toc19536773)

[DAFTAR GAMBAR xiii](#_Toc19536774)

[DAFTAR LAMPIRAN xvi](#_Toc19536775)

[BAB I](#_Toc19536776) [PENDAHULUAN 1](#_Toc19536777)

[1.1 Latar Belakang Masalah 1](#_Toc19536778)

[1.2 Rumusan Masalah 2](#_Toc19536779)

[1.3 Batasan Masalah 2](#_Toc19536780)

[1.4 Tujuan Penelitian 2](#_Toc19536781)

[1.5 Manfaat Penelitian 3](#_Toc19536782)

[1.6 Sistematika Penulisan 3](#_Toc19536783)

[BAB II](#_Toc19536784) [LANDASAN TEORI 5](#_Toc19536785)

[2.1 Sistem Informasi 5](#_Toc19536786)

[2.2 HTML 5](#_Toc19536787)

[2.3 CSS 6](#_Toc19536788)

[2.4 Bootstrap 7](#_Toc19536789)

[2.5 Javascript 7](#_Toc19536790)

[2.6 Bahasa Pemrograman PHP (PHP: *Hypertext Prepocessor*) 7](#_Toc19536791)

[2.7 Basis Data 9](#_Toc19536792)

[2.8 MySQL 9](#_Toc19536793)

[2.9 XAMPP 9](#_Toc19536794)

[2.10 PhpMyAdmin 10](#_Toc19536795)

[2.11 Composer 10](#_Toc19536796)

[2.12 Laravel 10](#_Toc19536797)

[2.13 Atom Text Editor 11](#_Toc19536798)

[2.14 Google Chrome 12](#_Toc19536799)

[2.15 Website 12](#_Toc19536800)

[2.15 UML (*Unified Modelling Language*) 13](#_Toc19536801)

[2.16 Use Case Diagram 13](#_Toc19536802)

[2.17 *Activity Diagram* 15](#_Toc19536803)

[2.18 *Class Diagram* 15](#_Toc19536804)

[2.19 *Flowchart* 16](#_Toc19536805)

[2.20 *Entity* *Relationship Diagram* 18](#_Toc19536806)

[2.21 Metode Waterfall 19](#_Toc19536807)

[2.21 Kuesioner 20](#_Toc19536808)

[2.22 Skala Likert 20](#_Toc19536809)

[BAB III](#_Toc19536810) [METODOLOGI PENELITIAN 21](#_Toc19536811)

[3.1 Metode Penelitian 21](#_Toc19536812)

[3.2 Perancangan Sistem 23](#_Toc19536813)

[3.2.1 *Flowchart* 23](#_Toc19536814)

[3.2.2 *Use Case Diagram* 31](#_Toc19536815)

[3.2.3 *Activity Diagram* 33](#_Toc19536816)

[3.2.4 *Class Diagram* 41](#_Toc19536817)

[3.2.5 *Entity Relationship Diagram* 43](#_Toc19536818)

[3.2.6 Normalisasi *Database* 44](#_Toc19536819)

[3.2.7 Perancangan *Database* 47](#_Toc19536820)

[3.2.8 Perancangan Antar Muka 54](#_Toc19536834)

[BAB IV](#_Toc19536843) [HASIL DAN PEMBAHASAN 59](#_Toc19536844)

[4.1 Hasil 59](#_Toc19536845)

[4.2 Pembahasan 59](#_Toc19536846)

[4.2.1 Kebutuhan Sistem 59](#_Toc19536847)

[4.2.2 Implementasi 60](#_Toc19536848)

[4.2.1.1 Implementasi Database 60](#_Toc19536849)

[4.2.1.2 Implementasi Interface 65](#_Toc19536850)

[4.2.2 Pengujian 83](#_Toc19536851)

[4.2.2.2 Rekapitulasi Hasil Perhitungan Kuesioner 89](#_Toc19536852)

[4.2.2.3 Kesimpulan Pengujian 90](#_Toc19536853)

[4.2.3 Perbaikan Sistem 90](#_Toc19536854)

[BAB V](#_Toc19536855) [PENUTUP 91](#_Toc19536856)

[1.1 Kesimpulan 91](#_Toc19536857)

[1.2 Saran 91](#_Toc19536858)

[DAFTAR PUSTAKA 93](#_Toc19536859)

[LAMPIRAN 96](#_Toc19536860)

# **DAFTAR TABEL**

[Tabel 2. 1 Simbol – simbol use case diagram 14](#_Toc19227396)

[Tabel 2. 2 Simbol – simbol activity diagram 15](#_Toc19227397)

[Tabel 2. 3 Simbol - simbol class diagram 16](#_Toc19227398)

[Tabel 2. 4 Simbol – simbol flowchart 17](#_Toc19227399)

[Tabel 2. 6 Simbol – simbol ERD 19](#_Toc19227400)

[Tabel 2. 7 Kriteria Penilaian Skala Likert 20](#_Toc19227401)

[Tabel 3. 1 Tabel Proposal Unnormal 44](#_Toc19311406)

[Tabel 3. 2 Tabel Proposal Normalisasi Kesatu (1NF) 45](#_Toc19311407)

[Tabel 3. 3 Tabel proposal Normalisasi Kedua (2NF) 46](#_Toc19311408)

[Tabel 3. 4 Tabel data\_mahasiswa 46](#_Toc19311409)

[Tabel 3. 5 Tabel data mahasiswa 46](#_Toc19311410)

[Tabel 3. 6 Tabel proposal Normalisasi Ketiga (3NF) 47](#_Toc19311411)

[Tabel 3. 7 Tabel judul\_ta Normalisasi Ketiga (3NF) 47](#_Toc19311412)

[Tabel 3. 8 Tabel admin\_account 47](#_Toc19311413)

[Tabel 3. 9 Tabel data\_dosen 48](#_Toc19311414)

[Tabel 3. 10 Tabel data\_mahasiswa 48](#_Toc19311415)

[Tabel 3. 11 Tabel dosen\_account 49](#_Toc19311416)

[Tabel 3. 12 Tabel judul\_ta 49](#_Toc19311417)

[Tabel 3. 13 Tabel kelas 50](#_Toc19311418)

[Tabel 3. 14 Tabel role 50](#_Toc19311419)

[Tabel 3. 15 Tabel hak\_akses\_dosen 50](#_Toc19311420)

[Tabel 3. 16 Tabel mahasiswa\_account 51](#_Toc19311421)

[Tabel 3. 17 Tabel proposal 51](#_Toc19311422)

[Tabel 3. 18 Tabel revisi\_ta 52](#_Toc19311423)

[Tabel 3. 19 Tabel sidang\_ta 52](#_Toc19311424)

[Tabel 3. 20 Tabel jadwal\_sidang 53](#_Toc19311425)

[Tabel 4. 1 Kebutuhan Hardware 59](#_Toc19227558)

[Tabel 4. 2 Kebutuhan Software 60](#_Toc19227559)

[Tabel 4. 3 admin\_account 61](#_Toc19227560)

[Tabel 4. 4 data\_dosen 61](#_Toc19227561)

[Tabel 4. 5 data\_mahasiswa 61](#_Toc19227562)

[Tabel 4. 6 dosen\_account 62](#_Toc19227563)

[Tabel 4. 7 judul\_ta 62](#_Toc19227564)

[Tabel 4. 8 kelas 62](#_Toc19227565)

[Tabel 4. 9 role 63](#_Toc19227566)

[Tabel 4. 10 hak\_akses\_dosen 63](#_Toc19227567)

[Tabel 4. 11 mahasiswa\_account 63](#_Toc19227568)

[Tabel 4. 12 proposal 64](#_Toc19227569)

[Tabel 4. 13 revisi\_ta 64](#_Toc19227570)

[Tabel 4. 14 sidang\_ta 65](#_Toc19227571)

[Tabel 4. 15 jadwal\_sidang 65](#_Toc19227572)

[Tabel 4. 16 Pertanyaan Kuisioner untuk Mahasiswa 84](#_Toc19227573)

[Tabel 4. 17 Pertanyaan Kuisioner untuk Administrator 84](#_Toc19227574)

[Tabel 4. 18 Pertanyaan Kuisioner untuk Dosen 85](#_Toc19227575)

[Tabel 4. 19 Hasil Jawaban Kuisioner 86](#_Toc19227576)

[Tabel 4. 20 Rekap Hasil Jawaban Kuisioner 86](#_Toc19227577)

# **DAFTAR GAMBAR**

[Gambar 2. 1 Siklus Informasi (Sumber: Dharmawijaya. 2018) 5](#_Toc18409256)

[Gambar 2. 2 Scirpt HTML 6](#_Toc18409257)

[Gambar 2. 3 Script PHP (Sumber: Cox. 2019) 9](#_Toc18409258)

[Gambar 2. 4 Tampilan XAMPP Control Panel 9](#_Toc18409259)

[Gambar 2. 5 Logo Laravel (Sumber: Kibria. 2018) 10](#_Toc18409260)

[Gambar 2. 6 Logo Google Chrome (Sumber: Wasim. 2018) 12](#_Toc18409261)

[Gambar 2. 7 Contoh Use Case Diagram (Sumber: Pratama. 2019) 14](#_Toc18409262)

[Gambar 2. 8 Bentuk umum class diagram (Sumber: Novitasari. 2018) 16](#_Toc18409263)

[Gambar 3. 1 Alur Metode Penelitian 21](#_Toc19311535)

[Gambar 3. 2 Flowchart Administrator 24](#_Toc19311536)

[Gambar 3. 3 Flowchart Mahasiswa 25](#_Toc19311537)

[Gambar 3. 4 Flowchart Koordinator 26](#_Toc19311538)

[Gambar 3. 5 Flowchart Pembimbing 27](#_Toc19311539)

[Gambar 3. 6 Flowchart Ketua Jurusan 28](#_Toc19311540)

[Gambar 3. 7 Flowchart Wali Dosen 29](#_Toc19311541)

[Gambar 3. 8 Flowchart System 30](#_Toc19311542)

[Gambar 3. 9 Use Case Diagram 32](#_Toc19311543)

[Gambar 3. 10 Activity Diagram Login User 33](#_Toc19311544)

[Gambar 3. 11 Activity Diagram Tambah data mahasiswa / dosen 34](#_Toc19311545)

[Gambar 3. 12 Activity Diagram Pengajuan Proposal Tugas Akhir 35](#_Toc19311546)

[Gambar 3. 13 Activity Diagram Pemilihan Pembimbing 36](#_Toc19311547)

[Gambar 3. 14 Activity Diagram Setujui / Tolak Pengajuan Proposal Tugas Akhir 37](#_Toc19311548)

[Gambar 3. 15 Activity Diagram Pengajuan Sidang Tugas Akhir 38](#_Toc19311549)

[Gambar 3. 16 Activity Diagram Setujui Pengajuan Sidang Tugas Akhir 39](#_Toc19311550)

[Gambar 3. 17 Activity Diagram Konfirmasi Pengajuan Sidang Tugas Akhir 40](#_Toc19311551)

[Gambar 3. 18 Activity Diagram Penjadwalan Sidang Tugas Akhir 41](#_Toc19311552)

[Gambar 3. 19 Class Diagram 42](#_Toc19311553)

[Gambar 3. 20 Entity Relationship Diagram (ERD) 43](#_Toc19311554)

[Gambar 3. 21 Rancangan Halaman Awal 54](#_Toc19311555)

[Gambar 3. 22 Rancangan Halaman Login 55](#_Toc19311556)

[Gambar 3. 23 Rancangan Halaman Home 55](#_Toc19311557)

[Gambar 3. 24 Rancangan Halaman Tampil Data TA Akhir di User Mahasiswa 56](#_Toc19311558)

[Gambar 3. 25 Rancangan Halaman Tampil Data 56](#_Toc19311559)

[Gambar 3. 26 Rancangan Halaman Tampil Data Pengajuan Tugas Akhir Mahasiswa 57](#_Toc19311560)

[Gambar 3. 27 Rancangan Halaman Tambah Data 57](#_Toc19311561)

[Gambar 3. 28 Rancangan Halaman Profil 58](#_Toc19311562)

[Gambar 4. 1 Tampilan Halaman Awal 66](#_Toc19227435)

[Gambar 4. 2 Tampilan Halaman Login 66](#_Toc19227436)

[Gambar 4. 3 Tampilan Halaman Home Admin 67](#_Toc19227437)

[Gambar 4. 4 Tampilan Halaman Home Dosen 67](#_Toc19227438)

[Gambar 4. 5 Tampilan Halaman Home Mahasiswa 68](#_Toc19227439)

[Gambar 4. 6 Tampilan Halaman Tambah Data Dosen 69](#_Toc19227440)

[Gambar 4. 7 Tampilan List Data Dosen 69](#_Toc19227441)

[Gambar 4.8 Tampilan List Hak Akses Dosen 70](#_Toc19227442)

[Gambar 4.9 Tampilan Halaman Tambah Kelas 70](#_Toc19227443)

[Gambar 4. 10 Tampilan List Kelas 71](#_Toc19227444)

[Gambar 4. 11 Tampilan Tambah Hak Akses Dosen 71](#_Toc19227445)

[Gambar 4. 12 Tampilan Tambah Data Mahasiswa 72](#_Toc19227446)

[Gambar 4. 13 Tampilan List Data Mahasiswa 72](#_Toc19227447)

[Gambar 4. 14 Tampilan Import Data 73](#_Toc19227448)

[Gambar 4. 15 Tampilan Ubah Data 73](#_Toc19227449)

[Gambar 4. 16 Tampilan Konfirmasi Sidang 74](#_Toc19227450)

[Gambar 4. 17 Tampilan List Pengajuan TA 74](#_Toc19227451)

[Gambar 4. 18 Tampilan Pembagian Pembimbing 75](#_Toc19227452)

[Gambar 4. 19 Tampilan Status Pengajuan TA 75](#_Toc19227453)

[Gambar 4. 20 Tampilan List Pengajuan Sidang TA 76](#_Toc19227454)

[Gambar 4. 21 Tampilan Persetujuan Sidang TA 76](#_Toc19227455)

[Gambar 4. 22 Tampilan Tambah Jadwal Sidang TA 77](#_Toc19227456)

[Gambar 4. 23 Tampilan List Jadwal Sidang TA 77](#_Toc19227457)

[Gambar 4. 24 Tampilan List Revisi Laporan TA 78](#_Toc19227458)

[Gambar 4. 25 Tampilan Tambah Pengajuan TA 78](#_Toc19227459)

[Gambar 4. 26 Tampilan Tambah Pengajuan TA 79](#_Toc19227460)

[Gambar 4. 27 Tampilan Ubah Pengajuan TA 79](#_Toc19227461)

[Gambar 4. 28 Tampilan List Judul TA 80](#_Toc19227462)

[Gambar 4. 29 Tampilan Tambah Pengajuan Sidang TA 80](#_Toc19227463)

[Gambar 4. 30 Tampilan Tambah Pengajuan Sidang dan Jadwal Sidang TA 81](#_Toc19227464)

[Gambar 4. 31 Tampilan Ubah Pengajuan Sidang TA 81](#_Toc19227465)

[Gambar 4. 32 Tampilan Tambah Revisi Laporan TA 82](#_Toc19227466)

[Gambar 4. 33 Tampilan Ubah Revisi TA 82](#_Toc19227467)

[Gambar 4. 34 Tampilan Revisi TA 83](#_Toc19227468)

[Gambar 4. 35 Tampilan Halaman Akun 83](#_Toc19227469)

# **DAFTAR LAMPIRAN**

Lampiran 1 Hasil Observasi

Lampiran 2 Kode Program

Lampiran 3 Kuisioner

Lampiran 4 Biodata Penulis

# **BAB I**

# **PENDAHULUAN**

## **Latar Belakang Masalah**

Perkembangan teknologi sangat pesat dan peranannya sangat penting untuk mendukung aktivitas manusia agar dapat mengoptimalkan waktu dengan lebih baik. Salah satu contoh teknologi yang membantu aktivitas manusia adalah sistem informasi. Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengelolaan data, mendukung operasi, bersifat manajerial dan strategi kegiatan dari suatu organisasi atau instansi dan menyediakan laporan-laporan bagi pihak tertentu. Instansi yang bergerak di bidang pendidikan juga membutuhkan dukungan sistem informasi dalam peningkatan mutu pelayanan terhadap proses pendidikan. Seperti sistem informasi akademik yang terdiri dari banyak proses antara lain perencanaan perkuliahan hingga proses akhir seperti bimbingan tugas akhir mahasiswa (Noviana, 2018).

Tugas akhir merupakan karya akhir mahasiswa yang wajib dilaksanakan pada jenjang diploma III sebagai syarat kelulusan serta untuk mendapatkan gelar Ahli Madya. Tugas Akhir adalah sarana untuk mengetahui kemampuan seorang mahasiswa apakah sudah menguasai ilmu yang diberikan dan layak untuk mengabdi di masyarakat sesuai dengan kompetensi yang diajarkan oleh kampus. Tugas Akhir atau sering disingkat TA merupakan langkah awal untuk dapat belajar dalam menghadapi dunia kerja yang akan dihadapi, dengan adanya Tugas Akhir mahasiswa dapat mempersiapkan diri untuk menyelesaikan proyek-proyek di masa kerja nanti (Aritias Anggaeni, 2013:38).

Proses pelaksanaan tugas akhir yang didokumentasikan merupakan sebuah keharusan ditengah kemajuan teknologi seperti sekarang. Pengolahan administrasi tugas akhir pada D3 Teknik Informatika Politeknik Negeri Indramayu umumnya masih terkomputerisasi secara sederhana belum ada sistem khusus yang digunakan untuk mengelola tugas akhir. Hal ini tentunya akan membuat proses administrasi tugas akhir menjadi kurang efisien.

Oleh karena itu dibutuhkan sebuah sistem atau software untuk membantu proses administrasi tugas akhir di D3 Teknik Informatika Politeknik Negeri Indramayu. Berdasarkan hal tersebut maka penulis membuat SistemInformasi Tugas Akhir (SISTA) Berbasis Web Menggunakan Framework Laraveluntuk mempermudah proses pelaksanaan tugas akhir khususnya proses administrasi tugas akhir. Dengan aplikasi ini diharapkan data tugas akhir pada jurusan tersebut bisa terstruktur dengan baik sehingga dapat mempermudah administrator, koordinator, dosen pembimbing, mahasiswa, wali dosen, dan ketua jurusan dalam mengkoordinasi proses pelaksanaan tugas akhir serta memberikan pelayanan yang baik untuk mahasiswa.

## **Rumusan Masalah**

Berdasarkan latar belakang yang telah dijelaskan, maka rumusan masalah terletak pada:

1. Bagaimana merancang aplikasi sistem informasi tugas akhir berbasis web ?
2. Bagaimana membuat sistem informasi tugas akhir berbasis web menggunakan berdasarkan rancangan yang sudah dibuat ?
3. Bagaimana sistem informasi tugas akhir berbasis web menggunakan framework laravel ini memberikan kemudahan pada pengguna agar proses pengelolaan tugas akhir dapat lebih mudah ?

## **Batasan Masalah**

Berdasarkan rumusan masalah diatas, maka masalah tersebut dapat dibatasi agar tepat pada sasaran, adapun batasan masalah dari penelitian adalah sebagai berikut :

1. Sistem ini berfokus pada pengajuan proposal, pembagian pembimbing, dan penjadwalan sidang.
2. Sistem ini dibangun untuk D3 Jurusan Teknik Informatika Politeknik Negeri Indramayu.
3. Sistem ini hanya dapat digunakan melalui web *browser*.

## **Tujuan Penelitian**

Tujuan dari penelitian ini diantaranya sebagai berikut :

1. Membuat aplikasi sistem informasi tugas akhir berbasis web menggunakan framework laravel yang dirancang dengan menggunakan UML (*Unified Modelling Language*), *flowchart*, dan perancangan antar muka (*mockup*).
2. Membuat sistem informasi tugas akhir berbasis web menggunakan bahasa pemrograman PHP, framework laravel, dan MySQL sebagai pengolah basis data.
3. Membuat sistem informasi tugas akhir berbasis web menggunakan framework laravel yang menyediakan fitur pengajuan proposal TA, pembagian pembimbing TA, penjadwalan sidang sehingga dapat mempermudah proses administrasi tugas akhir.

## **Manfaat Penelitian**

Hasil dari penelitian ini diharapkan dapat bermanfaat untuk beberapa pihak diantaranya :

1. Manfaat Bagi Penulis

Menerapkan wawasan dan ilmu pengetahuan yang telah diperoleh selama perkuliahan dan menambahkan pengetahuan baru yang diperoleh selama melaksanakan penelitian ini.

1. Manfaat Bagi Pengguna Aplikasi

Aplikasi ini sebagai salah satu sarana atau fasilitas untuk mendukung proses pelaksanaan tugas akhir, diharapkan dapat mempermudah pengguna untuk memberikan ataupun mendapatkan informasi mengenai tugas akhir.

## **Sistematika Penulisan**

Untuk memenuhi kaedah penulisan laporan dan pemahaman yang lebih sistematis, penulisan laporan tugas akhir ini tersusun dalam lima bab sesuai dengan panduan penulisan tugas akhir yang dikeluarkan oleh Politeknik Negeri Indramayu dengan penjelasan sebagai berikut :

**BAB I PENDAHULUAN**

Merupakan pendahuluan laporan tugas akhir yang mengemukakan Latar Belakang Masalah, Rumusan Masalah, Batasan Masalah, Manfaat Penelitan dan Sistematika Penulisan Laporan.

**BAB II LANDASAN TEORI**

Bagian ini berisikan tentang landasan teori yang berkaitan dengan proses penelitian tugas akhir seperti Sistem Informasi, Web, Framework Laravel, dan *Database*.

**BAB III METODOLOGI PENELITIAN**

Bagian ini menjelaskan tentang alur sistem dari pembuatan aplikasi. Terdapat flowchart, use case, dan class diagram yang menggambarkan alur dari pengguna aplikasi serta rancangan database.

**BAB IV HASIL DAN PEMBAHASAN**

Pada bab ini dibahas hasil-hasil dari tahapan analisis, tahapan implementasi dari penelitian yang telah dilakukan.

**BAB V PENUTUP**

Bagian terakhir yang berisikan tentang kesimpulan dari penelitian dan terdapat saran-saran berdasarkan penelitian tugas akhir ini.

# **BAB II**

# **LANDASAN TEORI**

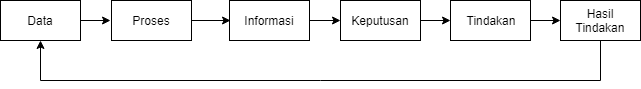
## **Sistem Informasi**

Sistem informasi merupakan sistem, yang berisi jaringan Sistem Pengolahan Data, yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem antara lain mengumpulkan data, mengelola data yang tersimpan, menyebarkan informasi (Aritias Anggaeni, 2013:38).

Sistem informasi merupakan sistem, yang berisi jaringan Sistem Pengolahan Data, yang dilengkapi dengan kanal-kanal komunikasi yang digunakan dalam sistem organisasi data. Elemen proses dari sistem antara lain mengumpulkan data, mengelola data yang tersimpan, menyebarkan informasi (Ghulam, 2015:30).

Keberadaan sistem informasi diperlukan organisasi untuk mendampingi proses-proses bisnis dari organisasi. Contohnya, proses penjualan supermarket didampingi oleh sistem informasi penjualan, yang mencatat pengumpulan data dan informasi tentang penjualan (Ghulam, 2015:31).

Adapun siklus informasi dapat dilihat pada Gambar 2.1.



Gambar 2. 1 Siklus Informasi (Sumber: Dharmawijaya, 2018)

## **HTML**

HTML adalah singkatan dari *Hypertext Markup Language*. Disebut *hypertext* karena di dalam HTML sebuah *text* biasa dapat berfungsi lain, kita dapat membuatnya menjadi *link* yang dapat berpindah dari satu halaman ke halaman lainnya hanya dengan meng-klik *text* tersebut. Kemampuan *text* inilah yang dinamakan *hypertext*, walaupun pada implementasinya nanti tidak hanya *text* yang dapat dijadikan *link*. Disebut *Markup Language* karena bahasa HTML menggunakan tanda (*mark*), untuk menandai bagian-bagian dari *text*. Misalnya, *text* yang berada di antara tanda tertentu akan menjadi tebal, dan jika berada di antara tanda lainnya akan tampak besar. Tanda ini di kenal sebagai HTML tag.

HTML bukanlah bahasa pemrograman, dan itu berarti HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. Sebagai gantinya, HTML

memungkinkan *user* untuk mengorganisir dan memformat dokumen, sama seperti Microsoft Word (Andre, 2017).

Dokumen HTML adalah *file* yang diakhiri dengan ekstensi .html atau .htm. Ekstensi *file* ini bisa dilihat dengan menggunakan web browser apapun (seperti Google Chrome, Safari, atau Mozila Firefox). Browser tersebut membaca *file* HTML dengan me-*render* kontennya sehingga *user* internet bisa melihat dan membacanya (Ariata, 2018).

Adapun script HTML dapat dilihat pada Gambar 2.2.



Gambar 2. 2 Scirpt HTML (Sumber: Muhardian, 2016)

## **CSS**

CSS adalah kependekan dari *Cascading Style Sheet*. CSS merupakan kumpulan kode – kode yang bertujuan untuk menghias dan mengatur gaya tampilan/*layout* halaman web supaya lebih elegan dan menarik. CSS adalah sebuah teknologi internet yang direkomendasikan oleh *World Wide Web* *Consortium* atau W3C pada tahun 1996. Awalnya, CSS dikembangkan di SGML pada tahun 1970, dan terus dikembangkan hingga saat ini. CSS telah mendukung banyak bahasa *markup* seperti HTML, XHTML, XML, SVG (*Scalable Vector Graphics*) dan Mozilla XUL (XML *User Interface Language*) (Rahman, 2013).

CSS mendeskripsikan presentasi atau tampilan sebuah halamam web. Sehingga, kode CSS digunakan untuk mempercantik dan memperbaiki tampilan antar muka sebuah website. Seperti, mewarnai tulisan, memberi *background* (latar) baik dengan warna ataupun gambar, menentukan *font*, memberi animasi, mengatur tampilan sesuai ukuran layar web browser yang digunakan, dan hal – hal lainnya yang berkaitan dengan desain sebuah website (Halwa, 2018).

## **Bootstrap**

Bootstrap merupakan framework untuk membangun desain web secara responsif. Artinya, tampilan web yang dibuat oleh bootstrap akan menyesuaikan ukuran layar dari browser yang kita gunakan baik di desktop, tablet ataupun mobile *device*. Fitur ini bisa diaktifkan ataupun dinon-aktifkan sesuai dengan keinginan kita sendiri. Sehingga, kita bisa membuat web untuk tampilan desktop saja dan apabila dirender oleh mobile browser maka tampilan dari web yang kita buat tidak bisa beradaptasi sesuai layar. Dengan bootstrap kita juga bisa membangun web dinamis ataupun statis (Fadul, 2018).

## **Javascript**

Javascript adalah sebuah bahasa pemrograman web yang bersifat *Client Side*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang permrosesannya dilakukan oleh client yang dimaksud untuk merujuk kepada web browser seperti Google Chrome, Mozilla dan lain sebagainya (Rendra, 2018).

Javascript adalah Bahasa pemrograman *Client Side* berbeda dengan bahasa pemrograman *Server Side* seperti PHP, C, Java dan bahasa *back end* lainnya yang seluruh kodenya akan berjalan pada sisi *server* dan tidak dapat muncul pada layar browser (*Inspect Element*) (Rendra, 2018).

Untuk menjalankan Javascript hanya membutuhkan aplikasi *text editor* sebagai contoh sublime, atom, dreamweaver dan web browser. Javascript memiliki fitur high level *programming language*, *client side* dan berorientasi objek (Rendra, 2018).

## **Bahasa Pemrograman PHP (PHP: *Hypertext Prepocessor*)**

PHP adalah singkatan dari "PHP: *Hypertext Prepocessor*", yaitu bahasa pemrograman yang digunakan secara luas untuk penanganan pembuatan dan pengembangan sebuah situs web dan bisa digunakan bersamaan dengan HTML. PHP diciptakan oleh Rasmus Lerdorf pertama kali tahun 1994. Pada awalnya PHP adalah singkatan dari "*Personal Home Page Tools*". Selanjutnya diganti menjadi FI ("*Forms Interpreter*"). Sejak versi 3.0, nama bahasa ini diubah menjadi "PHP: *Hypertext Prepocessor*" dengan singkatannya "PHP". PHP versi terbaru adalah versi ke-5 (Rasyid, 2014).

* Fungsi PHP

Pada umumnya, untuk membangun halaman web, PHP bukanlah menjadi bahasa pemrograman yang utama. Menggunakan HTML dan CSS sudah dapat menghasilkan halaman web statis, dimana kontennya bersifst tetap. Bahasa PHP merupakan bahasa pemrograman pelengkap yang digunakan untuk menghasilkan web yang dinamis dan interaktif. Berikut akan dijelaskan mengenai beberapa fungsi dari PHP (Rasyid, 2014) :

1. Mempersingkat Tatanan HTML dan CSS

Untuk membangun sebuah halaman web yang dinamis, PHP dapat berfungsi untuk mempersingkat penggunaan tatana HTML dan CSS. Sebagai contoh, Dalam suatu sistem karyawan memiliki jumlah baris 100. Jika menggunakan HTML dan CSS maka barisan tersebut menjadi sangat panjang. Sedangkan jika ditambah dengan menggunakan PHP, makan anda bisa mengatur beberapa baris yang dibutuhkan atau ditampilkan (Rasyid, 2014).

1. Input Data

Dengan menggunakan bahasa pemrograman PHP, Anda dapat menginput data dan menyimpan dalam sistem *Database*, seperti MySQL (Rasyid, 2014)..

1. Manajemen Cookie dan Session

Dalam PHP, Cookie dan Session digunakan untuk menyimpan informasi pengguna. Fungsi session\_start() untuk memulai *session* dan cookie dengan fungsi setcookie(). Sebagai contoh proses cookie, menyimpan *username* dan *password* pengguna di browser sehingga tidak harus mengisinya ulang ketika membuka situs yang sama. *Session* contohnya menyimpan informasi *login* yang hanya berlaku dalam satu sesi saja (Rasyid, 2014)..

1. Kompress Teks

Dalam PHP, Anda bisa mengkompres teks yang panjang menjadi lebih pendek dengan fungsi gzcompress() dan mengembalikan dengan fungsi gzuncompress() (Rasyid, 2014).

Adapun script PHP dapat dilihat pada Gambar 2.3.



Gambar 2. 3 Script PHP (Sumber: Cox, 2019)

## **Basis Data**

Pengertian basis data dan sistem basis data – Basis data adalah, basis data terdiri dari 2 (dua) kata, yaitu kata Basis dan Data. Basis bisa di artikan sebagai markas ataupun gudang, tempat berkumpul. Sedangkan data yaitu kumpulan fakta dunia nyata yang mewakili suatu objek, seperti manusia, barang, dan lain-lain yang direkam ke dalam bentuk angka, bentuk huruf, simbol, teks, bunyi, gambar atau juga  kombinasinya (Sora, 2014).

## **MySQL**

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL ( *database management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License(GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing (Anggaeni, 2013:39).

## **XAMPP**



Gambar 2. 4 Tampilan XAMPP Control Panel

XAMPP ialah perangkat lunak bebas yang mendukung banyak sistem operasi, merupakan campuran dari beberapa program. Yang mempunyai fungsi sebagai server yang berdiri sendiri (localhost), yang terdiri dari program MySQL *database*, Apache HTTP Server, dan penerjemah ditulis dalam bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empat sistem operasi), Apache, MySQL, PHP dan Perl. Program ini tersedia di bawah GNU *General Public License* dan bebas, adalah mudah untuk menggunakan web *server* yang dapat melayani tampilan halaman web yang dinamis. Jika ingin mendapatkan xampp dapat mendownload langsung dari situs resminya (Dudung, 2019). Adapun tampilan XAMPP control panel dapat dilihat pada Gambar 2.4.

## **PhpMyAdmin**

PhpMyAdministrator adalah sebuah aplikasi open source yang berfungsi untuk memudahkan manajemen MySQL. Dengan menggunakan phpmyAdministrator, anda dapat membuat database, membuat tabel, menambahkan, menghapus dan mengubah data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL secara manual (Daniel, 2016).

## **Composer**

Composer adalah *Dependecy Manager* yang ada di dalam pemrograman PHP. Dependensi adalah bergantung, atau bisa di sebut banyak *file* yang bergantung dalam *file – file* yang lain, dimana satu program bergantung pada program yang lainnya (Arif, 2018).

Composer merupakan utilitas yang dapat sangat membantu kita untuk mengelola proyek terhadap sebuah *library* atau *helper / plugin*. Masa depan akan berlaku seperti ini, dimana tidak mendownload direct dari situs resminya, melainkan menggunakan composer, pernah pakai linux, tentu tidak kesulitan menggunakan composer (Febiyan, 2014).

## **Laravel**



Gambar 2. 5 Logo Laravel (Sumber: Anonim. 2019)

Laravel adalah sebuah framework PHP bersifat *open source* yang dirilis dibawah lisensi MIT dan dibangun dengan konsep MVC (model view controller). Laravel adalah pengembangan website berbasis MVP yang ditulis dalam PHP yang dirancang untuk meningkatkan kuliatas perangkat lunak dengan mengurangi biaya pemeliharaan. Laravel merupakan framework PHP terbaik yang dikembangkan oleh Taylor Otwell (Yudhanto, 2018:8).

Sebagai sebuah framework PHP, laravel hadir sebagai platform web development yang bersifat open source. Laravel juga menarik karena sintaksnya yang ekspresif dan elegan, serta dirancang khusus untuk memudahkan dan mempercepat proses web development (Yudhanto, 2018:8).

MVC adalah sebuah pendekatan perangkat lunak yang memisahkan aplikasi logika dari presentasi. MVC memisahkan aplikasi berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, controller, dan user interface (Putra, 2018).

1. Model, Model mewakili struktur data. Biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain.
2. View, View adalah bagian yang mengatur tampilan ke pengguna. Bisa dikatakan berupa halaman web.
3. Controller, Controller merupakan bagian yang menjembatani model dan view.

Adapun logo laravel dapat dilihat pada Gambar 2.5.

## **Atom Text Editor**

*Text Editor* adalah suatu *software* aplikasi atau suatu program komputer yang memungkinkan anda sebagai penggunanya untuk membuat, mengubah atau mengedit *file* teks yang ada berupa *plain text*. *Text editor* ini sebenarnya bisa digunakan untuk membuat program-program komputer dan mengedit *source code* dari bahasa pemograman. Selain itu, *text* *editor* juga bisa dimanfaatkan untuk membuat halaman web atau template web design dan juga membuat aplikasi tertentu. *Software* aplikasi satu ini memang secara umum ditujukan untuk mempermudah aktivitas pemrograman (Indra, 2016).

## **Google Chrome**



Gambar 2. 6 Logo Google Chrome (Sumber: Anonim. 2019)

Google Chrome adalah sebuah aplikasi peramban yang digunakan untuk menjelajah dunia maya seperti halnya Firefox, Opera ataupun Microsoft Edge. Jika Firefox dikembangkan oleh Mozilla, Google Chrome dibuat dan dirancang oleh Google, perusahaan internet terbesar di dunia yang juga empunya Android. Proyek *open source* yang digunakan oleh Google disebut Chromium, menggunakan mesin *rendering* Webkit sampai dengan versi 27 dan dirancang untuk bekerja dengan kecepatan di atas rata-rata namun tetap ringan dijalankan di perangkat desktop dan mobile (Winarso, 2016).

## **Website**

Suatu [halaman web](https://id.wikipedia.org/wiki/Halaman_web" \o "Halaman web) yang saling berhubungan yang umumnya berada pada [peladen](https://id.wikipedia.org/wiki/Peladen" \o "Peladen) yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau [organisasi](https://id.wikipedia.org/wiki/Organisasi" \o "Organisasi). Sebuah situs web biasanya ditempatkan setidaknya pada sebuah [server web](https://id.wikipedia.org/wiki/Server_web) yang dapat diakses melalui jaringan seperti [Internet](https://id.wikipedia.org/wiki/Internet), ataupun [jaringan wilayah lokal](https://id.wikipedia.org/wiki/Jaringan_wilayah_lokal" \o "Jaringan wilayah lokal) (LAN) melalui alamat Internet yang dikenali sebagai [URL](https://id.wikipedia.org/wiki/URL). Gabungan atas semua situs yang dapat diakses publik di Internet disebut pula sebagai [*World Wide Web*](https://id.wikipedia.org/wiki/World_Wide_Web) atau lebih dikenal dengan singkatan WWW (Harefa, 2016).

Sebuah halaman web merupakan berkas yang ditulis sebagai berkas teks biasa (*plain text*) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis HTML atau XHTML, kadang-kadang pula disisipi dengan sekelumit bahasa skrip. Berkas tersebut kemudian diterjemahkan oleh peramban web dan ditampilkan seperti layaknya sebuah halaman pada monitor komputer (Harefa, 2016).

## **UML (*Unified Modelling Language*)**

*Unified Modeling Language* adalah bahasa standar yang digunakan untuk menjelaskan dan memvisualisaikan artifak dari proses analisis dan disain berorientasi objek. UML menyediakan standar pada notasi dan diagram yang bisa digunakan untuk memodelkan suatu system. UML dikembangkan oleh 3 pendekar “berorientasi objek”, yaitu Grady Booch, Jim Rumbaugh, dan Ivar Jacobson. UML menjadi bahasa yang bisa digunakan untuk berkomunikasi dalam perspektif objek antara user dengan developer, antara developer dengan developer, antara developer analis dengan developer disain, dan antara developer disain dengan developer pemrograman (Efendi, 2014).

UML memungkinkan *developer* melakukan permodelan secara visual, yaitu penekanan pada penggambaran, bukan didominasi oleh narasi. Permodelan visual membantu untuk menangkap struktur dan kelaukan dari objek, mempermudah penggambaran interaksi antara elemen dalam sistem, dan mempertahankan konsistensi antara disain dan implementasi dalam pemrograman (Efendi, 2014).

## **Use Case Diagram**

*Use Case Diagram* adalah gambaran graphical dari beberapa atau semua *actor, use case*, dan interaksi diantaranya yang memperkenalkan suatu sistem. Use case diagram tidak menjelaskan secara detil tentang penggunaan *use case*, tetapi hanya memberi gambaran singkat hubungan antara *use case*, aktor, dan sistem.Didalam use case ini akan diketahui fungsi - fungsi apa saja yang berada pada sistem yang dibuat ([Pratama](https://www.codepolitan.com/coder/aditya90654387), 2019).

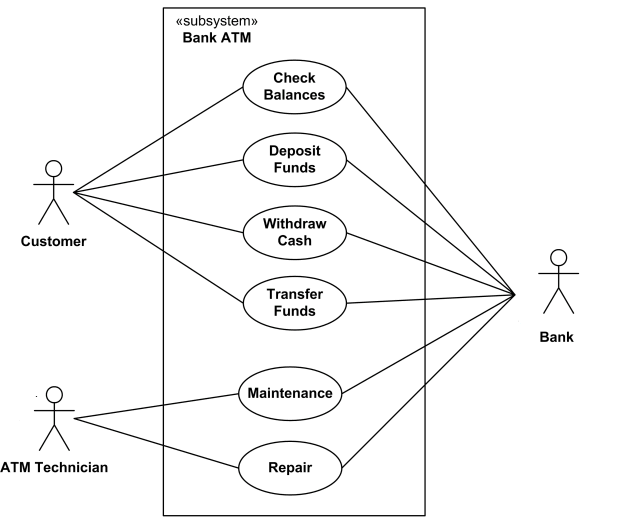
Menurut Arifwicaksanaa (2016) *use case* merupakan sebuah teknik yang digunakan dalam pengembangan sebuah software atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem yang bersangkutan, *use case* menjelaskan interaksi yang terjadi antara ‘aktor’ — inisiator dari interaksi sistem itu sendiri dengan sistem yang ada, sebuah *use case* direpresentasikan dengan urutan langkah yang sederhana. Berikut adalah simbol – simbol *use case* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol – simbol use case diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | *Actor* | Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan use case |
|  | *Include* | Menspesifikasikan bahwa use case sumber secara eksplisit |
|  | *Generalization* | Sebuah elemen yang menjadi spesialisasi dari elemen yang lain. |
|  | *Use case* | Deskripsi dari urutan aksi – aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor |

Sumber: Shy-gan

Contoh *use case* diagram dapat dilihat pada Gambar 2.3.



Gambar 2. 7 Contoh Use Case Diagram (Sumber: Ahaddin, 2015)

## ***Activity Diagram***

*Activity diagram* pada dasarnya menggambarkan macam-macam alir aktifitas yang akan dirancang dalam sebuah sistem.  Dimana masing-masing alir memiliki awal, decision yang mungkin terjadi pada sistem, dan akhir dalam sistem tersebut. Activity diagram pada dasarnya memiliki struktur yang hampir mirip dengan *flowchart* atau diagram alir dalam perancangan sistem secara terstruktur. *Activity diagram* ini dibuat berdasarkan sebuah *use case* atau beberapa *use case* dalam *use case diagram* (Ahaddin, 2015).

Adapun simbol – simbol *activity diagram* dapat dilihat pada tabel 2.2.

Tabel 2. 2 Simbol – simbol activity diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Activity | Memperlihatkan bagaimana masing – masing kelas antarmuka saling berinteraksi satu sama lain |
|  | Action | State dari sistem yang memcerminkan eksekusi dari suatu aksi |
|  | *Initial Node* | Bagaimana objek dibentuk atau diawali |
|  | *Activity Final Node* | Bagaimana objek dibentuk dan diakhiri |
|  | *Desicion* | Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu |
|  | *Line Connector* | Digunakan untuk menghubungkan satu symbol dengan simbil lain |

Sumber: Ahaddin (2015)

## ***Class Diagram***

Class diagram merupakan salah satu diagram utama dari UML untuk menggambarkan class atau blueprint object pada sebuah sistem. Analisis pembentukan class diagram merupakan aktivitas inti yang sangat mempengaruhi arsitektur piranti lunak yang dirancang hingga ke tahap pengkodean (Natalia Tanuwijaya, 2016).

Diagram kelas atau class diagram sangat membantu dalam visualisasi struktur  kelas  dari  suatu  sistem.  Hal  ini  disebabkan  karena class adalah deskripsi kelompok obyek-obyek dengan property, operasi dan relasi yang sama (Novitasari, 2018).

Berikut merupakan simbol – simbol class diagram dapat dilihat pada tabel 2.3.

Tabel 2. 3 Simbol - simbol class diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | Class | Menggambarkan sebuah kelas yang terdiri dari attribute dan method. |
|  | Relation | Menggambarkan hubungan komponen – komponen didalam static diagram. |

Sumber: Kamaludin (2012)

Adapun bentuk umum *class diagram* dapat dilihat pada Gambar 2.8.



Gambar 2. 8 Bentuk umum class diagram (Sumber: Novitasari, 2018)

## ***Flowchart***

Flowchart**atau diagram alir** adalah suatu bagan diagram dengan simbol simbol grafis yang menyatakan aliran algoritma secara detail dan prosedur sistem secara logika (Pratama, 2017).

Berikut adalah jenis-Jenis flowchart (Pratama, 2017) :

* FlowchartDokumen adalah jenis flowchart yang menunjukan kontrol sebuah sistem aliran dokumen. Bagan dari flowchart data ini menunjukan arus data dari laporan dan formulir tembusan.
* Flowchart Data adalah jenis flowchart yang menunjukkan kontrol dari sebuah sistem aliran data.
* Flowchart Sistem adalah jenis flowchart yang menunjukkan kontrol dari sebuah sistem aliran secara fisik. Bagan dari flowchartdata ini menunjukan apa yang dikerjakan oleh sebuah sistem.
* Flowchart Program adalah jenis flowchart yang menunjukkan kontrol dari sebuah program dalam sebuah sistem.Bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

Berikut merupakan simbol – simbol *flowchart* dapat dilihat pada tabel 2.4.

Tabel 2. 4 Simbol – simbol flowchart

Sumber: Salamadian (2017)

|  |  |
| --- | --- |
| https://i2.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-terminator.png?resize=142%2C90&ssl=1 | **Terminal Point Symbol / Simbol Titik Terminal** menunjukkan permulaan (start) atau akhir (stop) dari suatu proses. |
| https://i0.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart.png?resize=124%2C72&ssl=1 | **Flow Direction Symbol / Simbol Arus**adalah simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain (connecting line). Simbol ini juga berfungsi untuk menunjukkan garis alir dari proses. |
| https://i2.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-proses.png?resize=129%2C83&ssl=1 | **Processing Symbol / Simbol Proses**digunakan untuk menunjukkan kegiatan yang dilakukan oleh komputer. Pada bidang industri (proses produksi barang), simbol ini menggambarkan kegiatan inspeksi atau yang biasa dikenal dengan simbol inspeksi. |
| https://i0.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-decision.png?resize=100%2C87&ssl=1 | **Decision Symbol / Simbol Keputusan** merupakan simbol yang digunakan untuk memilih proses atau keputusan berdasarkan kondisi yang ada. Simbol ini biasanya ditemui pada flowchart program. |
| https://i1.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-input-output.png?resize=120%2C78&ssl=1 | **Input-Output / Simbol Keluar-Masuk**menunjukkan proses input-output yang terjadi tanpa bergantung dari jenis peralatannya. |
| https://i1.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-predefined-proses.png?resize=132%2C80&ssl=1 | **Predefined Process / Simbol Proses Terdefinisi** merupakan simbol yang digunakan untuk menunjukkan pelaksanaan suatu bagian prosedur (sub-proses). Dengan kata lain, prosedur yang terinformasi di sini belum detail dan akan dirinci di tempat lain |
| https://i2.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-manual-input.png?resize=121%2C78&ssl=1 | **Manual Input Symbol** digunakan untuk menunjukkan input data secara manual menggunakan online keyboard. |
| https://i2.wp.com/salamadian.com/wp-content/uploads/2018/12/flowchart-simbol-operasi-manual.png?resize=116%2C78&ssl=1 | **Manual Operation Symbol / Simbol Kegiatan Manual**digunakan untuk menunjukkan kegiatan/proses yang tidak dilakukan oleh komputer. |
|  | **Document Symbol**  Jika Anda menemukan simbol ini artinya input berasal dari dokumen dalam bentuk kertas, atau output yang perlu dicetak di atas kertas. |
| https://i0.wp.com/salamadian.com/wp-content/uploads/2018/01/simbol-flowchart-multiple-document.png?resize=143%2C120&ssl=1 | **Multiple Documents** sama seperti document symbol hanya saja dokumen yg digunakan lebih dari satu dalam simbol ini |

Sumber: Salamadian (2017)

Tabel 2. 4 Simbol – simbol flowchart (lanjutan)

## ***Entity* *Relationship Diagram***

*Entity Relationship Diagram* (ERD) adalah suatu model untuk menjelaskan suatu hubungan data – data pada suatu basis data.Peranan *Entity Relationship Diagram* (ERD) adalah untuk merangkai data – data yang saling berhubungan berdasarkan objek – objek yang ada pada suatu basis data yang telah dihubungkandengan suatu relasi. Setelah kita mengetahui peranan Entity Relationship Diagram (ERD), berikutnya saya akan memberitahukan apa saja fungsi dari *Entity Relationship Diagram* (ERD)itu sendiri (Julianto, 2016).

Adapun simbol – simbol ERD dapat dilihat pada tabel 2.6.

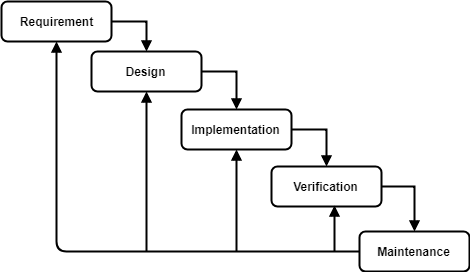
Tabel 2. 5 Simbol – simbol ERD

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Entitas | Kumpulan dari objek yang dapat diidentifikasikan secara unik. |
|  | Relasi | Hubungan yang terjadi antara salah satu lebih entitas. Jenis hubungan antara lain. One to one, one to many, dan manu to many. |
|  | Atribut | Karakteristik dari entitas atau relasi yang merupakan penjelasan detail tentang entitas. |
|  | Garis | Hubungan antara entitas dengan atributnya dan himpunan entitas dengan himpunan relasinya. |

Sumber: Setiadi (2007)

## **Metode Waterfall**

Metode air terjun atau yang sering disebut metode *waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan pada pengembangan perangkat lunak, dimulai dengan spesifikasi kebutuhan pengguna lalu berlanjut melalui tahapan-tahapan perencanaan (*planning*), permodelan (*modeling*), konstruksi (*construction*), serta penyerahan sistem ke para pelanggan/pengguna (*deployment*), yang diakhiri dengan dukungan pada perangkat lunak lengkap yang dihasilkan (Galandi, 2016).



Gambar 2.6 Model Metode Waterfall (Sumber: Galandi, 2016)

## **2.21** **Kuesioner**

Kuesioner adalah daftar pertanyaan yang dikirim kepada responden baik secara Iangsung maupun tidak Iangsung. Kuesioner atau yang sering juga disebut dengan istilah angket secara umum berupa pertanyaan atau pernyataan yang dapat dijawab sesuai dengan bentuk angket yang diberikan (Rohman, 2019).

Berikut merupakan jenis – jenis kuesioner (Rohman, 2019) :

* **Kuesioner terbuka** merupakan daftar pertanyaan yang memberi kesempatan kepada responden untuk menuliskan pendapat mengenal pertanyaan yang diberikan peneliti.
* **Kuesioner tertutup** merupakan daftar pertanyaan yang alternatif jawabannya telah disediakan oleh peneliti. Cara ini seringkali dianggap efektif karena responden dapat Iangsung membubuhkan tanda centang (√) dalam kolom yang disediakan. **Contoh kuesioner tertutup** sebagai berikut.

## **2.22 Skala Likert**

Skala likert adalah skala pengukuran yang dikembangkan oleh Likert (1932). Skala likert mempunyai empat atau lebih butir-butir pertanyaan yang dikombinasikan sehingga membentuk sebuah skor/nilai yang merepresentasikan sifat individu, misalkan pengetahuan, sikap, dan perilaku. Dalam proses analisis data, komposit skor, biasanya jumlah atau rataan, dari semua butir pertanyaan dapat digunakan. Penggunaan jumlah dari semua butir pertanyaan valid karena setiap butir pertanyaan adalah indikator dari variabel yang direpresentasikannya (Budiaji, 2013).

Berikut merupakan keterangan dari kriteria penilaian skala likert menurut Ubaid Salim (2017:524) dapat dilihat pada tabel 2.7 :

Tabel 2. 6 Kriteria Penilaian Skala Likert

|  |  |
| --- | --- |
| **Skala Jawaban** | **Nilai** |
| Sangat baik (A) | 4 |
| Baik (B) | 3 |
| Cukup (C) | 2 |
| Kurang (D) | 1 |

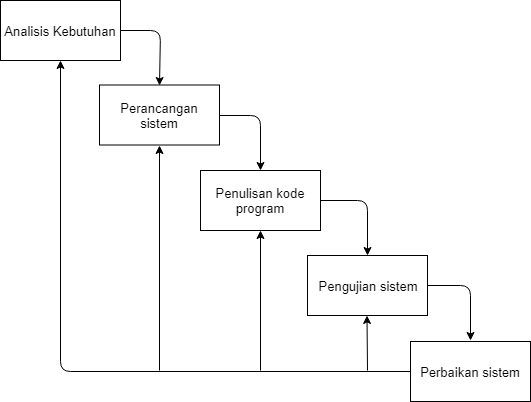
# **BAB III**

# **METODOLOGI PENELITIAN**

## **Metode Penelitian**

Metode yang penulis gunakan dalam penelitian ini adalah metode *waterfall*. *Waterfall* apabila diartikan secara *literature* berarti air terjun. Namun demikian, bagi ilmu komputer dan juga teknologi informasi, *waterfall* merupakan salah satu jenis metode yang digunakan dalam melakukan sebuah pengembangan sistem (Dini, 2016).

Langkah – langkah pembuatan sistem informasi tugas akhir berbasis web dapat ditunjukan pada Gambar 3.1



Gambar 3. 1 Alur Metode Penelitian

Adapun penjelasan dari Gambar 3.1 yaitu sebagai berikut :

* + - 1. **Analisi Kebutuhan**

Sistem yang digunakan untuk pengolahan administrasi tugas akhir pada Jurusan Teknik Informatika Politeknik Negeri Indramayu masih terkomputerisasi secara sederhana belum ada sistem khusus yang digunakan untuk mengelola tugas akhir. Sistem yang diusulkan kali ini adalah sistem yang terkomputerisasi berbasis website, sehingga semakin mudah menggunakannya karena data yang dikelola dapat terstruktur dengan baik sehingga dapat mempermudah semua pihak yang memanajemen tugas akhir.

* 1. **Observasi**

Penulis melakukan pengamatan dari data – data yang didapatkan yang kemudian penulis menyimpulkan ke dalam proses yang terstruktur untuk di implementasikan ke dalam sistem aplikasi berbasis web. Data – data tersebut dihasilkan melalui tanya jawab yang dilakukan penulis dengan koordinator TA. Adapun data yang didapatkan berupa :

* Data mahasiswa tingkat 3
* Data dosen
* Fitur – fitur yang dibutuhkan didalam aplikasi sistem informasi tugas akhir.
  1. **Studi Pustaka**

Penulis mencari bahan – bahan atau referensi dari berbagai sumber seperti jurnal ilmiah dan internet yang dapat digunakan sebagai referensi penulis dalam pembuatan sistem informasi tugas akhir berbasis web menggunakan framework laravel yang dapat digunakan oleh mahasiswa dan dosen pada jurusan Teknik Informatika Politeknik Negeri Indramayu.

* + - 1. **Perancangan Sistem**

Tahapan berikutnya adalah pembuatan rancangan dari sebuah sistem. Pada tahapan ini penulis menggunakan *flowchart*, UML (*Unified Modelling Language*), dan perancangan antarmuka (*mockup*) untuk memberikan gambaran mengenai alur sistem. Selain itu penulis menggunakan *use case diagram* yang digunakan selama proses analisa untuk menangkap *requirement* sistem dan untuk memahami bagaimana sistem bekerja.

* + - 1. **Penulisan Kode Program**

Untuk menjalankan desain sistem yang sudah dibuat, maka desain sistem tersebut diterjemahkan ke dalam kode dan juga *script*, sehingga nantinya desain dari sistem tersebut bisa berjalan dengan lancar dan juga baik.

* + - 1. **Pengujian Sistem**

Pengujian sistem dilakukan dengan melakukan setiap proses untuk menguji fungsionalitas sistem dan mengamati kemungkinan kesalahan yang terjadi pada setiap proses. Pengujian ini dilakukan menggunakan kuisioner dengan memberikan beberapa pertanyaan kepada pengguna untuk menentukan kelayakan penggunaan aplikasi.

* + - 1. **Perbaikan Sistem**

Tahap akhir dari model *Waterfall* adalah **perbaikan sistem** yang sudah dibuat, dan dijalankan. Perbaikan sistem termasuk dalam memperbaiki kesalahan-kesalahan yang tidak ditemukan pada langkah sebelumnya.

## **Perancangan Sistem**

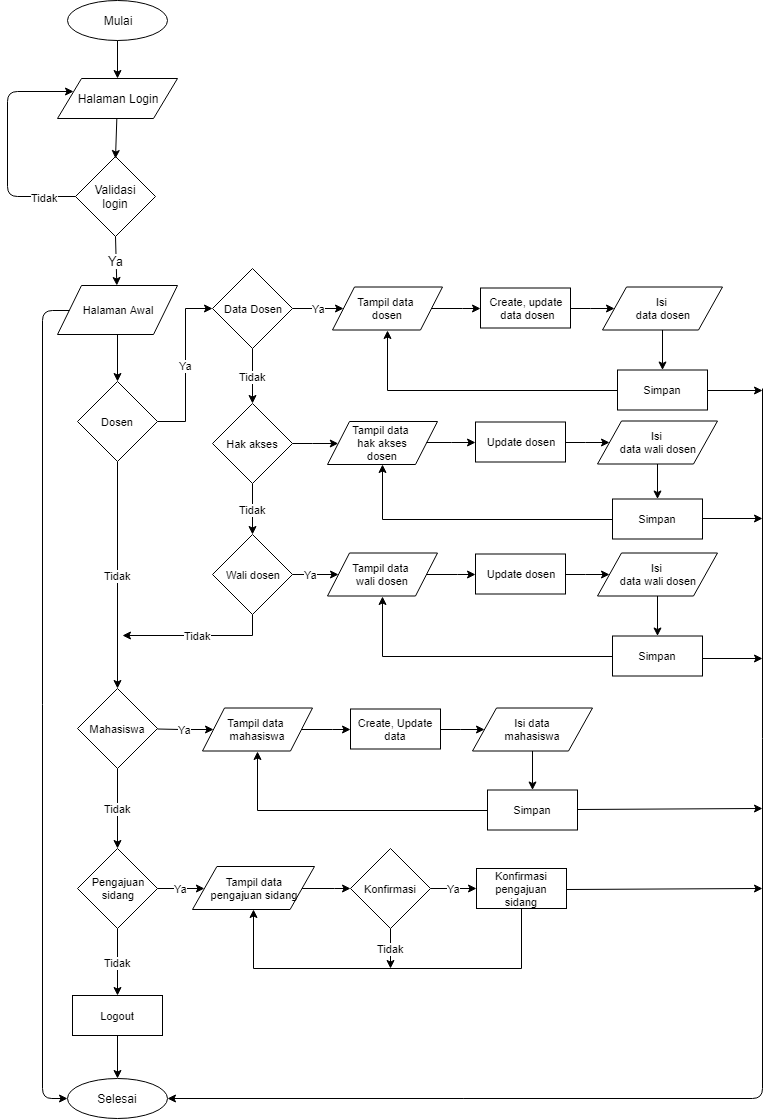
Perancangan sistem ini menggunakan *flowchart*, UML (*Unified Modelling Language*),dan perancangan antarmuka (*mockup*) untuk memberikan gambaran mengenai alur sistem.. Pada perancangan ini akan digambarkan secara garis besar mengenai aplikasi Sistem Informasi Tugas Akhir Berbasis Web Menggunakan Framework Laravel yang akan dijelaskan pada sub bab selanjutnya.

## ***Flowchart***

*Flowchart* merupakan gambaran alur di dalam sistem yang sedang dirancang, yang menunjukan bagaimana alur penggunaan pada masing – masing fitur yang ada di dalam sistem yang sedang dirancang.

1. ***Flowchart* Administrator**

Rancangan *flowchart* administrator terdiri dari tiga menu utama yaitu dosen, mahasiswa, dan pengajuan sidang. Dalam menu dosen, administrator dapat mengelola data dosen, akun dosen, hak akses dosen, serta menambahkan data wali dosen. Dalam menu mahasiswa, administrator dapat mengelola data mahasiswa, dan akun mahasiswa. Dan dalam menu pengajuan sidang administrator dapat melakukan konfirmasi pengajuan sidang yang dilakukan oleh mahasiswa dan telah disetujui oleh dosen pembimbing. Adapun rancangan flowchart administrator dapat dilihat pada Gambar 3.2.



Gambar 3. 2 Flowchart Administrator

1. ***Flowchart* Mahasiswa**

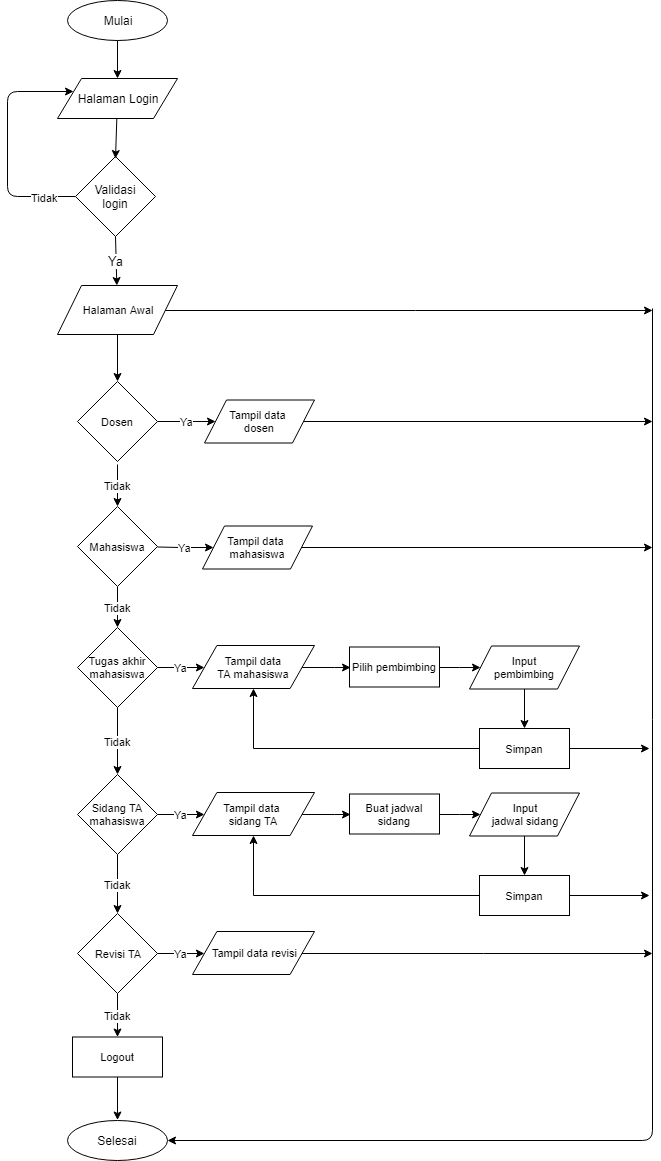
Rancangan *flowchart* mahasiswa terdiri dari tiga menu utama yaitu menu tugas akhir yang digunakan untuk melakukan pengajuan tugas akhir, menu sidang TA yang digunakan untuk melakukan pengajuan sidang TA serta melihat jadwal sidang TA, dan menu revisi yang digunakan untuk mengunggah laporan TA yang telah diperbaiki.



Gambar 3. 3 Flowchart Mahasiswa

1. ***Flowchart* Koordinator**

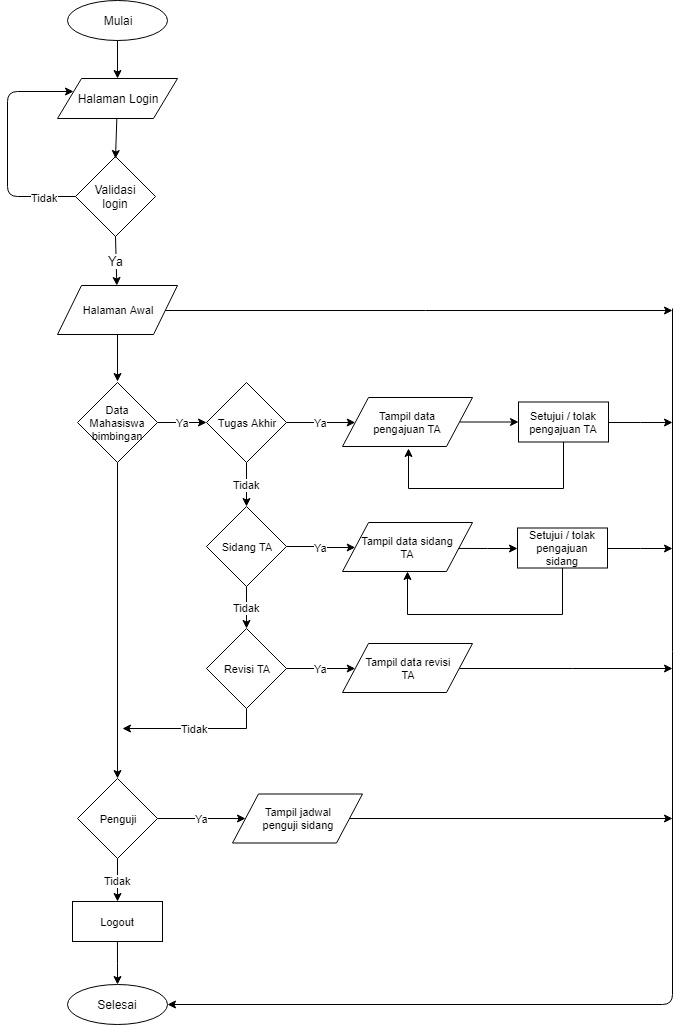
Rancangan *flowchart* koordinator terdiri dari lima menu yaitu menu dosen yang digunakan untuk melihat data dosen, menu mahasiswa yang digunakan untuk melihat data mahasiswa, menu tugas akhir mahasiswa yang digunakan untuk melihat data pengajuan TA mahasiswa serta untuk menentukan dosen pembimbing, menu sidang TA yang digunakan untuk melihat data pengajuan sidang TA dan membuat jadwal sidang TA mahasiswa yang telah disetujui oleh dosen pembimbing dan telah dikonfirmasi oleh administrator, dan menu revisi digunakan untuk melihat laporan TA mahasiswa yang sudah diperbaiki.



Gambar 3. 4 Flowchart Koordinator

1. ***Flowchart* Pembimbing**

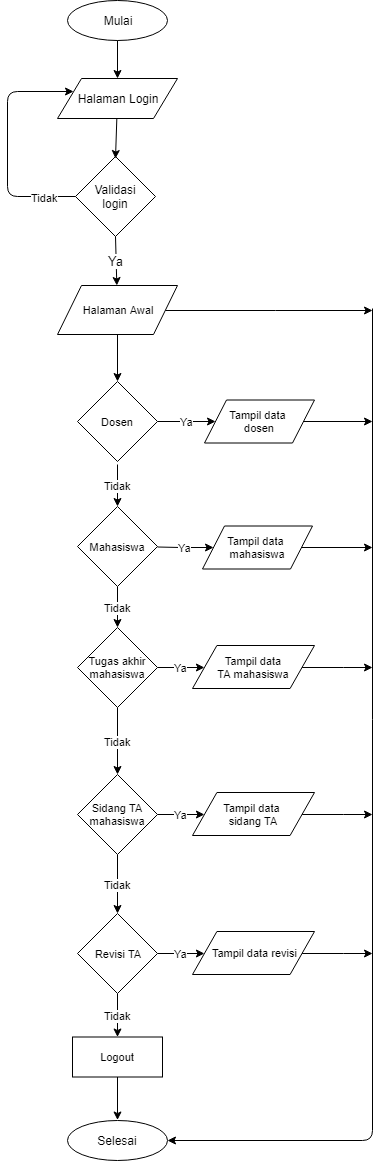
Rancangan *flowchart* pembimbing terdiri dari dua menu yaitu menu data mahasiswa bimbingan yang digunakan untuk melihat data mahasiswa bimbingan seperti pengajuan TA, pengajuan sidang, serta pembimbing dapat melakukan persetujuan terhadap pengajuan TA dan pengajuan sidang dari mahasiswa, dan menu penguji yang digunakan untuk melihat jadwal dosen tersebut menjadi penguji sidang TA mahasiswa.



Gambar 3. 5 Flowchart Pembimbing

1. ***Flowchart* Ketua Jurusan**

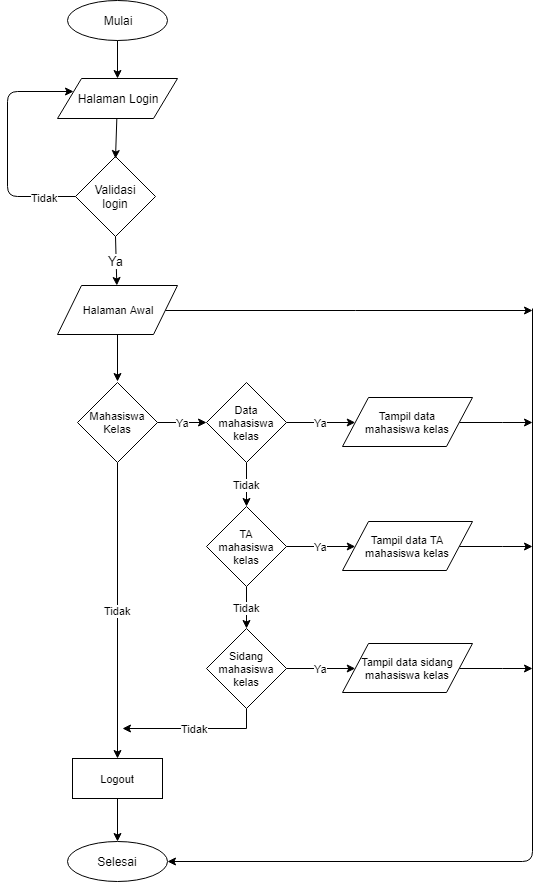
Rancangan *flowchart* ketua terdiri dari lima menu yaitu menu dosen yang digunakan untuk melihat data dosen, menu mahasiswa yang digunakan untuk melihat data mahasiswa, menu tugas akhir mahasiswa yang digunakan untuk melihat data pengajuan TA mahasiswa, menu sidang TA yang digunakan untuk melihat data pengajuan sidang TA, dan menu revisi digunakan untuk melihat laporan TA mahasiswa yang sudah diperbaiki.



Gambar 3. 6 Flowchart Ketua Jurusan

1. ***Flowchar*t Wali Dosen**

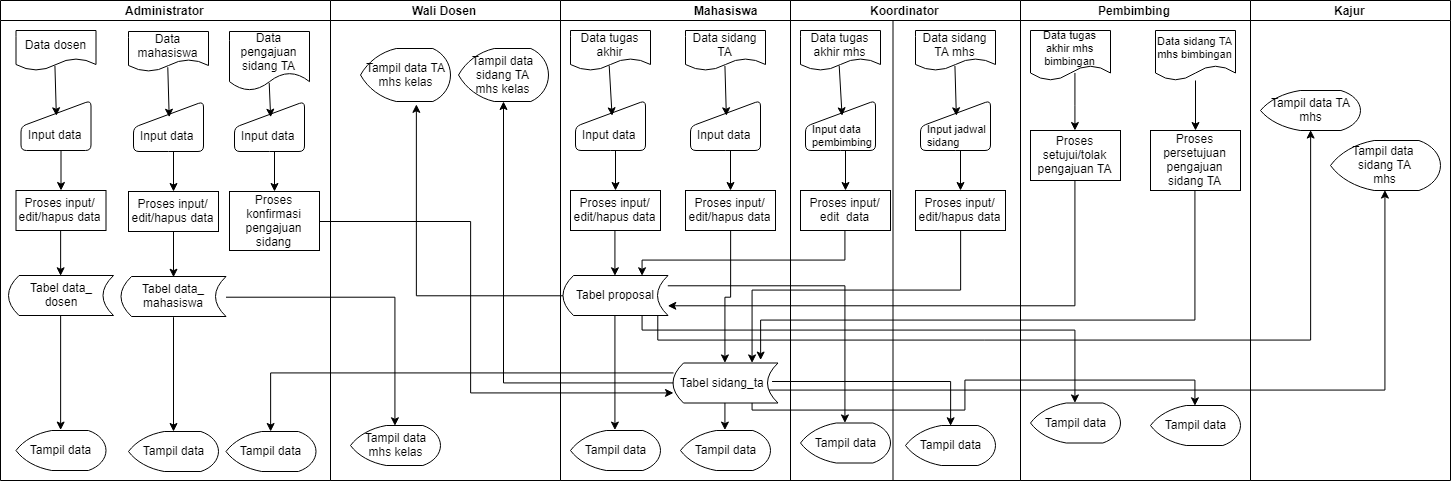
Rancangan *flowchart* wali dosen terdiri dari satu menu yaitu menu mahasiswa kelas dimana wali dosen dapat melihat data mahasiswa kelasnya, seperti data tugas akhir dan dapat melihat data sidang TA mahasiswa kelasnya.



Gambar 3. 7 Flowchart Wali Dosen

1. ***Flowchart System***

*Flowchart system* dari pembuatan aplikasi sistem informasi tugas akhir berbasis web ini, yaitu suatu bagan yang memperlihatkan urutan *procedure* dan proses dari beberapa *filed* di dalam beberapa fitur tertentu. Adapun *flowchart system* dari pembuatan aplikasi ini dapat dilihat pada Gambar 3.8.



Gambar 3. 8 Flowchart System

## ***Use Case Diagram***

Pada *use case diagram* yang dibuat oleh penulis dalam pembuatan aplikasi sistem informasi tugas akhir ini memiliki beberapa aktor diantaranya aktor administrator, aktor mahasiswa, aktor koordinator, aktor pembimbing, aktor kajur, dan aktor wali dosen, dimana setiap aktor tersebut memiliki hak akses yang berbeda – beda dalam penggunaan aplikasi sistem informasi tugas akhir berbasis *web* ini.

Aktor administrator memiliki hak akses untuk mengelola data mahasiswa, akun mahasiswa, data dosen, akun dosen, dan melakukan konfirmasi sidang tugas akhir apabila pengajuan sidang sudah disetujui oleh dosen pembimbing.

Aktor mahasiswa memiliki hak akses untuk melakukan pengajuan proposal tugas akhir, melakukan pengajuan sidang tugas akhir, serta mengunggah file laporan tugas akhir yang sudah diperbaiki atau direvisi.

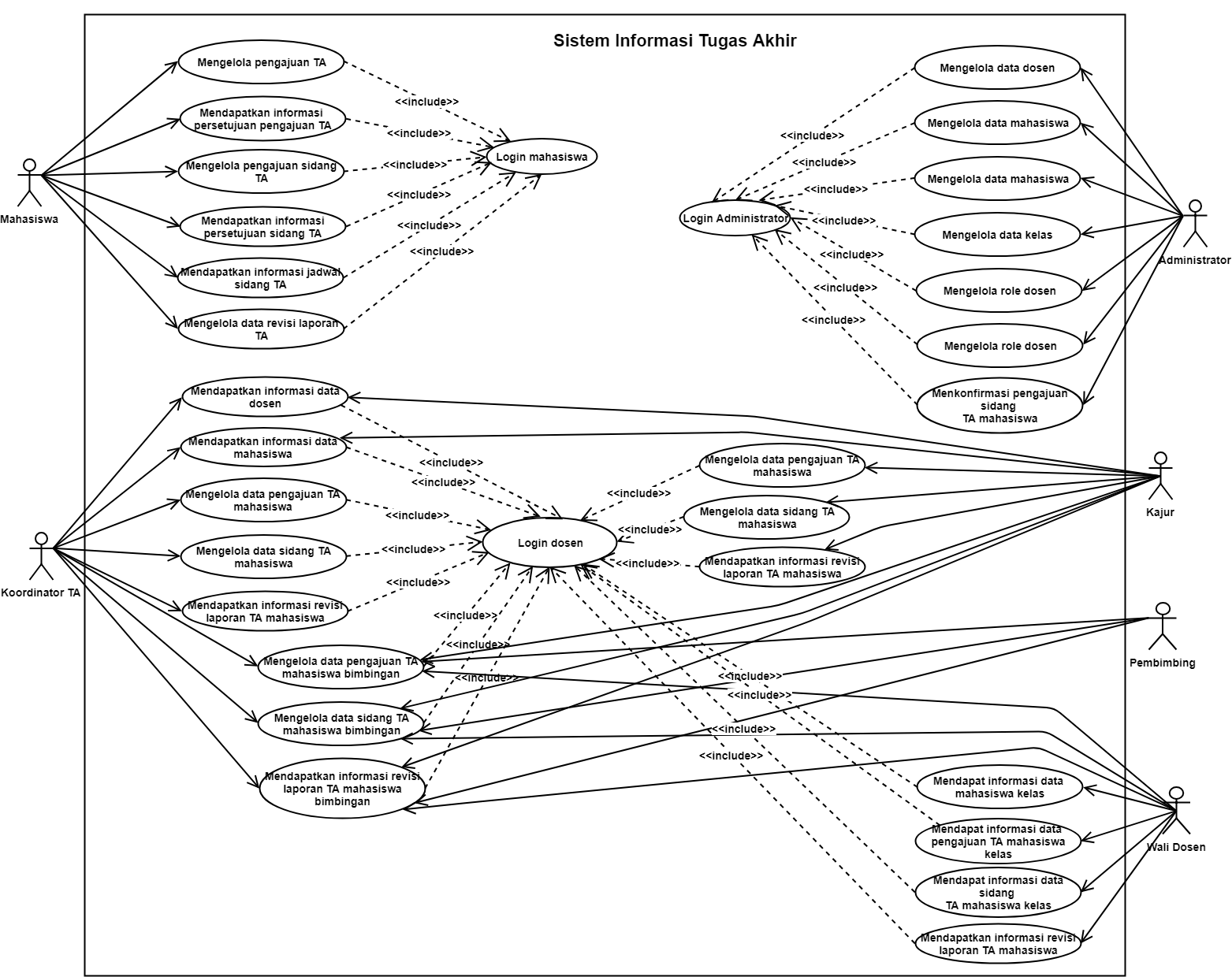
Aktor koordinator memiliki hak akses untuk melihat informasi mengenai tugas akhir mahasiswa dan mengelola data tugas akhir mahasiswa seperti menentukan pembimbing dan melakukan penjadwalan sidang tugas akhir apabila pengajuan sidang sudah disetujui oleh dosen pembimbing dan sudah dikonfirmasi oleh admin.

Aktor pembimbing memiliki hak akses untuk melihat informasi mengenai tugas akhir mahasiswa bimbingannya dan mengelola data tugas akhir mahasiswa bimbingannya seperti menyetujui ataupun menolak pengajuan tugas akhir mahasiswa dan menyetujui pengajuan sidang tugas akhir mahasiswa.

Aktor kajur memiliki hak akses untuk melihat informasi data mahasiswa, data dosen, serta data tugas akhir mahasiswa. Aktor kajur hanya dapat melihat data mengenai tugas akhir dan tidak dapat mengelola data tugas akhir.

Aktor wali dosen memiliki hak akses untuk melihat data mahasiswa kelasnya dan data tugas akhir khusus milik mahasiswa kelasnya seperti data pengajuan tugas akhir dan data pengajuan sidang tugas akhir.

Berikut adalah use case diagram dari aplikasi Sistem Informasi Tugas Akhir berbasis *web* dapat dilihat pada Gambar 3.9.



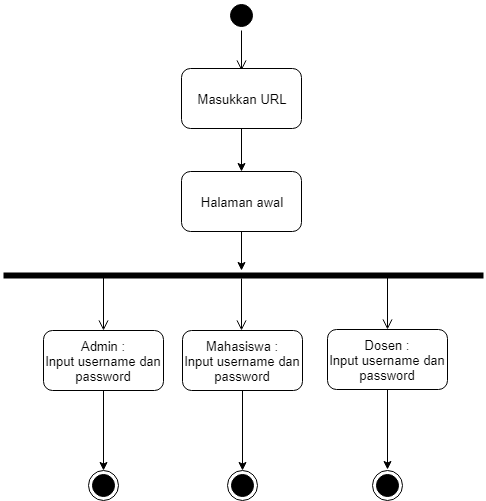
Gambar 3. 9 Use Case Diagram

## ***Activity Diagram***

*Activity diagram* merupakan gambaran aktifitas dalam sistem yang sedang dirancang, bagaimana masing – masing alur dirancang dan bagaimana itu berakhir.

1. ***Activity Diagram Login User***

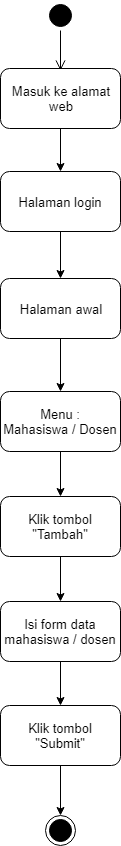
*User* harus melakukan login untuk dapat menggunakan fitur – fitur yang ada pada sistem. Terdapat 3 jenis *user* yaitu administrator, mahasiswa, dan dosen. Rancangan *activity diagram login user* dapat dilihat pada Gambar 3.10.



Gambar 3. 10 Activity Diagram Login User

1. ***Activity Diagram* Tambah data mahasiswa / dosen**

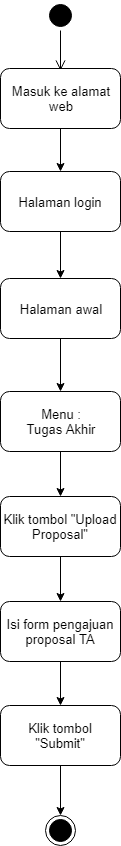
Mahasiswa dan dosen dapat menggunakan fitur – fitur yang ada pada sistem apabila admin sudah menambahkan akun. Maka admin harus menambahkan data mahasiswa dan dosen sehingga mahasiswa dan dosen tersebut mendapatkan akun untuk bisa masuk ke dalam sistem. Rancangan *activity diagram* tambah data mahasiswa / dosen dapat dilihat pada Gambar 3.11.



Gambar 3. 11 Activity Diagram Tambah data mahasiswa / dosen

1. ***Activity Diagram* Pengajuan Proposal Tugas Akhir**

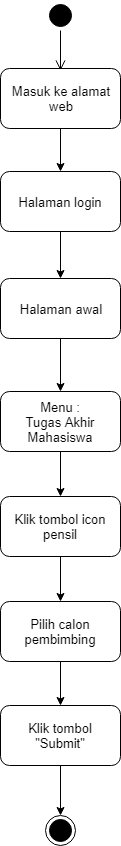
Mahasiswa yang telah memiliki akun dapat menggunakan fitur dalam sistem salah untuk pengajuan proposal tugas akhir dengan mengisi form pengajuan sidang serta mengunggah *file* proposal tugas akhir. Rancangan *activity diagram* pengajuan sidang tugas akhir dapat dilihat pada Gambar 3.12.



Gambar 3. 12 Activity Diagram Pengajuan Proposal Tugas Akhir

1. ***Activity Diagram* Pembagian Pembimbing**

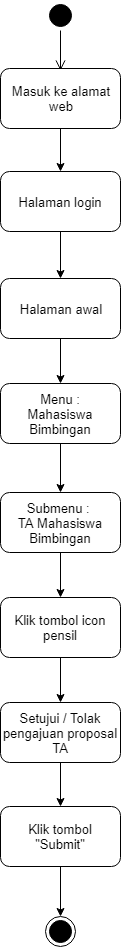
Setelah melakukan pengajuan proposal TA, koordinator TA dapat melakukan pemilihan pembimbing untuk setiap mahasiswa yang melakukan pengajuan proposal TA. Rancangan *activity diagram* pemilihan pembimbing dapat dilihat pada Gambar 3.13.



Gambar 3. 13 Activity Diagram Pemilihan Pembimbing

1. ***Activity Diagram* Setujui / Tolak Pengajuan Proposal Tugas Akhir**

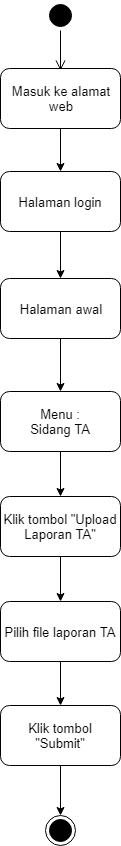
Setelah koordinator TA melakukan pemilihan pembimbing maka dosen pembimbing dapat menyetujui ataupun menolak pengajuan proposal TA mahasiswa bimbingannya. Rancangan *activity diagram* setujui / tolak pengajuan proposal tugas akhir dapat dilihat pada Gambar 3.14.



Gambar 3. 14 Activity Diagram Setujui / Tolak Pengajuan Proposal Tugas Akhir

1. ***Activity Diagram* Pengajuan Sidang Tugas Akhir**

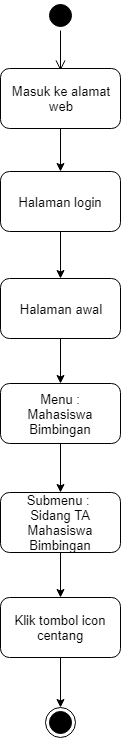
Mahasiswa juga dapat melakukan pengajuan sidang TA dengan mengunggah file Laporan Tugas Akhir. Rancangan *activity diagram* pengajuan sidang tugas akhir dapat dilihat pada Gambar 3.15.



Gambar 3. 15 Activity Diagram Pengajuan Sidang Tugas Akhir

1. ***Activity Diagram* Setujui Pengajuan Sidang Tugas Akhir**

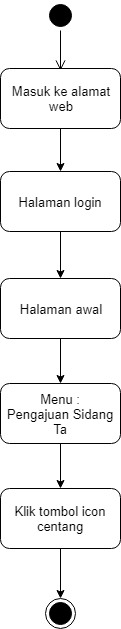
Setelah mahasiswa melakukan pengajuan sidang TA, selanjutnya dosen pembimbing melakukan persetujuan terhadap pengajuan sidang TA tersebut. Rancangan *activity diagram* setujui pengajuan sidang tugas akhir dapat dilihat pada Gambar 3.16.



Gambar 3. 16 Activity Diagram Setujui Pengajuan Sidang Tugas Akhir

1. ***Activity Diagram* Konfirmasi Pengajuan Sidang Tugas Akhir**

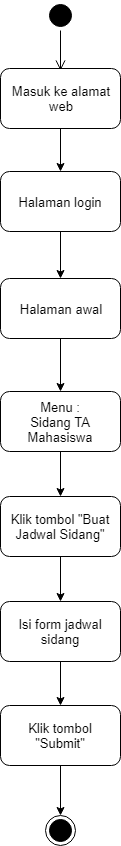
Apabila pengajuan sidang TA mahasiswa disetujui oleh dosen pembimbing maka admin melakukan konfirmasi pengajuan sidang TA. Rancangan *activity diagram* konfirmasi pengajuan sidang tugas akhir oleh admin dapat dilihat pada Gambar 3.17.



Gambar 3. 17 Activity Diagram Konfirmasi Pengajuan Sidang Tugas Akhir

1. ***Activity Diagram* Penjadwalan Sidang Tugas Akhir**

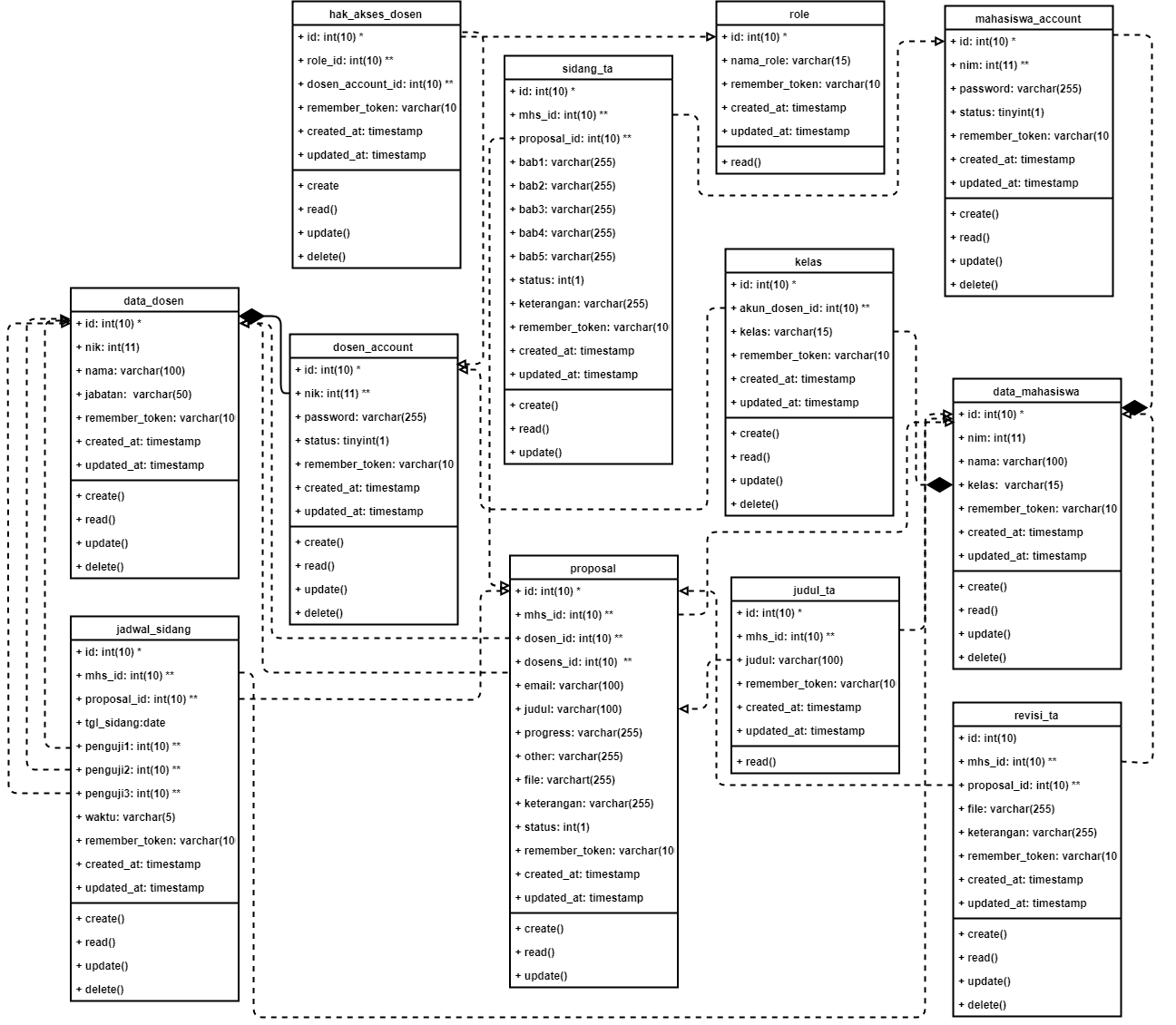
Koordinator TA dapat membuat jadwal sidang apabila pengajuan sidang TA telah disetujui oleh kedua dosen pembimbing dan dikonfirmasi oleh admin. Rancangan *activity diagram* penjadwalan sidang tugas akhir dapat dilihat pada Gambar 3.18.



Gambar 3. 18 Activity Diagram Penjadwalan Sidang Tugas Akhir

## ***Class Diagram***

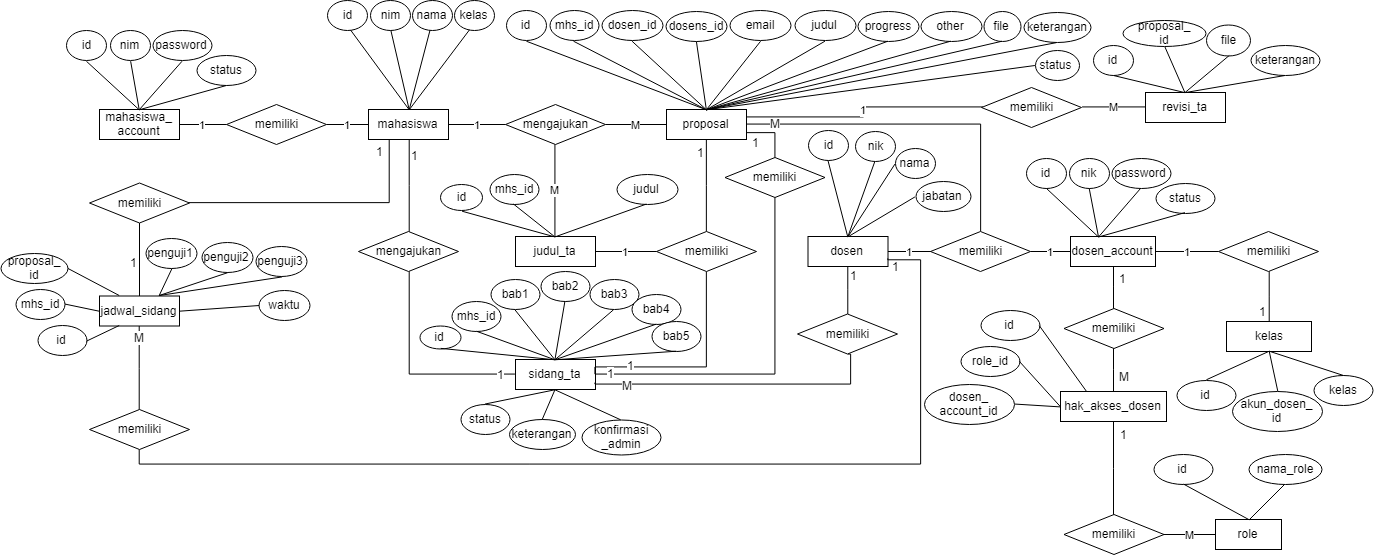
*Class diagram* menunjukan interaksi antar kelas dalam sistem serta memperlihatkan struktur dari sistem berupa kelas, atribut, dan relasi antar kelas. Adapun *class diagram* dari aplikasi ini terdapat pada Gambar 3.19.



Gambar 3. 19 Class Diagram

## ***Entity Relationship Diagram***

*Entity Relationship Diagram* (ERD)menunjukan struktur tabel dan memperlihatkan relasi antar tabel dalam *database* yang digunakan dalam pembuatan aplikasi sistem informasi tugas akhir ini. Adapun class diagram dari aplikasi ini terdapat pada Gambar 3.20.



Gambar 3. 20 Entity Relationship Diagram (ERD)

## **Normalisasi *Database***

Berikut merupakan hasil normalisasi *database* pada aplikasi Sistem Informasi Tugas Akhir (SISTA) Berbasis WebMenggunakan Framework Laravel.

1. **Tabel *Unnormal***

Berikut merupakan tabel proposal dalam bentuk *unnormal* dapat dilihat pada tabel 3.1.

Tabel 3. 1 Tabel Proposal Unnormal

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | mhs\_id | nim | nama | kelas | dosen\_id | dosens\_id | nm\_dosen1 | nm\_dosen2 | judul | file |
| 1 | 1 | 1603034 | Amelia | D3TI.3B | 1 | 2 | Eka Ismantohadi | Muhamad Mustamiin | Sistem Informasi TA | Amelia proposal.docx |
| 2 |  |  |  |  | 4 | 3 | A. Lubis Ghozali | Muhammad Anis Al Hilmi | Aplikasi Manajemen Restoran | Moeh. Rifaldhi.docx |
| 3 |  |  |  |  | 4 | 5 | A. Lubis Ghozali | Fachrul Pralienka Bani M. | Aplikasi Quick Count Pilkada | Ambar N.Q.docx |
| 4 | 4 | 1603056 | Samsul | D3TI.3B | 2 | 6 | A. Lubis Ghozali | Esti Mulyani | Aplikasi Perhitungan Honor Dosen | Samsul.docx |

1. **Tabel Normalisasi Kesatu (1NF)**

Proses selanjutnya terdapat bentuk normalisasi kesatu (1NF) dari suatu tabel unnormal. Berikut ini adalah tabel normalisai kesatu (1NF) dapat dilihat pada tabel 3.2.

Tabel 3. 2 Tabel Proposal Normalisasi Kesatu (1NF)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| id | mhs\_id | nim | nama | kelas | dosen\_id | dosens\_id | nm\_dosen1 | nm\_dosen2 | judul | file |
| 1 | 1 | 1603034 | Amelia | D3TI.3B | 1 | 2 | Eka Ismantohadi | Muhamad Mustamiin | Sistem Informasi TA | Amelia proposal.docx |
| 2 | 2 | 1603048 | Moeh. Rifaldhi | D3TI.3B | 4 | 3 | A. Lubis Ghozali | Muhammad Anis Al Hilmi | Aplikasi Manajemen Restoran | Moeh. Rifaldhi.docx |
| 3 | 3 | 1603033 | Ambar N.Q | D3TI.3B | 4 | 5 | A. Lubis Ghozali | Fachrul Pralienka Bani M. | Aplikasi Quick Count Pilkada | Ambar N.Q.docx |
| 4 | 4 | 1603056 | Samsul | D3TI.3B | 2 | 6 | A. Lubis Ghozali | Esti Mulyani | Aplikasi Perhitungan Honor Dosen | Samsul.docx |

1. **Tabel Normalisasi Kedua (2NF)**

Proses selanjutnya pemisahan tabel atau pembuatan tabel baru dalam bentuk normalisasi kedua (2NF). Berikut ini adalah tabel normalisai kedua (2NF) dapat dilihat pada tabel di bawah ini.

Tabel 3. 3 Tabel proposal Normalisasi Kedua (2NF)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| id | mhs\_id | dosen\_id | dosens\_id | judul | file |
| 1 | 1 | 1 | 2 | Sistem Informasi TA | Amelia proposal.docx |
| 2 | 2 | 4 | 3 | Aplikasi Manajemen Restoran | Moeh. Rifaldhi.docx |
| 3 | 3 | 4 | 5 | Aplikasi Quick Count Pilkada | Ambar N.Q.docx |
| 4 | 4 | 2 | 6 | Aplikasi Perhitungan Honor Dosen | Samsul.docx |

Tabel 3. 4 Tabel data\_mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| id | Nim | nama | kelas |
| 1 | 1603034 | Amelia | D3TI.3B |
| 2 | 1603048 | Moeh. Rifaldhi | D3TI.3B |
| 3 | 1603033 | Ambar N.Q | D3TI.3B |
| 4 | 1603056 | Samsul | D3TI.3B |

Tabel 3. 5 Tabel data mahasiswa

|  |  |
| --- | --- |
| id | nama |
| 1 | Muhamad Mustamiin |
| 2 | Muhammad Anis Al Hilmi |
| 3 | Fachrul Pralienka Bani M. |
| 4 | Esti Mulyani |

1. **Tabel Normalisasi Ketiga (3NF)**

Proses selanjutnya menghilangkan ketergantungan terhadap *key*-nya dalam bentuk normalisasi ketiga (3NF). Berikut ini adalah tabel normalisai ketiga (3NF) dapat dilihat pada tabel di bawah ini.

Tabel 3. 6 Tabel proposal Normalisasi Ketiga (3NF)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | mhs\_id | dosen\_id | dosens\_id | file |
| 1 | 1 | 1 | 2 | Amelia proposal.docx |
| 2 | 2 | 4 | 3 | Moeh. Rifaldhi.docx |
| 3 | 3 | 4 | 5 | Ambar N.Q.docx |
| 4 | 4 | 2 | 6 | Samsul.docx |

Tabel 3. 7 Tabel judul\_ta Normalisasi Ketiga (3NF)

|  |  |  |
| --- | --- | --- |
| id | mhs\_id | judul |
| 1 | 2 | Sistem Informasi TA |
| 4 | 3 | Aplikasi Manajemen Restoran |
| 4 | 5 | Aplikasi Quick Count Pilkada |
| 2 | 6 | Aplikasi Perhitungan Honor Dosen |

## **Perancangan *Database***

Berikut ini merupakan pernacangan dari tabel – tabel dalam database pembuatan aplikasi Sistem Informasi Tugas Akhir (SISTA) Berbasis WebMenggunakan Framework Laravel.

## **Tabel admin\_account**

Tabel ini digunakan untuk menyimpan data login administrator. Daftar kolom – kolom tabel admin\_account dapat dilihat pada Tabel 3.8.

Tabel 3. 8 Tabel admin\_account

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| Nama | varchar(100) | Menyimpan nama |
| Username | varchar(20) | Menyimpan username |
| Password | varchar(255) | Menyimpan password |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel data\_dosen**

Tabel ini digunakan untuk menyimpan dan mengambil data dosen yang isi data sebelumnya sudah ada. Daftar kolom – kolom tabel data\_dosen dapat dilihat pada Tabel 3.9.

Tabel 3. 9 Tabel data\_dosen

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| Nik | int(11) | Menyimpan NIK |
| Nama | varchar(100) | Menyimpan nama |
| Jabatan | varchar(50) | Menyimpan jabatan |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel data\_mahasiswa**

Tabel ini digunakan untuk menyimpan dan mengambil data mahasiswa yang isi data sebelumnya sudah ada. Daftar kolom – kolom tabel data\_mahasiswa dapat dilihat pada Tabel 3.10.

Tabel 3. 10 Tabel data\_mahasiswa

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| Nim | int(11) | Menyimpan NIM |
| Nama | varchar(100) | Menyimpan nama |
| Kelas | varchar(15) | Menyimpan kelas |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel dosen\_account**

Tabel ini digunakan untuk menyimpan data login dosen. Daftar kolom – kolom tabel dosen\_account dapat dilihat pada Tabel 3.11.

Tabel 3. 11 Tabel dosen\_account

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| Nik | int(11) | Menyimpan username berdasarkan NIK |
| password | varchar(255) | Menyimpan password |
| status | boolean | Menyimpan status akun |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel judul\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data judul tugas akhir yang isi data sebelumnya sudah ada. Daftar kolom – kolom tabel judul\_ta dapat dilihat pada Tabel 3.12.

Tabel 3. 12 Tabel judul\_ta

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| mhs\_id | int(10) | Menyimpan id mahasiswa |
| Judul | varchar(191) | Menyimpan judul tugas akhir |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel kelas**

Tabel ini digunakan untuk menyimpan dan mengambil data kelas beserta dosen walinya yang isi data sebelumnya sudah ada. Daftar kolom – kolom tabel kelas dapat dilihat pada Tabel 3.13.

Tabel 3. 13 Tabel kelas

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| dosen\_id | int(10) | Menyimpan id dosen |
| Kelas | varchar(15) | Menyimpan judul tugas akhir |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel role**

Tabel ini digunakan untuk menyimpan dan mengambil data role untuk login dosen yang isi data sebelumnya sudah ada. Daftar kolom – kolom tabel role dapat dilihat pada Tabel 3.14.

Tabel 3. 14 Tabel role

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| nama\_role | varchar(15) | Menyimpan nama role |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel hak\_akses\_dosen**

Tabel ini digunakan untuk menyimpan dan mengambil data hak akses dosen. Daftar kolom – kolom tabel hak\_akses\_dosen dapat dilihat pada Tabel 3.15.

Tabel 3. 15 Tabel hak\_akses\_dosen

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| role\_id | int(10) | Menyimpan nama role |
| dosen\_account\_id | int(10) | Menyimpan id dosen |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel mahasiswa\_account**

Tabel ini digunakan untuk menyimpan dan mengambil data akun mahasiswa. Daftar kolom – kolom tabel dosen\_mahasiswa dapat dilihat pada Tabel 3.16.

Tabel 3. 16 Tabel mahasiswa\_account

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| Nim | int(11) | Menyimpan username berdasarkan NIM |
| password | varchar(255) | Menyimpan password |
| status | boolean | Menyimpan status akun mahasiswa |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel proposal**

Tabel ini digunakan untuk menyimpan dan mengambil data pengajuan proposal tugas akhir yang isi data sebelumnya sudah di *upload* oleh mahasiswa.. Daftar kolom – kolom tabel proposal dapat dilihat pada Tabel 3.17.

Tabel 3. 17 Tabel proposal

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| mhs\_id | int(10) | Menyimpan id mahasiswa |
| dosen\_id | int(10) | Menyimpan id pembimbing |
| dosens\_id | int(10) | Menyimpan id pembimbing |
| email | varchar(100) | Menyimpan email |
| Judul | varchar(191) | Menyimpan judul tugas akhir |
| progress | varchar(255) | Menyimpan progress pengajuan proposal |
| Other | varchar(255) | Menyimpan keterangan pengajuan proposal |
| File | varchar(255) | Menyimpan file proposal |
| keterangan | varchar(255) | Menyimpan keterangan dari pembimbing |
| status | int(1) | Menyimpan status pengajuan tugas akhir |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel revisi\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data revisi tugas akhir beserta yang isi data sebelumnya sudah di *upload* oleh mahasiswa.. Daftar kolom – kolom tabel revisi\_ta dapat dilihat pada Tabel 3.18.

Tabel 3. 18 Tabel revisi\_ta

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| mhs\_id | int(10) | Menyimpan id mahasiswa |
| proposal\_id | int(10) | Menyimpan id proposal |
| File | varchar(255) | Menyimpan file revisi |
| keterangan | varchar(255) | Menyimpan keterangan revisi |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel sidang\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data pengajuan sidang tugas akhir yang isi data sebelumnya sudah di *upload* oleh mahasiswa. Daftar kolom – kolom tabel sidang\_ta dapat dilihat pada Tabel 3.19.

Tabel 3. 19 Tabel sidang\_ta

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| mhs\_id | int(10) | Menyimpan id mahasiswa |
| proposal\_id | int(10) | Menyimpan id proposal |
| bab1 | varchar(255) | Menyimpan file laporan tugas akhir bab 1 |
| bab2 | varchar(255) | Menyimpan file laporan tugas akhir bab 2 |
| bab3 | varchar(255) | Menyimpan file laporan tugas akhir bab 3 |
| bab4 | varchar(255) | Menyimpan file laporan tugas akhir bab 4 |
| bab5 | varchar(255) | Menyimpan file laporan tugas akhir bab 5 |
| file\_lain | varchar(255) | Menyimpan file persyaratan sidang |
| status | int(11) | Menyimpan persetujuan sidang dari pembimbing |
| keterangan | varchar(255) | Menyimpan keterangan mengenai adanya kekurangan kelengkapan persyaratan sidang |
| konfirmasi\_admin | varchar(1) | Menyimpan persetujuan sidang dari admin |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Tabel jadwal\_sidang**

Tabel ini digunakan untuk menyimpan dan mengambil data jadwal sidang tugas akhir yang isi data sebelumnya sudah dioleh coordinator TA. Daftar kolom – kolom tabel jadwal\_sidang dapat dilihat pada Tabel 3.20.

Tabel 3. 20 Tabel jadwal\_sidang

|  |  |  |
| --- | --- | --- |
| **Nama Kolom** | **Tipe Data** | **Keterangan** |
| Id | int(10) | Menyimpan id (PK) |
| mhs\_id | int(10) | Menyimpan id mahasiswa |
| proposal\_id | int(10) | Menyimpan id proposal |
| tgl\_sidang | date | Menyimpan tanggal sidang |
| penguji1 | int(10) | Menyimpan id dosen |
| penguji2 | int(10) | Menyimpan id dosen |
| waktu | varchar(255) | Menyimpan waktu mulai sidang |
| remember\_token | varchar(100) | Menyimpan token |
| created\_at | timestamp | Menyimpan waktu *insert* data |
| updated\_at | timestamp | Menyimpan waktu *update* data |

## **Perancangan Antar Muka**

Beberapa rancangan desain sistem atau antar muka (*mock up*) dari aplikasi sistem informasi tugas akhir berbasis web yang akan dijelaskan pada sub bab berikut.

## **Rancangan Halaman Awal**

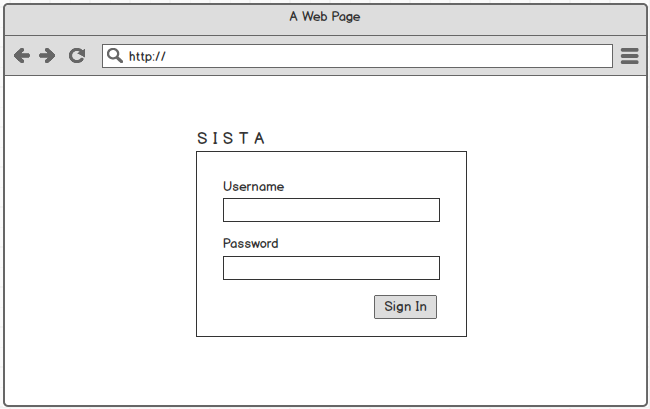
Halaman awal merupakan halaman yang pertama kali ditampilkan ketika *user* mengakses url dari aplikasi sistem informasi tugas akhir berbasis web. Dalam halaman ini pengguna memilih Adapun hasil rancangan halaman awal aplikasi dapat dilihat pada Gambar 3.21.



Gambar 3. 21 Rancangan Halaman Awal

## **Rancangan Halaman Login**

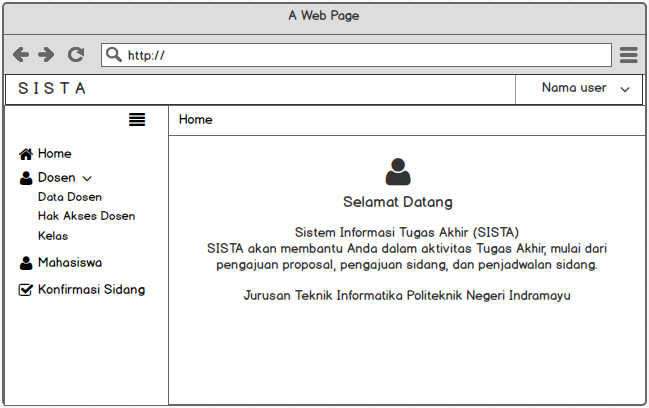
Halaman login merupakan halaman yang digunakan ketika *user* akan melakukan login ke dalam aplikasi sistem informasi tugas akhir berbasis web agar pengguna dapat mengakses fitur – fitur yang ada di dalam aplikasi. Adapun hasil rancangan halaman login aplikasi dapat dilihat pada Gambar 3.22.



Gambar 3. 22 Rancangan Halaman Login

## **Rancangan Halaman Home**

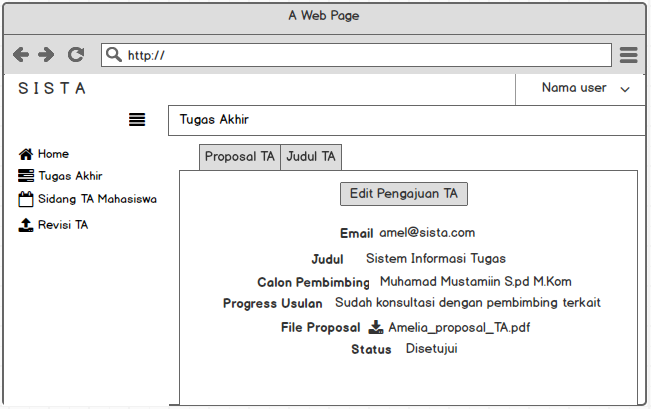
Halaman home merupakan halaman utama dari *user* setelah melakukan proses *login*. Adapun hasil rancangan halaman home aplikasi dapat dilihat pada Gambar 3.23.



Gambar 3. 23 Rancangan Halaman Home

## **Rancangan Halaman Tampil Data Pengajuan TA di User Mahasiswa**

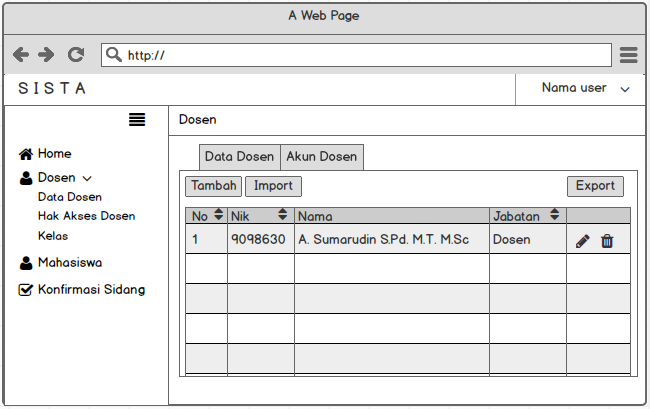
Halaman tampil data peengajuan TA merupakan suatu halaman yang menampilkan data pengajuan tugas akhir milik masing – masing mahasiswa yang telah di buat melalui proses input data. Adapun rancangan halaman tampil data pengajuan TA dapat dilihat pada Gambar 3.24.



Gambar 3. 24 Rancangan Halaman Tampil Data TA Akhir di User Mahasiswa

## **Rancangan Halaman Tampil Data**

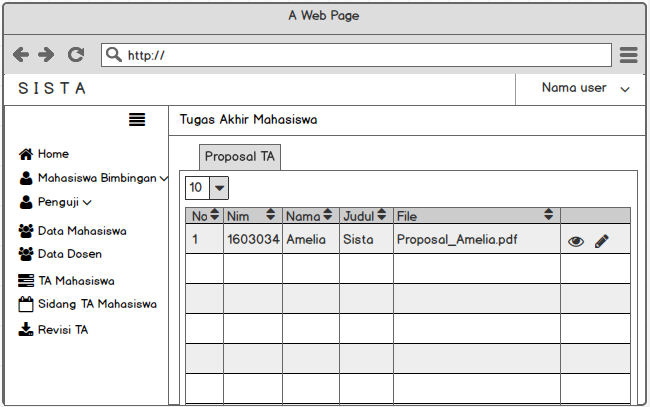
Halaman tampil data merupakan suatu halaman yang menampilkan data – data yang telah dibuat melalui proses input data. Adapun rancangan halaman tampil data dapat dilihat pada Gambar 3.25.



Gambar 3. 25 Rancangan Halaman Tampil Data

## **Rancangan Halaman Tampil Data Pengajuan Tugas Akhir Mahasiswa**

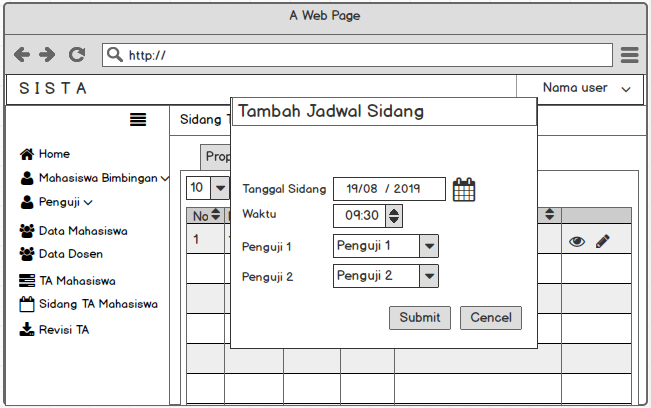
Halaman tampil data pengajuan tugas akhir merupakan suatu halaman yang menampilkan data tugas akhir yang telah di buat oleh mahasiswa melalui proses input data. Adapun rancangan halaman tampil data pengajuan tugas akhir dapat dilihat pada Gambar 3.26.



Gambar 3. 26 Rancangan Halaman Tampil Data Pengajuan Tugas Akhir Mahasiswa

## **Rancangan Halaman Tambah Data**

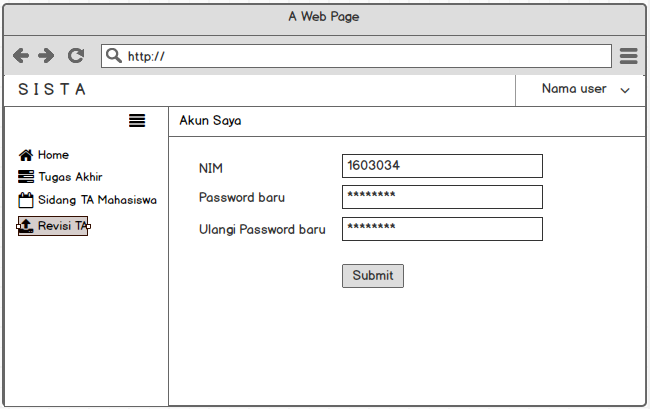
Halaman tambah data merupakan suatu halaman dimana user dapat menambahkan data untuk disimpan ke dalam *database*. Adapun rancangan halaman tambah data dapat dilihat pada Gambar 3.27.



Gambar 3. 27 Rancangan Halaman Tambah Data

## **Rancangan Halaman Akun**

Halaman profil merupakan suatu halaman dimana *user* dapat melakukan perubahan data akun. Adapun rancangan halaman tampil data pengajuan TA dapat dilihat pada Gambar 3.28.



Gambar 3. 28 Rancangan Halaman Profil

# **BAB IV**

# **HASIL DAN PEMBAHASAN**

## **Hasil**

Penelitian ini menghasilkan sebuah aplikasi sistem informasi tugas akhir. Aplikasi ini digunakan untuk mempermudah proses pelaksanaan tugas akhir khususnya proses administrasi tugas akhir. Aplikasi ini juga digunakan untuk memberi informasi mengenai tugas akhir kepada administrator, koordinator, dosen pembimbing, mahasiswa, wali dosen, dan ketua jurusan yang sudah terdaftar sebagai pengguna di sistem.

## **Pembahasan**

Setelah melakukan pengumpulan data, menganalisis kebutuhan sistem dan melakukan perancangan sistem dan perancangan antar muka, maka penulis melakukan implementasi terhadap semua yang telah disusun untuk terciptanya aplikasi sistem informasi tugas akhir.

## **Kebutuhan Sistem**

Dalam pembuatan aplikasi sistem informasi tugas akhir berbasis web menggunakan framework laravel ini terdapat kebutuhan baik dalam bentuk *hardware* maupun *software*. Adapun kebutuhan-kebutuhan tersebut dijelaskan pada sub bab selanjutnya.

**4.2.1.1 Kebutuhan *Hardware***

Adapun kebutuhan *hardware* untuk membuat sistem informasi tugas akhir berbasis *web* ini dapat dilihat pada Tabel 4.1.

Tabel 4. 1 Kebutuhan Hardware

|  |  |
| --- | --- |
| **Jenis *Harware*** | **Kebutuhan *Hardware*** |
| *Processor* | Intel Core i3 |
| RAM | 2 GB atau lebih |
| *Hardisk* | 180 atau lebih |

**4.2.1.2 Kebutuhan *Software***

Adapun kebutuhan *software* untuk membuat sistem informasi tugas akhir berbasis *web* ini dapat dilihat pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Software

|  |  |
| --- | --- |
| Jenis *Software* | Kebutuhan *Software* |
| Sistem Operasi | Windows 7 Profesional |
| Desain Aplikasi | Belsamiq Mockup |
| Bahasa Pemrograman | PHP, Framework Laravel |
| *Software* Pengolah | Atom Text Editor |
| Web Server | Apache |
| Web Browser | Google Chrome, Mozilla Firefox, Microsoft Edge |

## **Implementasi**

Implementasi merupakan tahapan setelah melakukan pengumpulan data, menganalisis kebutuhan sistem dan melakukan perancangan sistem dan perancangan antar muka yang dilakukan untuk terciptanya aplikasi sistem informasi tugas akhir berbasis web.

## **4.2.1.1 Implementasi Database**

*Database* sista merupakan nama *database* yang dibuat di *server hosting* yang penulis gunakan dalam pembuaatan aplikasi sistem informasi tugas akhir. *Database* sista merupakan sebuah tempat untuk menyimpan data berupa tabel – tabel *database* yang telah dibuat.

*Database* dalam pembuatan aplikasi sistem informasi tugas akhir berbasis web ini terdiri dari 13 tabel yaitu tabel admin\_account, tabel data\_dosen, tabel data\_mahasiswa, tabel dosen\_account, tabel judul\_ta, tabel kelas, tabel role, tabel hak\_akses\_dosen, tabel mahasiswa\_account, tabel proposal, tabel revisi\_ta, tabel sidang\_ta, tabel jadwal\_sidang.

1. **Tabel admin\_account**

Tabel ini digunakan untuk menyimpan dan mengambil data akun administrator yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel admin\_account dapat dilihat pada Tabel 4.3.

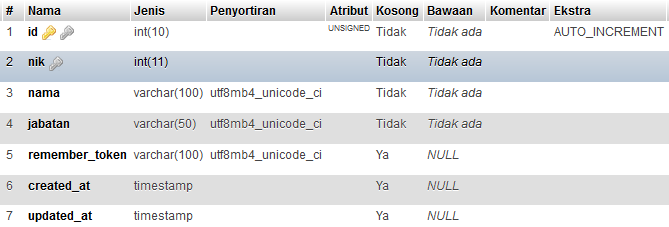
Tabel 4. 3 admin\_account



1. **Tabel data\_dosen**

Tabel ini digunakan untuk menyimpan dan mengambil data dosen yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel data\_dosen dapat dilihat pada Tabel 4.4.

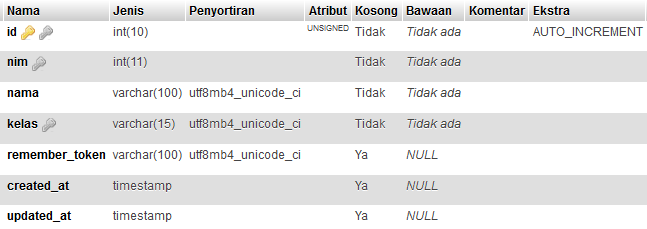
Tabel 4. 4 data\_dosen



1. **Tabel data\_mahasiswa**

Tabel ini digunakan untuk menyimpan dan mengambil data mahasiswa yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel data\_mahasiswa dapat dilihat pada Tabel 4.5.

Tabel 4. 5 data\_mahasiswa



1. **Tabel dosen\_account**

Tabel ini digunakan untuk menyimpan dan mengambil data akun dosen yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel dosen\_account dapat dilihat pada Tabel 4.6.

Tabel 4. 6 dosen\_account



1. **Tabel judul\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data judul tugas akhir yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel judul\_ta dapat dilihat pada Tabel 4.7.

Tabel 4. 7 judul\_ta



1. **Tabel kelas**

Tabel ini digunakan untuk menyimpan dan mengambil data kelas yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel kelas dapat dilihat pada Tabel 4.8.

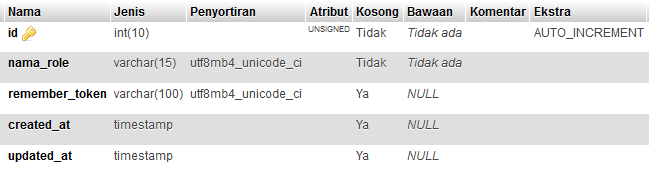
Tabel 4. 8 kelas



1. **Tabel role**

Tabel ini digunakan untuk menyimpan dan mengambil data role yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel role dapat dilihat pada Tabel 4.9.

Tabel 4. 9 role



1. **Tabel hak\_akses\_dosen**

Tabel ini digunakan untuk menyimpan dan mengambil data hak akses dosen yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel hak\_akses\_dosen dapat dilihat pada Tabel 4.10.

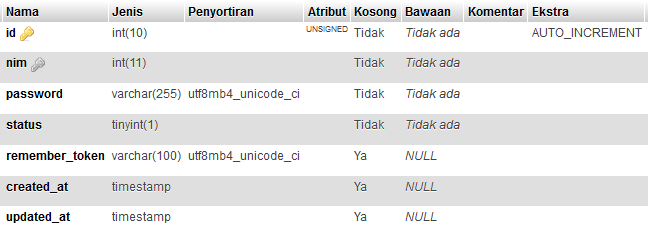
Tabel 4. 10 hak\_akses\_dosen



1. **Tabel mahasiswa\_account**

Tabel ini digunakan untuk menyimpan dan mengambil data akun mahasiswa yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel mahasiswa\_account dapat dilihat pada Tabel 4.11.

Tabel 4. 11 mahasiswa\_account



1. **Tabel proposal**

Tabel ini digunakan untuk menyimpan dan mengambil data proposal tugas akhir yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel proposal dapat dilihat pada Tabel 4.12.

Tabel 4. 12 proposal



1. **Tabel revisi\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data revisi tugas akhir yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel revisi\_ta dapat dilihat pada Tabel 4.13.

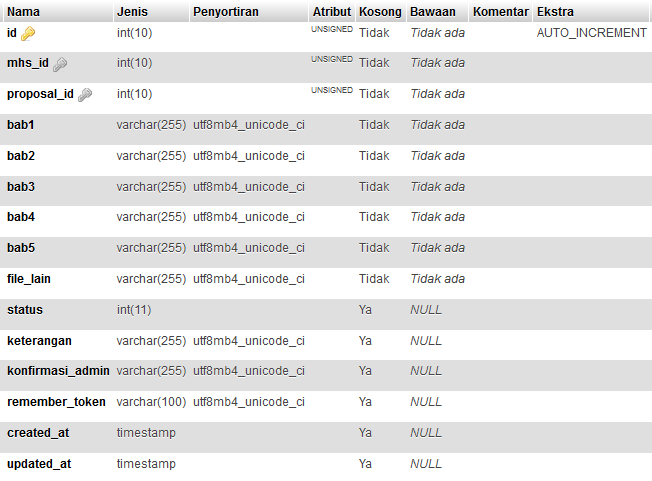
Tabel 4. 13 revisi\_ta



1. **Tabel sidang\_ta**

Tabel ini digunakan untuk menyimpan dan mengambil data sidang tugas akhir yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel sidang\_ta dapat dilihat pada Tabel 4.14.

Tabel 4. 14 sidang\_ta



1. **Tabel jadwal\_sidang**

Tabel ini digunakan untuk menyimpan dan mengambil data jadwal sidang tugas akhir yang isi data sebelumnya sudah ada. Adapun hasil dari pembuatan tabel jadwal\_sidang dapat dilihat pada Tabel 4.15.

Tabel 4. 15 jadwal\_sidang

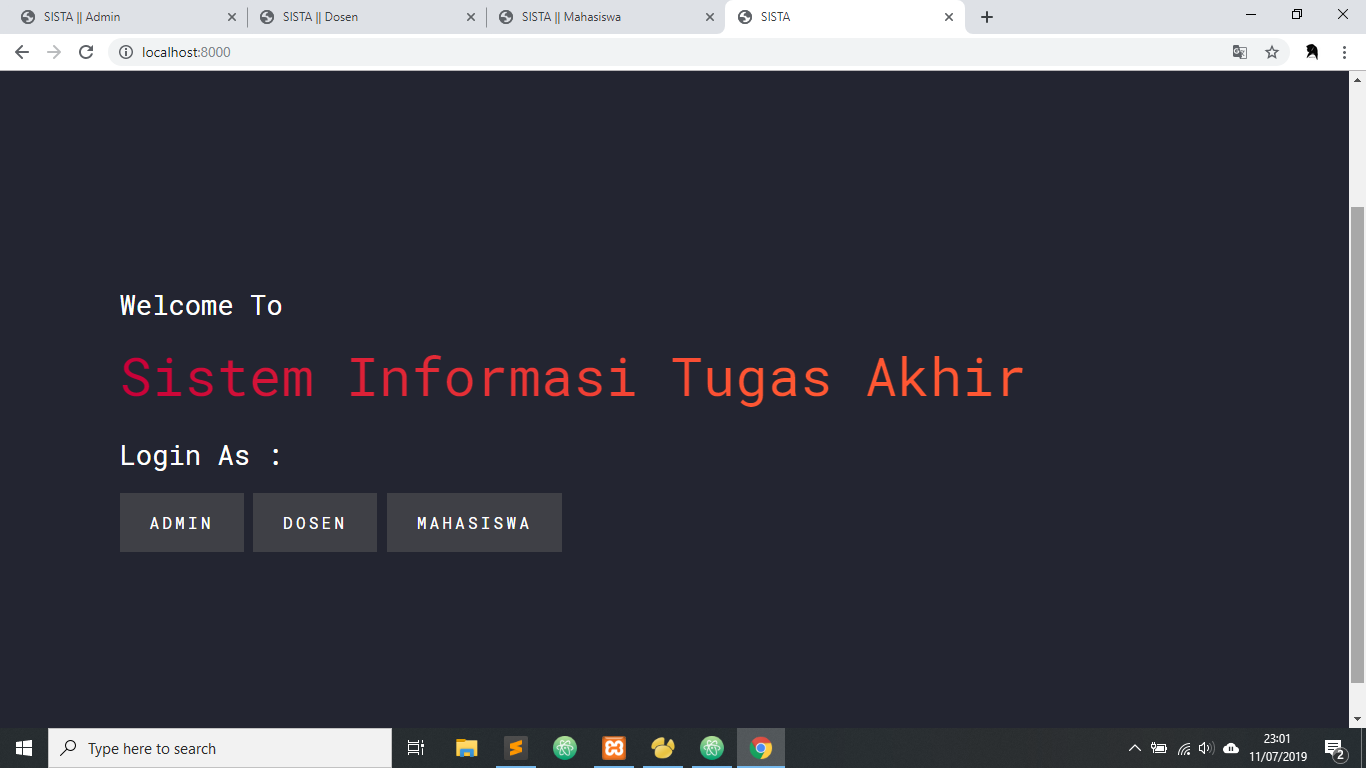


## **Implementasi Interface**

Implementasi desain *interface* dan program merupakan tahapan untuk mengimplementasikan desain yang telah dirancang pada bab sebelumnya. Setiap desain halaman dari aplikasi dibuat menggunakan *software* Atom Text Editor. Untuk kode program sendiri dapat dilihat pada lampiran. Berikut beberapa tampilan interface dari aplikasi system informasi tugas akhir (SISTA) berbasis web menggunakan framework laravel.

1. **Tampilan Awal**

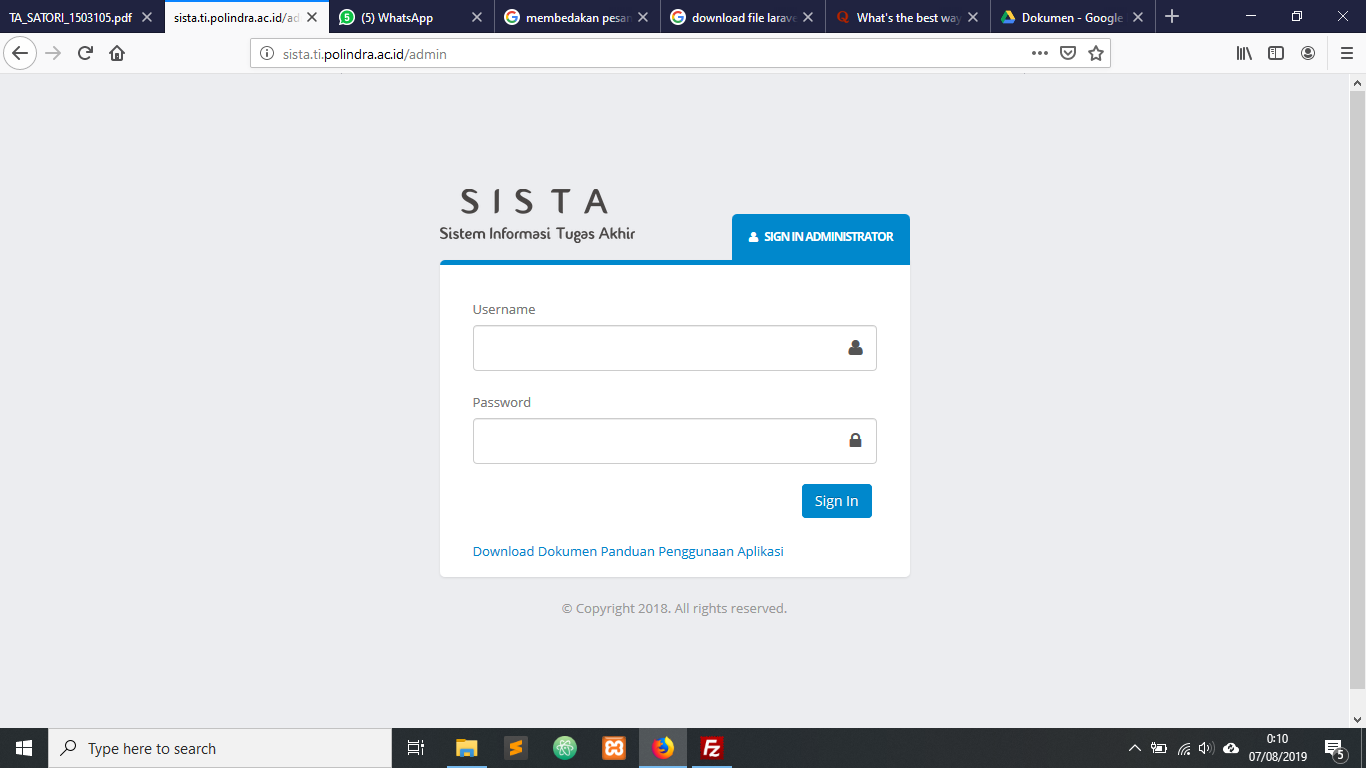
Halaman awal merupakan halaman yang pertama kali tampil ketika *user* mengakses URL website SISTA. Adapun hasil tampilan halaman awal dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Tampilan Halaman Awal

1. **Tampilan Login**

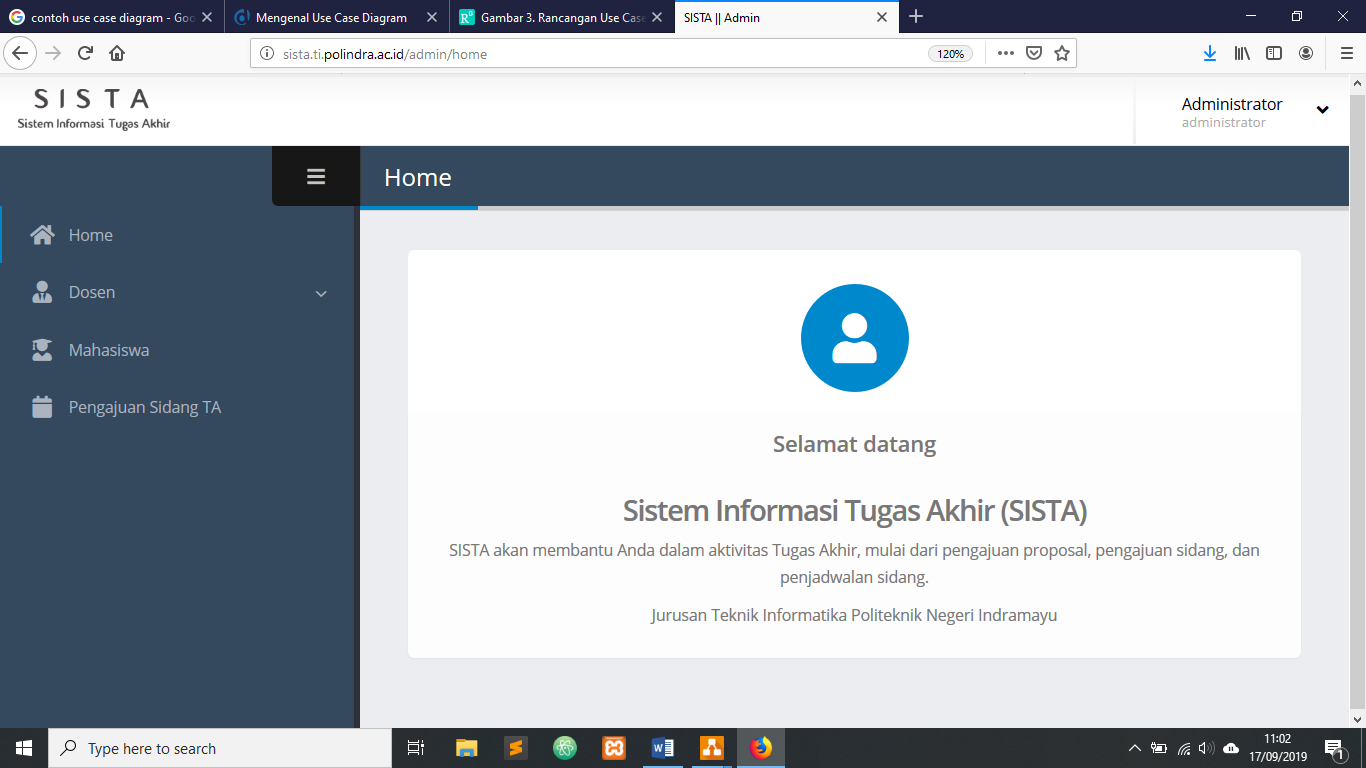
Halaman login merupakan halaman yang digunakan *user* untuk *login* ke dalam aplikasi agar dapat mengakses fitur – fitur di dalam aplikasi. Adapun hasil tampilan halaman login dapat dilihat pada Gambar 4.2.



Gambar 4. 2 Tampilan Halaman Login

1. **Tampilan Home**

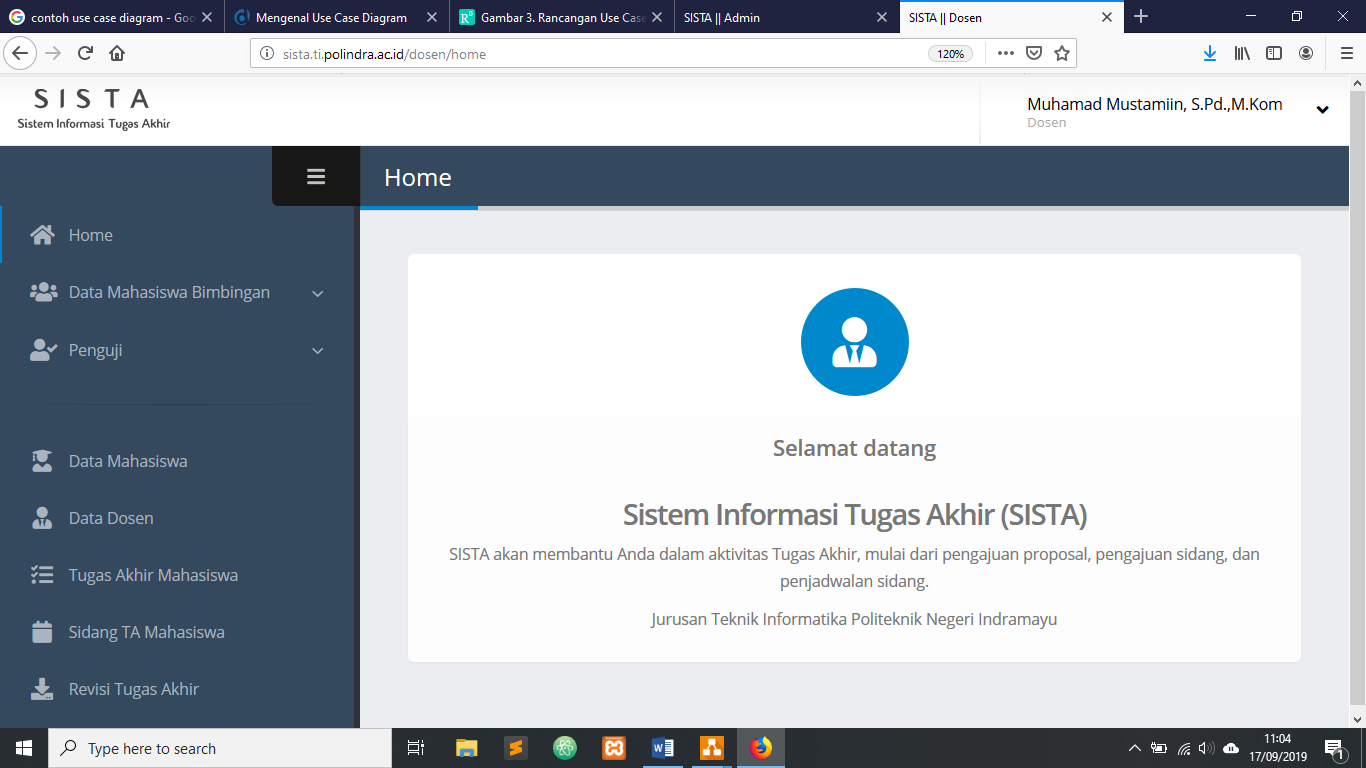
Halaman home adalah halaman yang pertama kali dimuat ketika *user* administrator, *user* dosen, dan *user* mahasiswa berhasil melakukan *login*. Adapun hasil tampilan halaman *home*  beserta penjelasan dari setiap menu dapat dilihat pada gambar dibawah ini.



Gambar 4. 3 Tampilan Halaman Home Admin

Adapun penjelasan dari Gambar 4.3 yaitu :

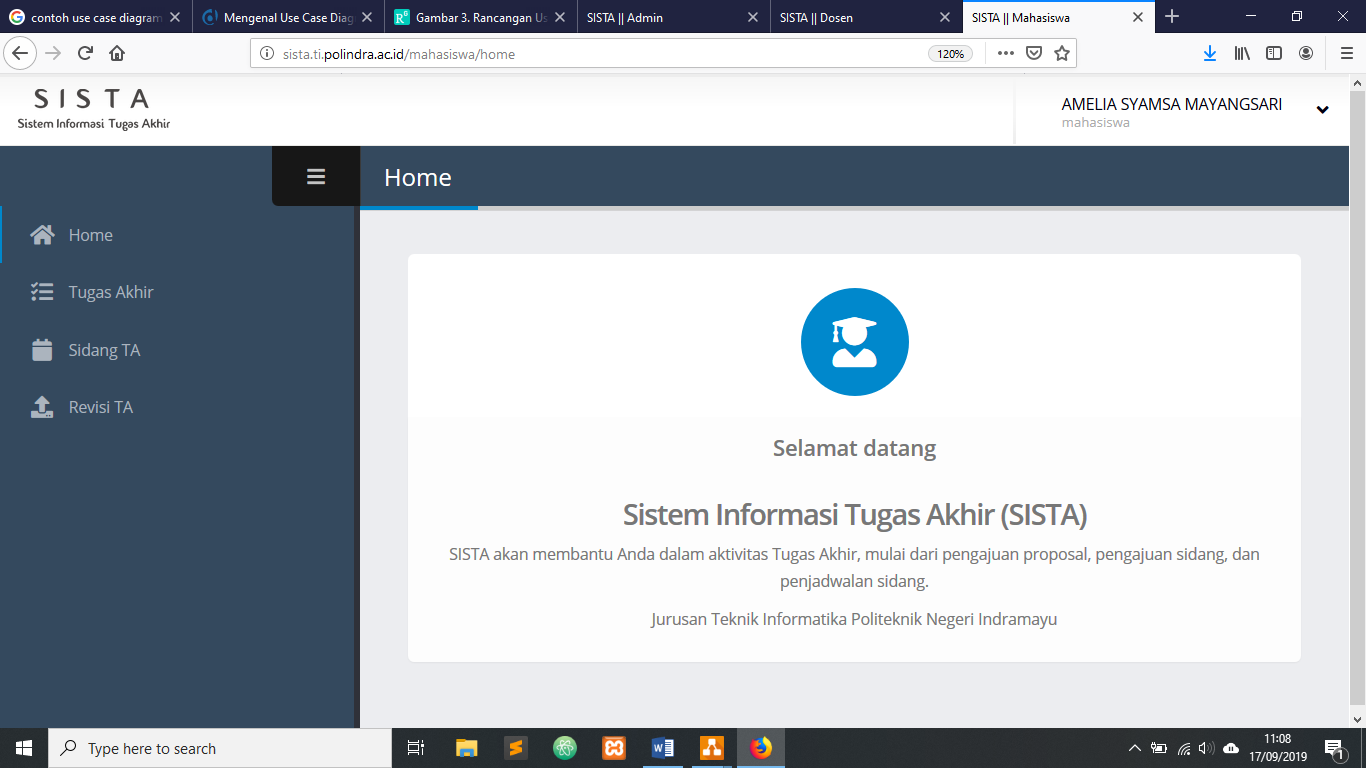
1. Menu Dosen berfungsi untuk mengelola data dosen, akun dosen, hak akses dosen, dan data kelas.
2. Menu Mahasiswa berfungsi untuk mengelola data mahasiswa dan akun mahasiswa.
3. Menu Pengajuan Sidang TA berfungsi untuk melakukan konfirmasi pengajuan sidang TA dari mahasiswa.



Gambar 4. 4 Tampilan Halaman Home Dosen

Adapun penjelasan dari Gambar 4.4 yaitu :

1. Menu Data Mahasiswa Bimbingan digunakan untuk mengelola data terkait tugas akhir dari mahasiswa bimbingannya.
2. Menu Jadwal Pengujian digunakan untuk melihat jadwal sidang dimana dosen yang sedang *login* sebagai pengujinya.
3. Menu Data Mahasiswa digunakan untuk melihat data mahasiswa yang sudah diinputkan oleh administrator.
4. Menu Data Dosen digunakan untuk melihat data mahasiswa yang sudah diinputkan oleh administrator.
5. Menu Tugas Akhir Mahasiswa digunakan untuk mengelola data pengajuan TA dari seluruh mahasiswa.
6. Menu Sidang TA Mahasiswa digunakan untuk mengelola data pengajuan sidang TA dari seluruh mahasiswa.
7. Menu Revisi digunakan untuk mengunduh file laporan tugas akhir mahasiswa.

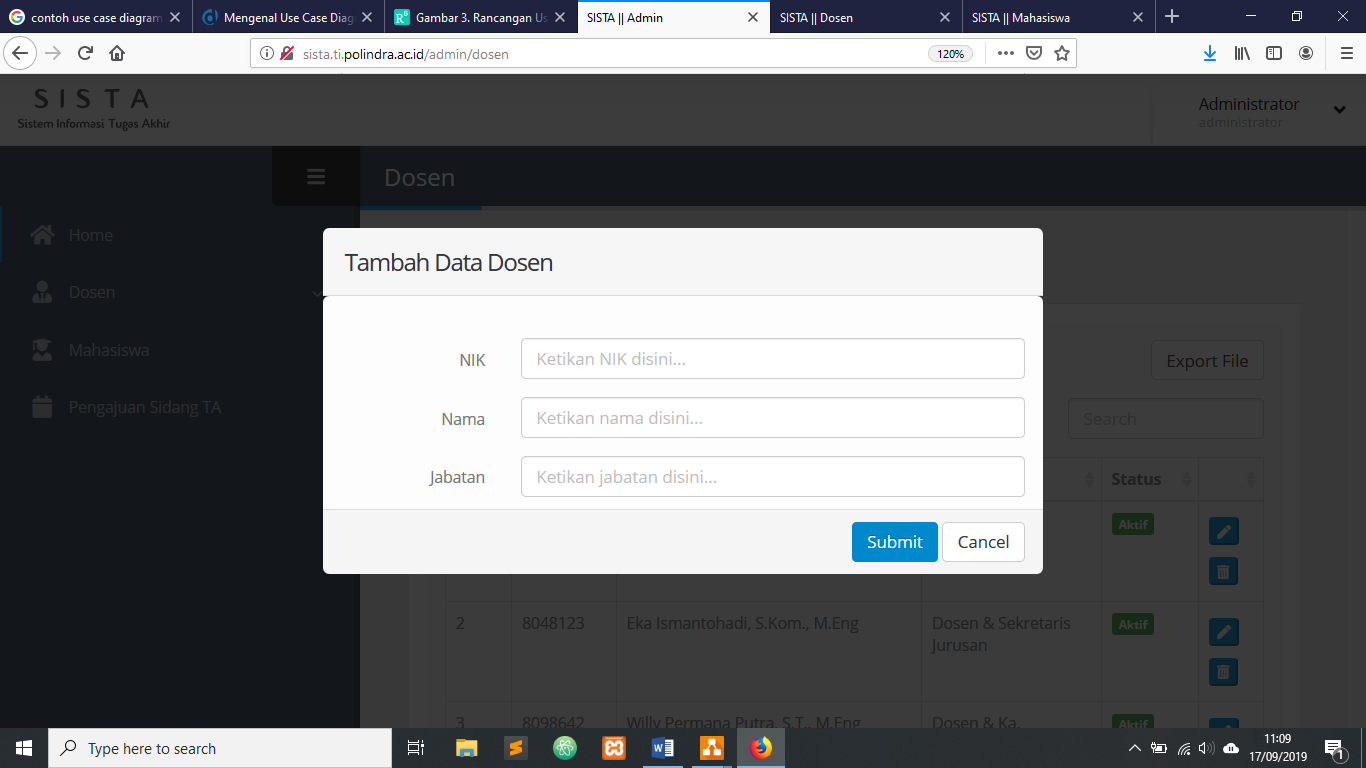


Gambar 4. 5 Tampilan Halaman Home Mahasiswa

Adapun penjelasan dari Gambar 4.5 yaitu :

1. Menu Tugas Akhir digunakan untuk mengelola pengajuan tugas akhir.
2. Menu Sidang TA digunakan untuk mengelola pengajuan sidang tugas akhir.
3. Menu Revisi TA digunakan untuk mengunggah file laporan tugas akhir.
4. **Tampilan Tambah Data Dosen**

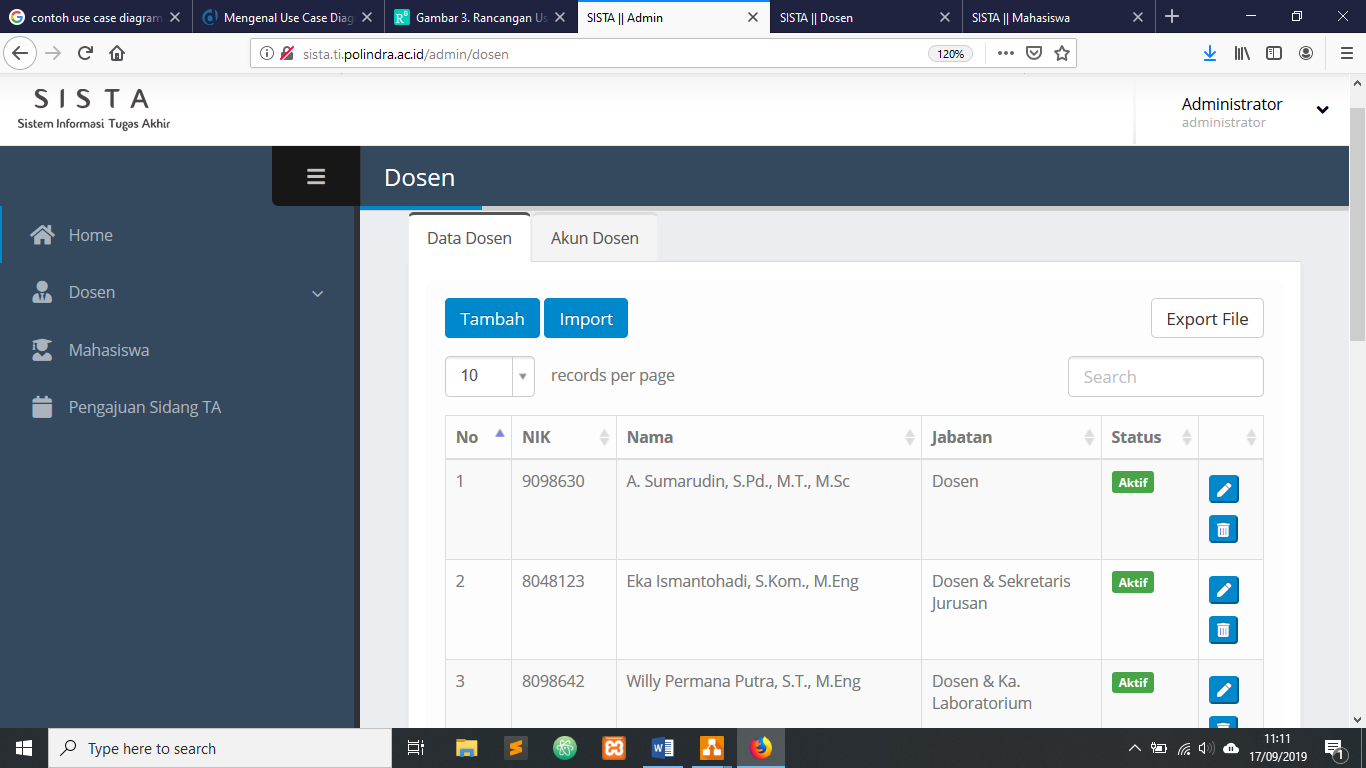
Halaman tambah data dosen merupakan sebuah halaman dimana administrator dapat menambahkan data dosen sekaligus membuat akun dosen. Adapun hasil tampilan halaman tambah data dapat dilihat pada Gambar 4.6.



Gambar 4. 6 Tampilan Halaman Tambah Data Dosen

1. **Tampilan List Data Dosen**

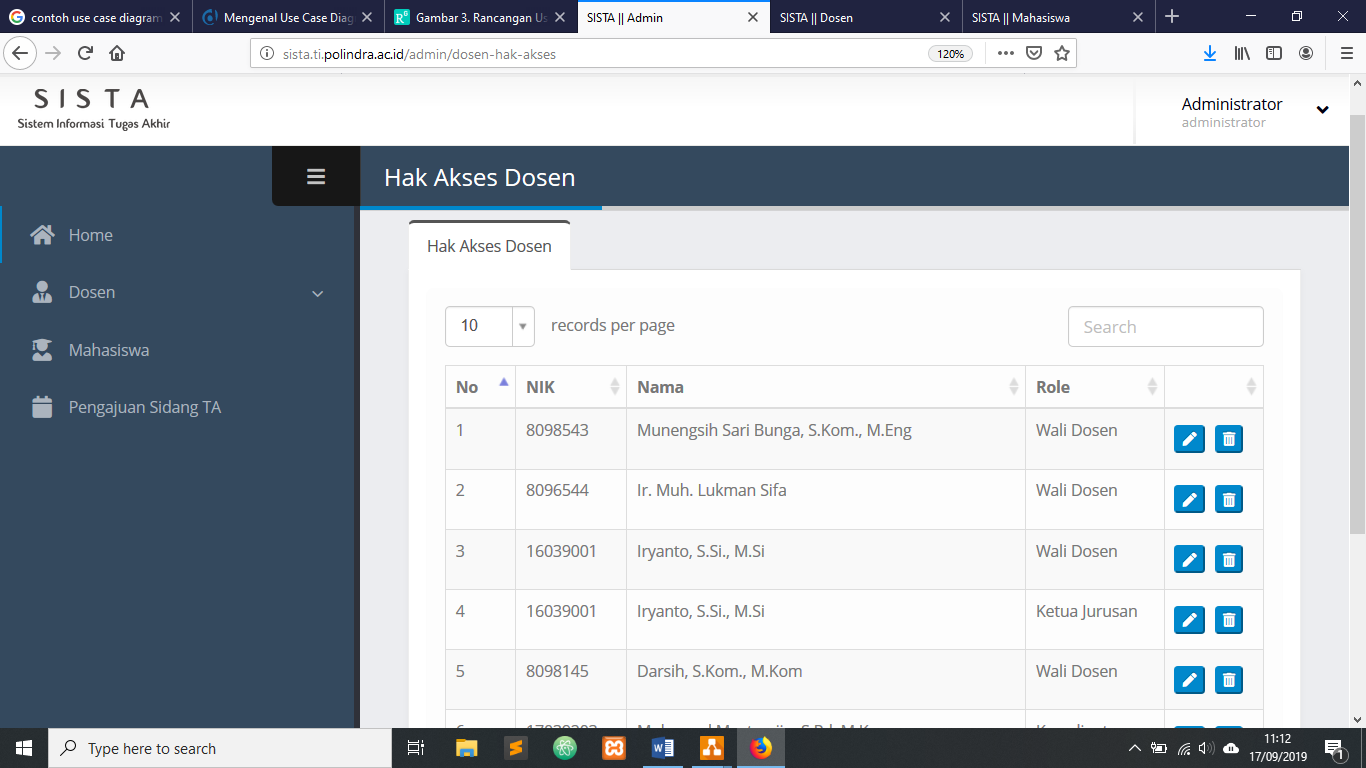
Halaman list data dosen adalah suatu halaman dimana administrator dapat melihat data dosen yang telah dibuat sebelumnya. Dalam halaman ini juga administrator dapat mengelola data dosen seperti mengaktifkan atau menonaktifkan akun dosen dan menghapus data dosen. Adapun hasil tampilan halaman list data dosen dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Tampilan List Data Dosen

1. **Tampilan List Hak Akses Dosen**

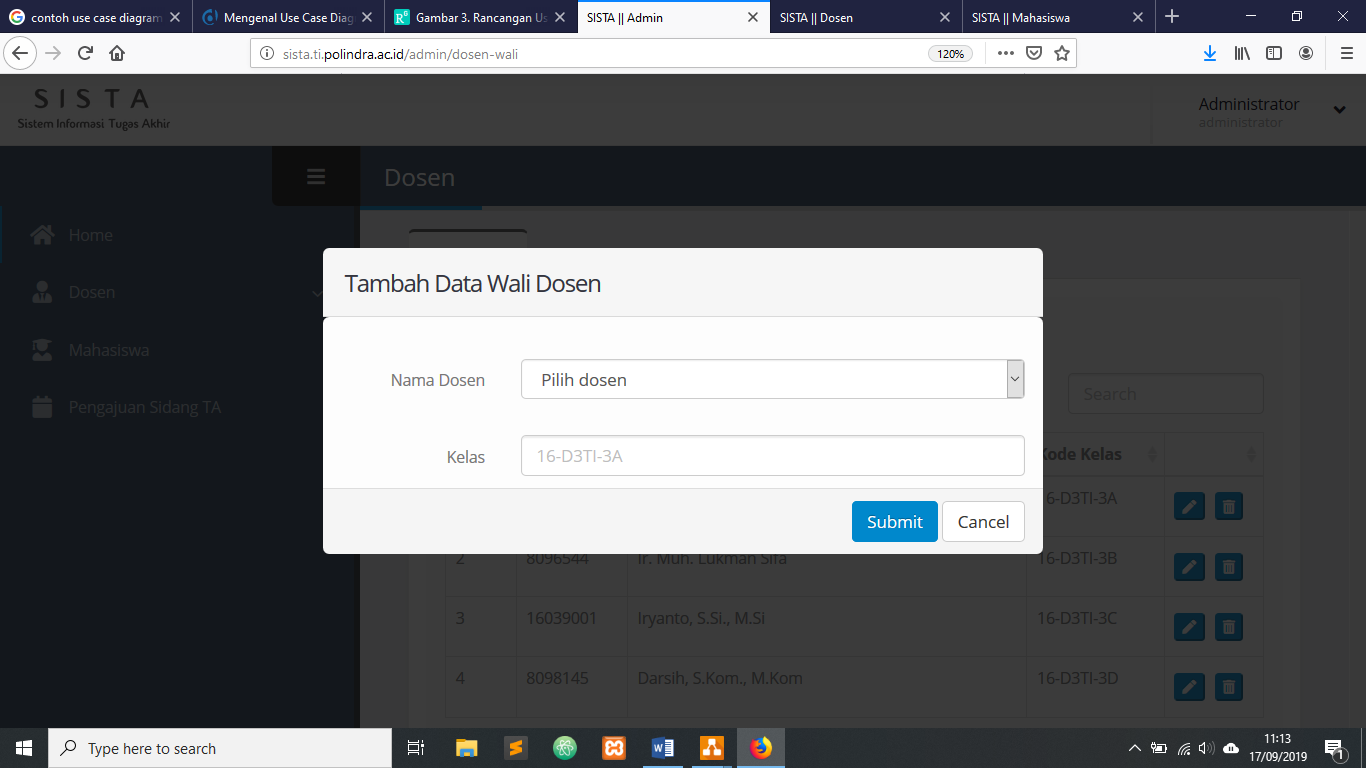
Halaman list data dosen adalah suatu halaman dimana administrator dapat melihat hak akses dosen yang telah dibuat sebelumnya. Dalam halaman ini juga administrator dapat mengelola hak akses dosen yaitu menghapus hak akses dosen. Adapun hasil tampilan halaman list hak akses dosen dapat dilihat pada Gambar 4.8.



Gambar 4.8 Tampilan List Hak Akses Dosen

1. **Tampilan Tambah Kelas**

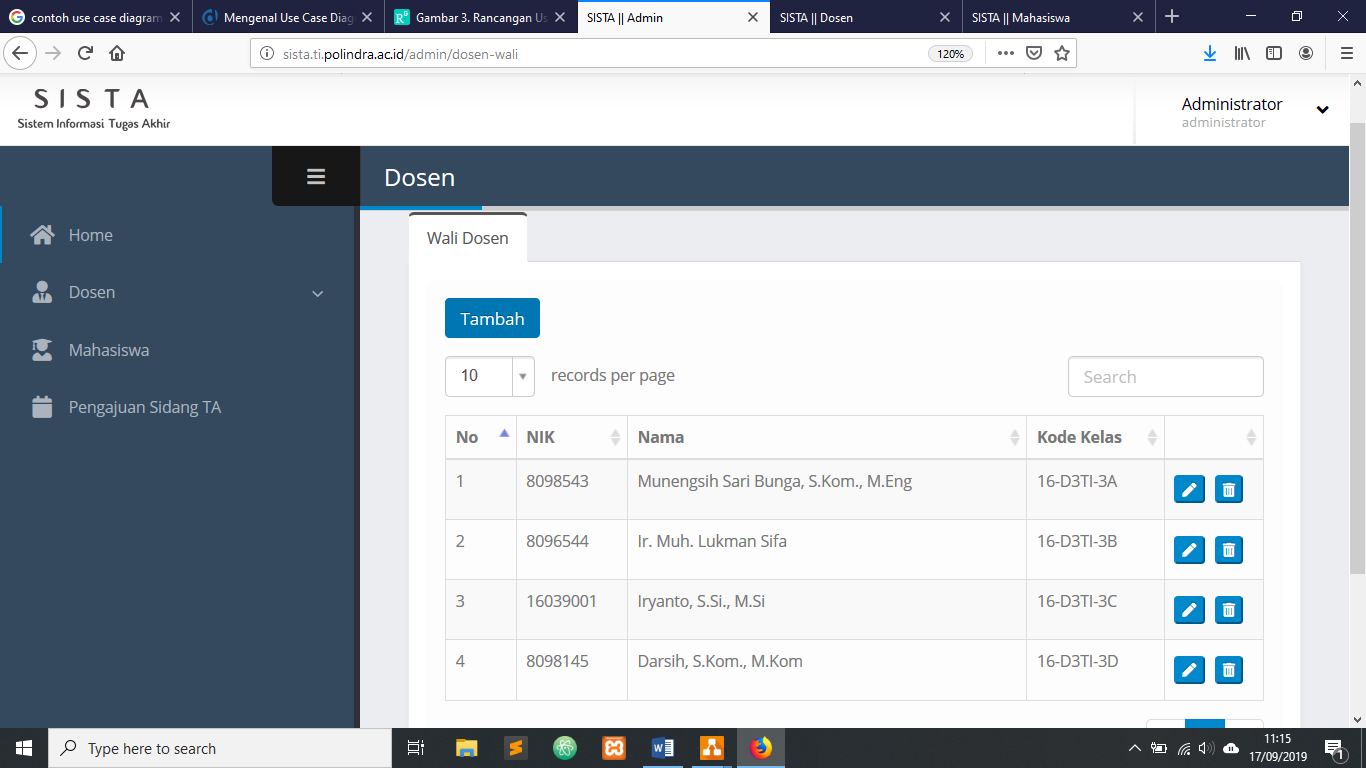
Halaman tambah kelas merupakan sebuah halaman dimana administrator dapat menambahkan data kelas sekaligus menentukan dosen wali dari kelas tersebut. Adapun hasil tampilan halaman tambah kelas dapat dilihat pada Gambar 4.9.



Gambar 4.9 Tampilan Halaman Tambah Kelas

1. **Tampilan List Kelas**

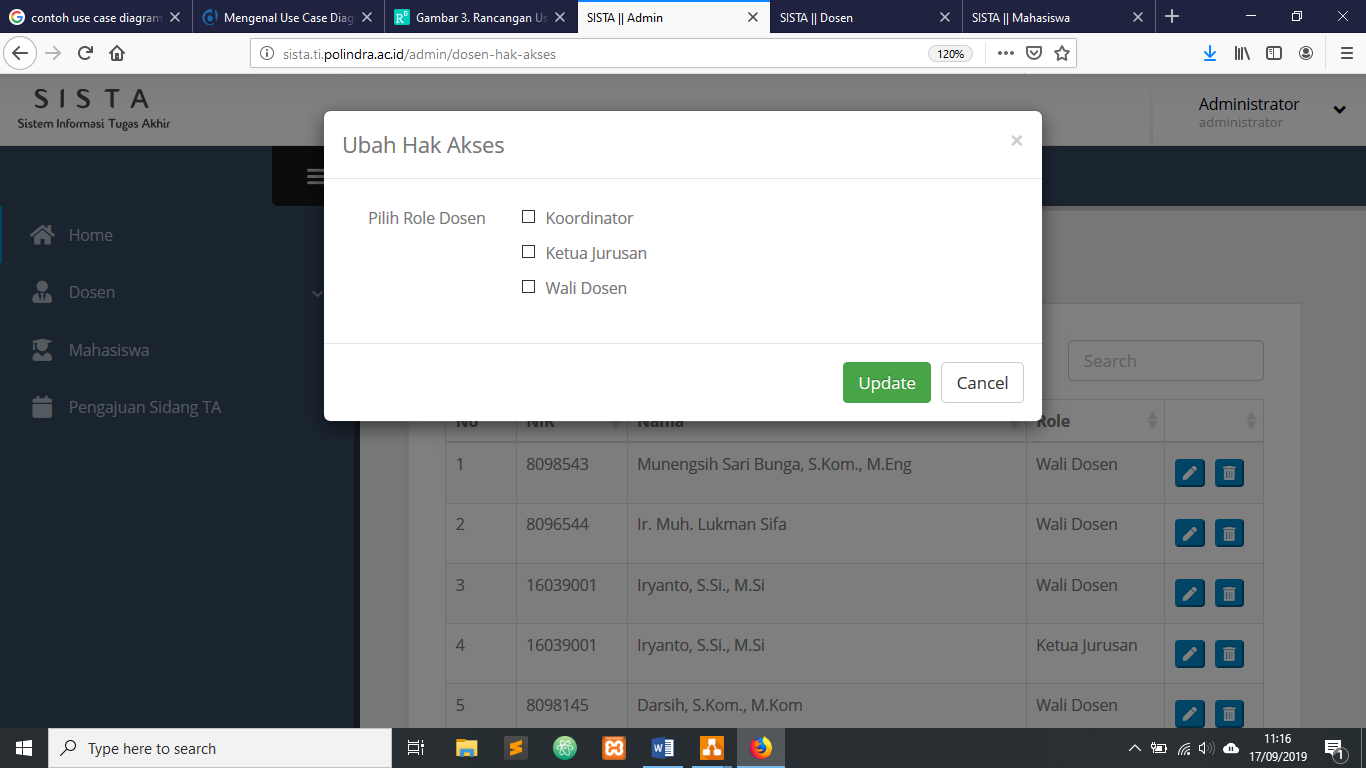
Halaman list kelas adalah suatu halaman dimana administrator dapat melihat data kelas serta wali dosen dari kelas tersebut yang telah dibuat sebelumnya. Dalam halaman ini juga administrator dapat mengelola data kelas yaitu menghapus kelas. Adapun hasil tampilan halaman list kelas dapat dilihat pada Gambar 4.10.



Gambar 4. 10 Tampilan List Kelas

1. **Tampilan Tambah Hak Akses Dosen**

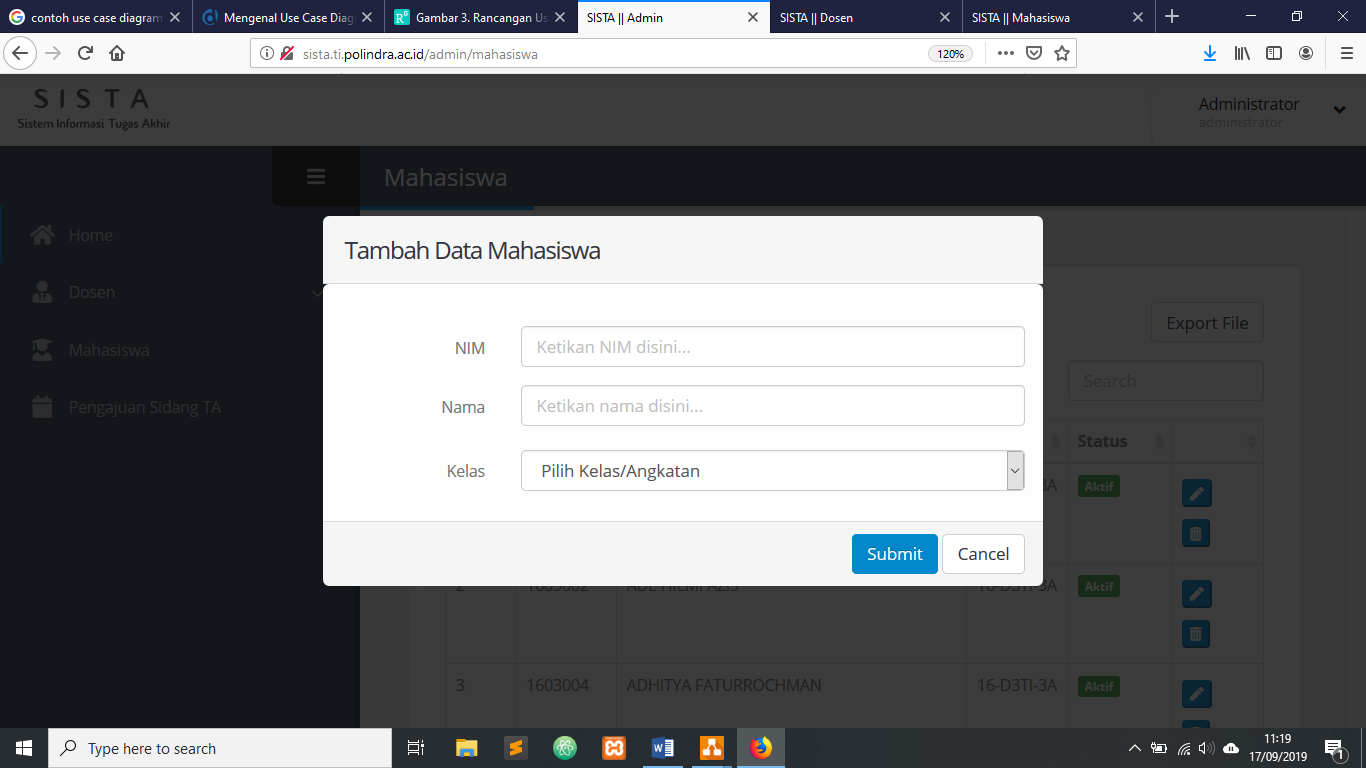
Halaman tambah hak akses merupakan sebuah halaman dimana administrator dapat menambahkan data hak akses dosen. Adapun hasil tampilan halaman tambah hak akses dapat dilihat pada Gambar 4.11.



Gambar 4. 11 Tampilan Tambah Hak Akses Dosen

1. **Tampilan Tambah Data Mahasiswa**

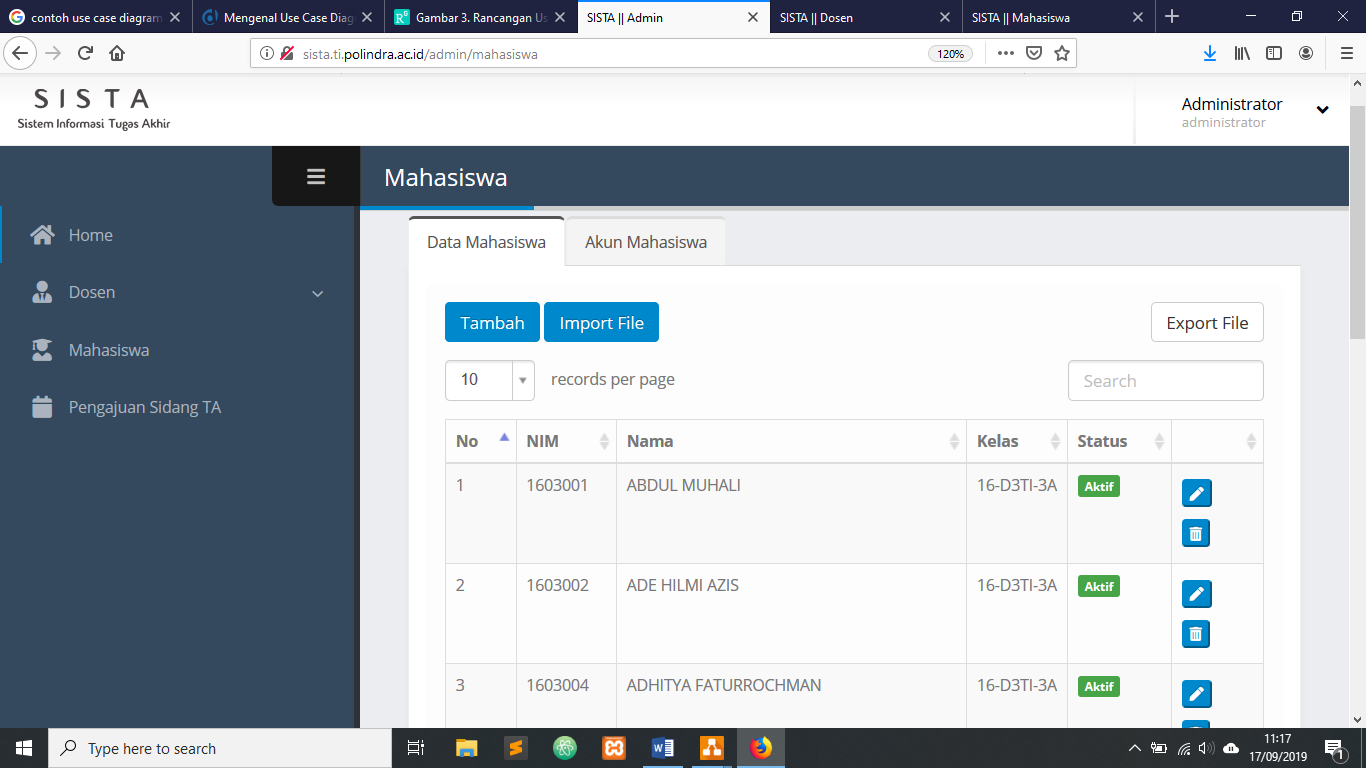
Halaman tambah data mahasiswa merupakan sebuah halaman dimana administrator dapat menambahkan data mahasiswa sekaligus membuat akun mahasiswa. Adapun hasil tampilan halaman tambah data mahasiswa dapat dilihat pada Gambar 4.12.



Gambar 4. 12 Tampilan Tambah Data Mahasiswa

1. **Tampilan List Data Mahasiswa**

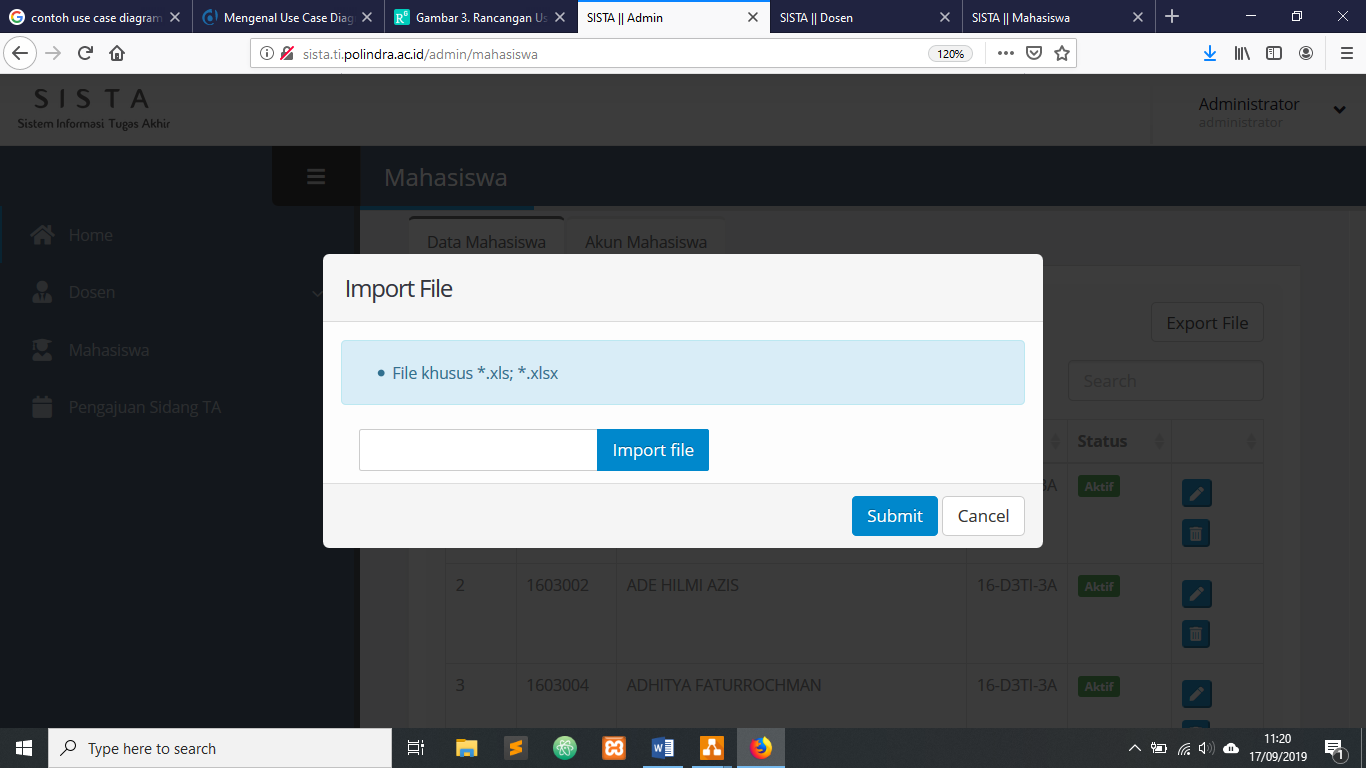
Halaman list data mahasiswa adalah suatu halaman dimana administrator dapat melihat data mahasiswa yang telah dibuat sebelumnya. Dalam halaman ini juga administrator dapat mengelola data mahasiswa seperti mengaktifkan atau menonaktifkan akun mahasiswa dan menghapus data mahasiswa. Adapun hasil tampilan halaman list data mahasiswa dapat dilihat pada Gambar 4.13.



Gambar 4. 13 Tampilan List Data Mahasiswa

1. **Tampilan Import Data**

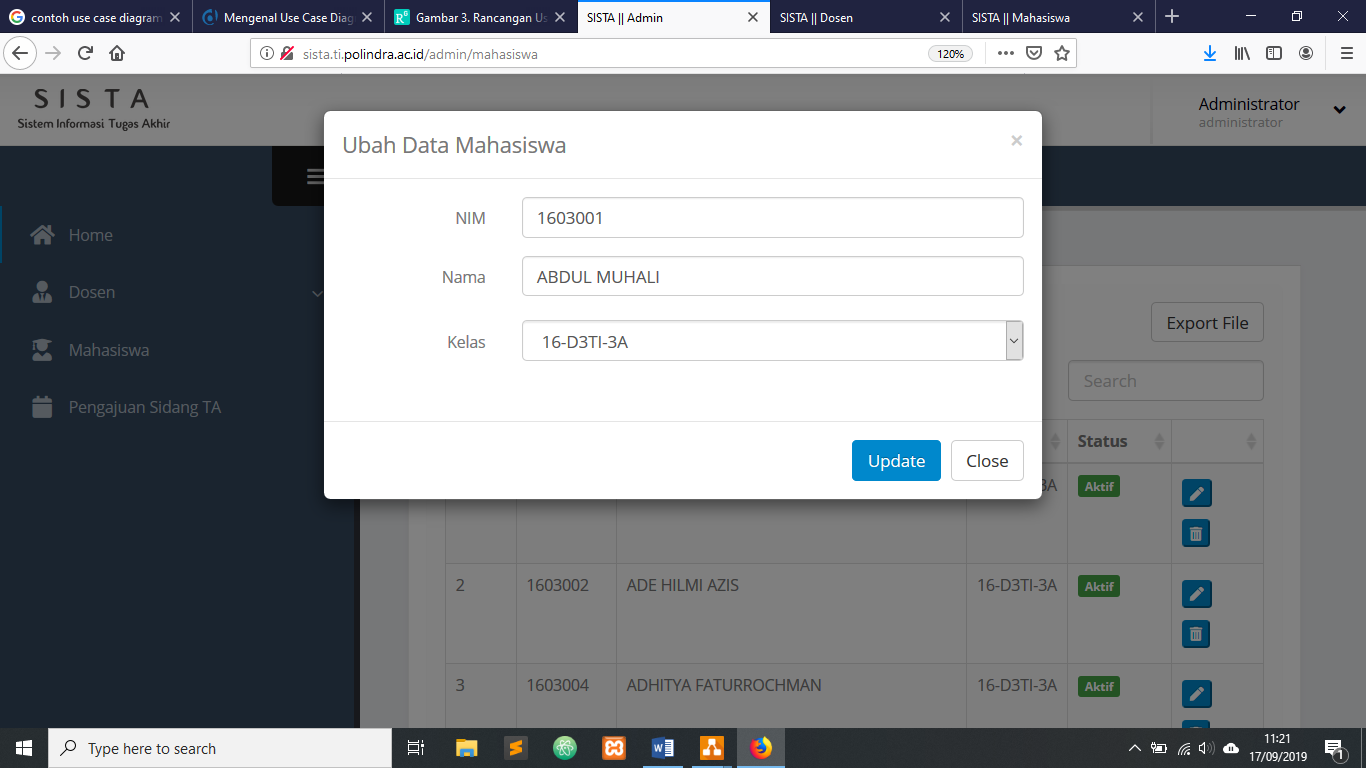
Halaman import data adalah suatu halaman dimana admin dapat menambahkan data dosen ataupun mahasiswa melalui file excel yang telah ditentukan format penulisannya. Adapun hasil tampilan halaman import data dapat dilihat pada Gambar 4.14.



Gambar 4. 14 Tampilan Import Data

1. **Tampilan Ubah Data**

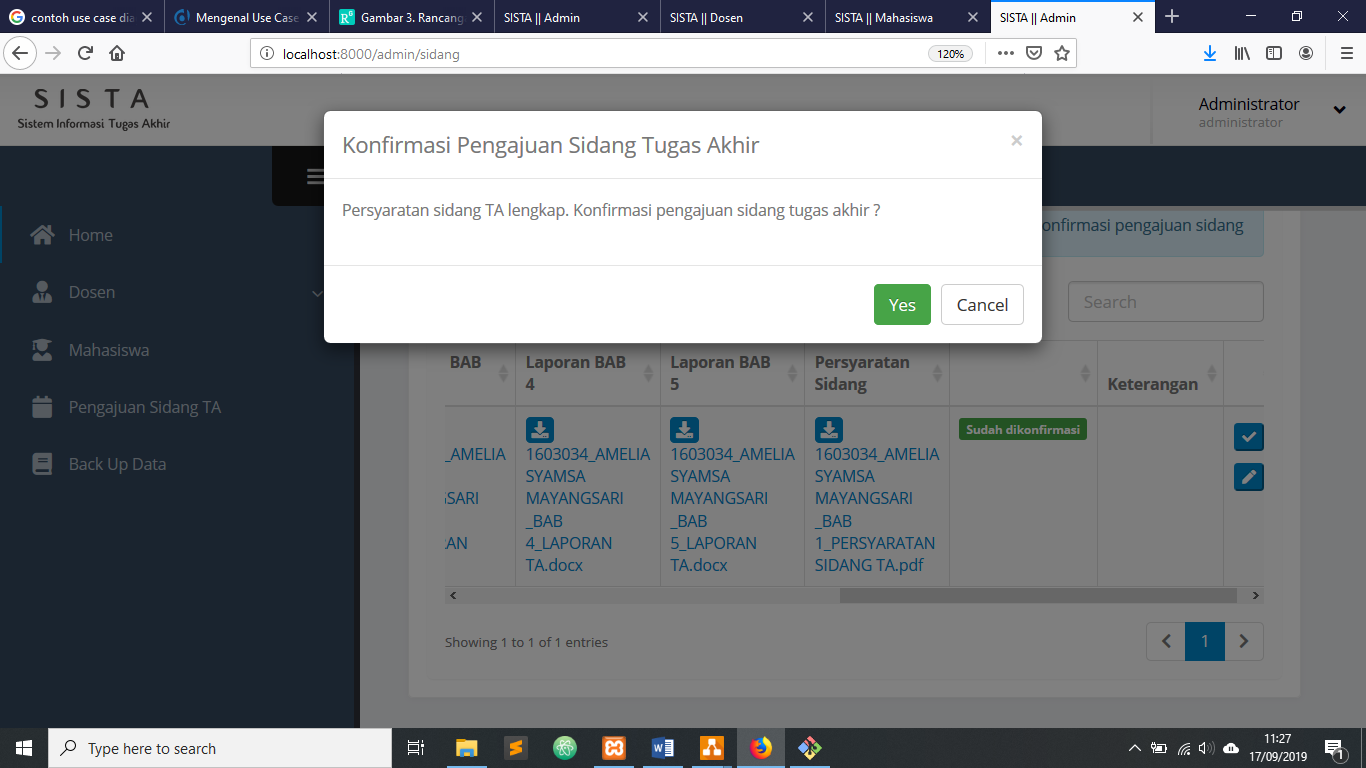
Halaman ubah data merupakan suatu halamn dimana admin dapat melakukan perubahan data yang telah dibuat sebelumnya. Adapun hasil tampilan halaman ubah data dapat dilihat pada Gambar 4.15.



Gambar 4. 15 Tampilan Ubah Data

1. **Tampilan Konfirmasi Sidang TA**

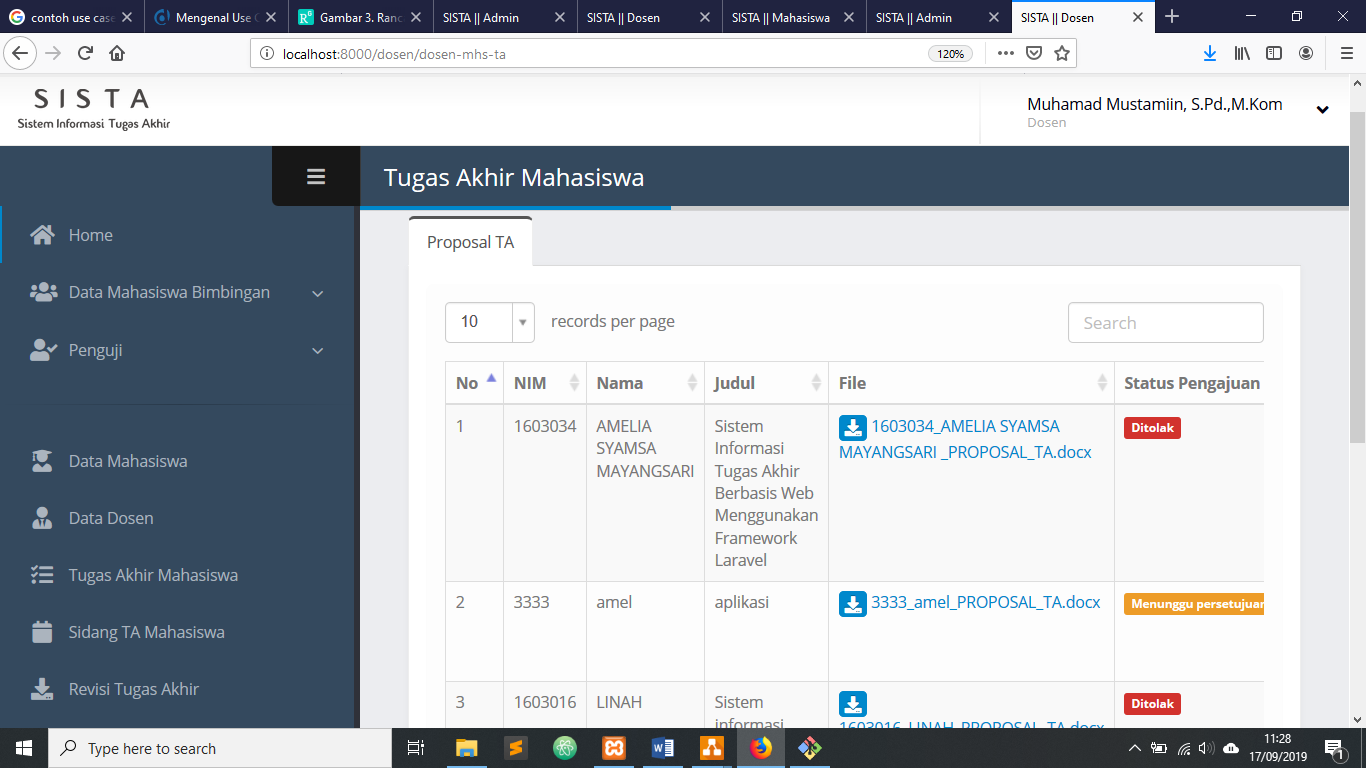
Halaman konfirmasi sidang TA merupakan halaman dimana admin perlu mengkonfirmasi pengajuan sidang TA oleh mahasiswa yang sudah disetujui oleh kedua pembimbing. Adapun hasil tampilan konfirmasi sidang TA dapat dilihat pada Gambar 4.16.



Gambar 4. 16 Tampilan Konfirmasi Sidang

1. **Tampilan List Pengajuan TA**

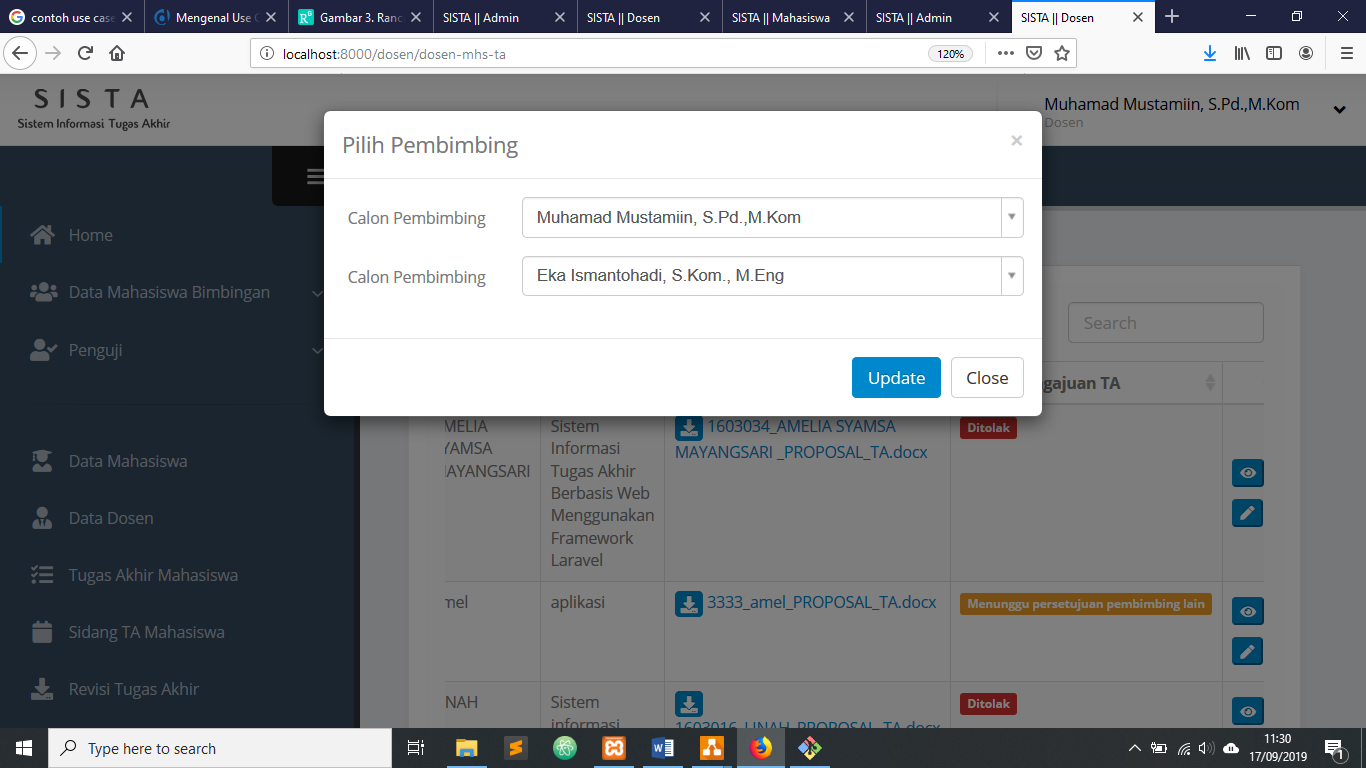
Halaman list pengajuan TA adalah suatu halaman dimana ketua jurusan dan koordinator TA dapat melihat data pengajuan TA seluruh dari mahasiswa, dosen pembimbing dapat melihat data pengajuan TA dari mahasiswa bimbingannya, wali dosen dapat melihat data pengajuan TA dari mahasiswa kelasnya. Selain itu ketua jurusan, koordinator TA, dosen pembimbing, dan wali dosen dapat mengunduh file proposal TA dan melihat detail pengajuan TA. Adapun hasil tampilan halaman list pengajuan TA dapat dilihat pada Gambar 4.17.



Gambar 4. 17 Tampilan List Pengajuan TA

1. **Tampilan Pembagian Pembimbing**

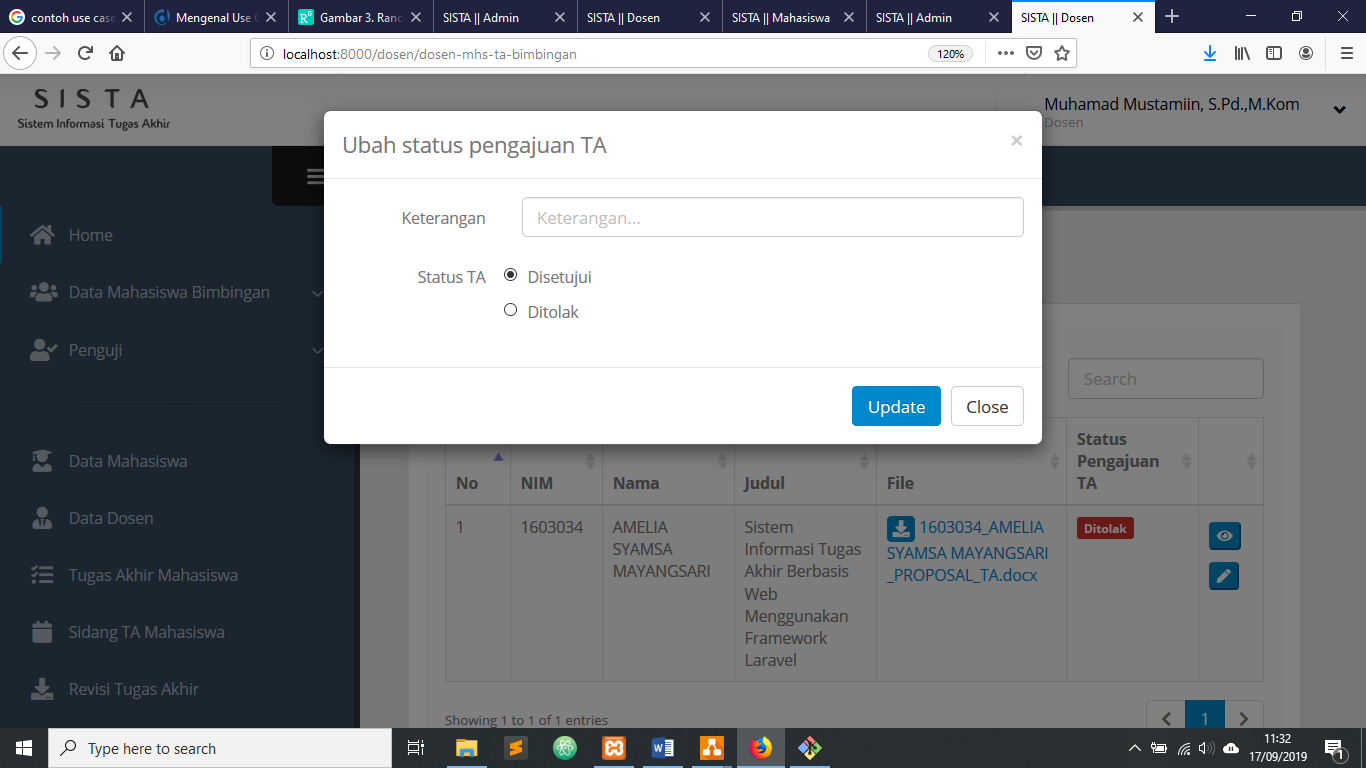
Halaman pembagian pembimbing adalah sebuah halaman dimana koordinator TA dapat menentukan pembimbing untuk setiap mahasiswa yang melakukan pengajuan tugas akhir. Adapun hasil tampilan halaman penentuan pembimbing dapat dilihat pada Gambar 4.18.



Gambar 4. 18 Tampilan Pembagian Pembimbing

1. **Tampilan Status Pengajuan TA**

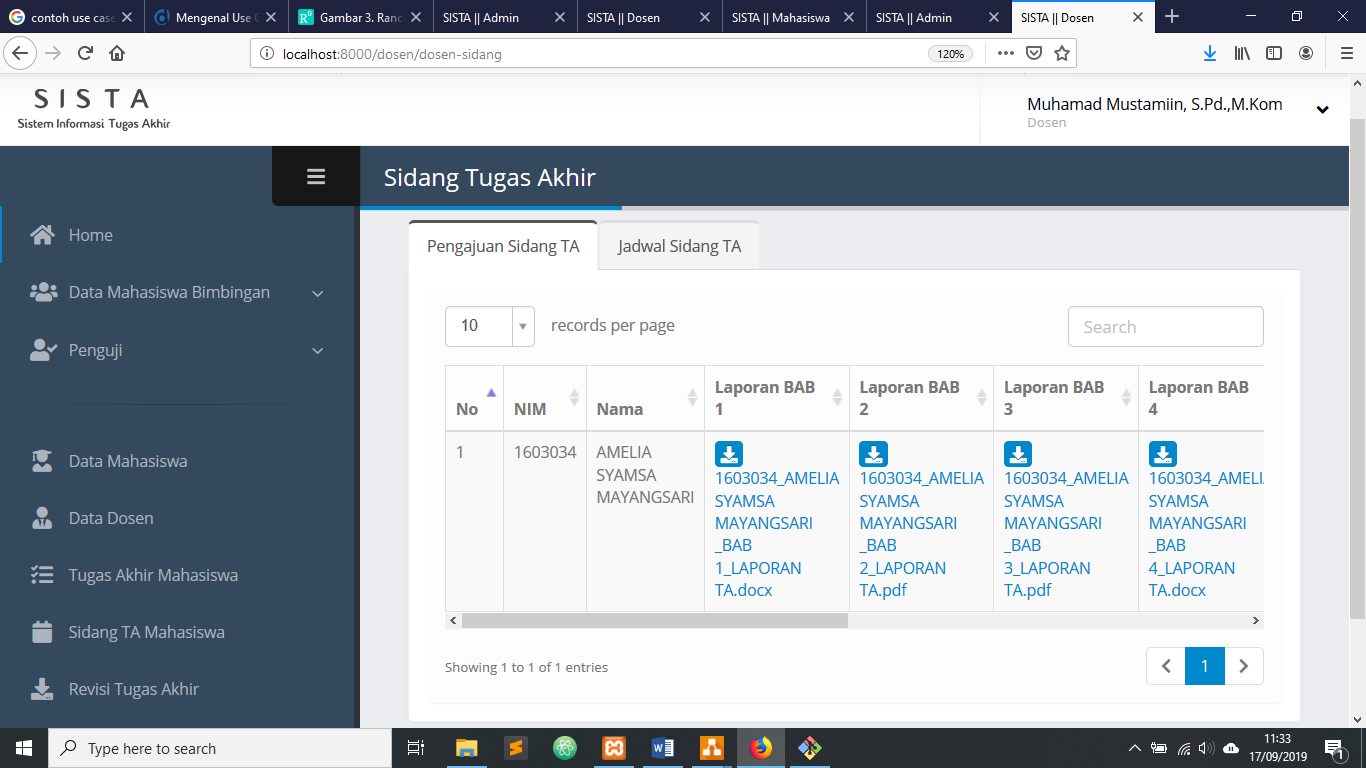
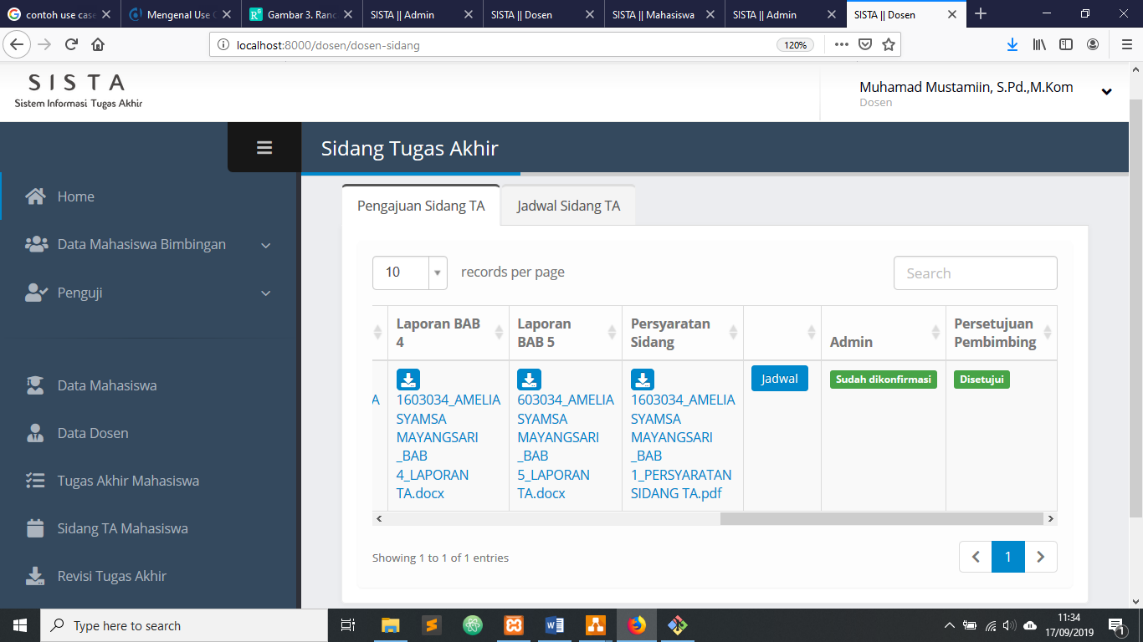
Halaman status pengajuan TA merupakan sebuah halaman dimana dosen pembimbing menyetujui ataupun menolak pengajuan TA dari mahasiswa bimbingannya. Adapun hasil tampilan halaman status pengajuan TA dapat dilihat pada Gambar 4.19.



Gambar 4. 19 Tampilan Status Pengajuan TA

1. **Tampilan List Pengajuan Sidang TA**

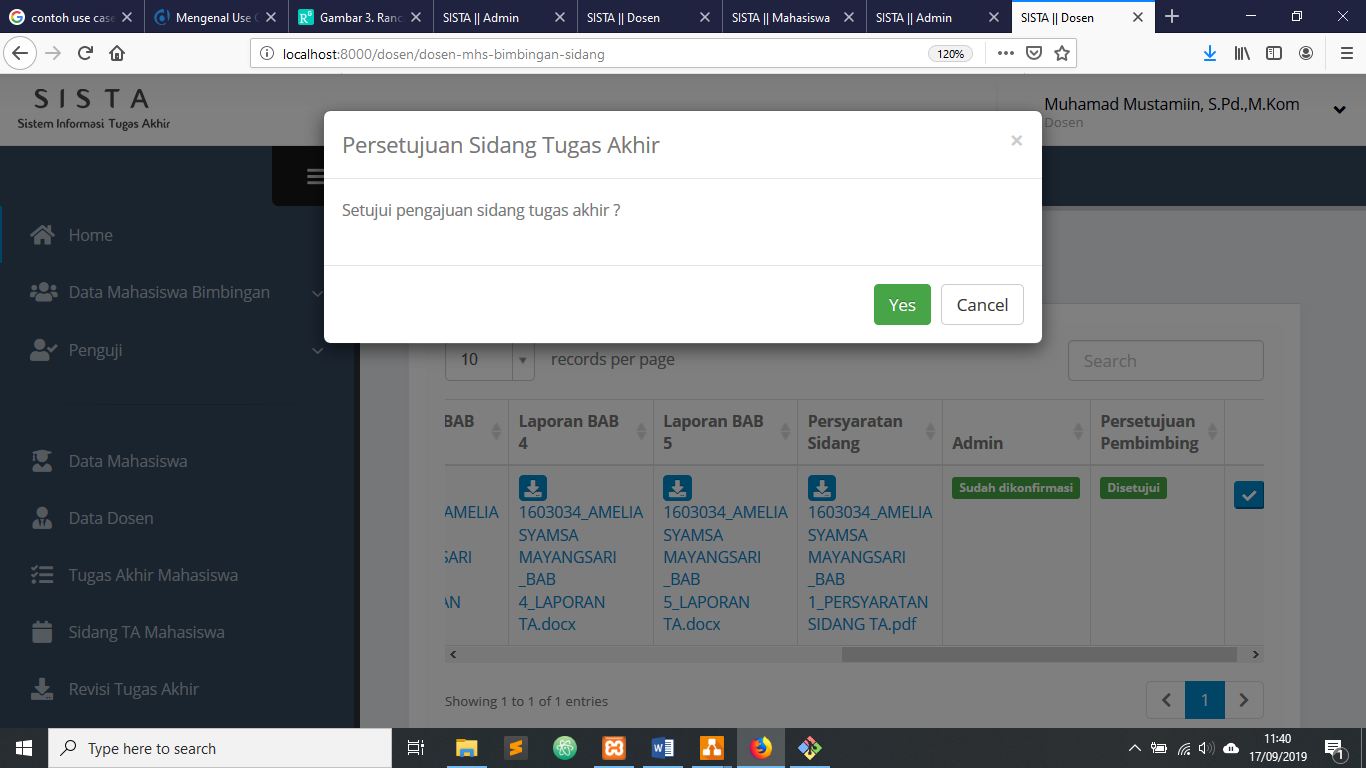
Halaman list pengajuan sidang TA adalah suatu halaman dimana ketua jurusan dan koordinator TA dapat melihat data pengajuan sidang TA seluruh dari mahasiswa, dosen pembimbing dapat melihat data pengajuan sidang TA dari mahasiswa bimbingannya, wali dosen dapat melihat data pengajuan sidang TA dari mahasiswa kelasnya. Selain itu ketua jurusan, koordinator TA, dosen pembimbing, dan wali dosen dapat mengunduh file laporan TA per bab. Adapun hasil tampilan halaman list pengajuan sidang TA dapat dilihat pada Gambar 4.20.



Gambar 4. 20 Tampilan List Pengajuan Sidang TA

1. **Tampilan Persetujuan Sidang TA**

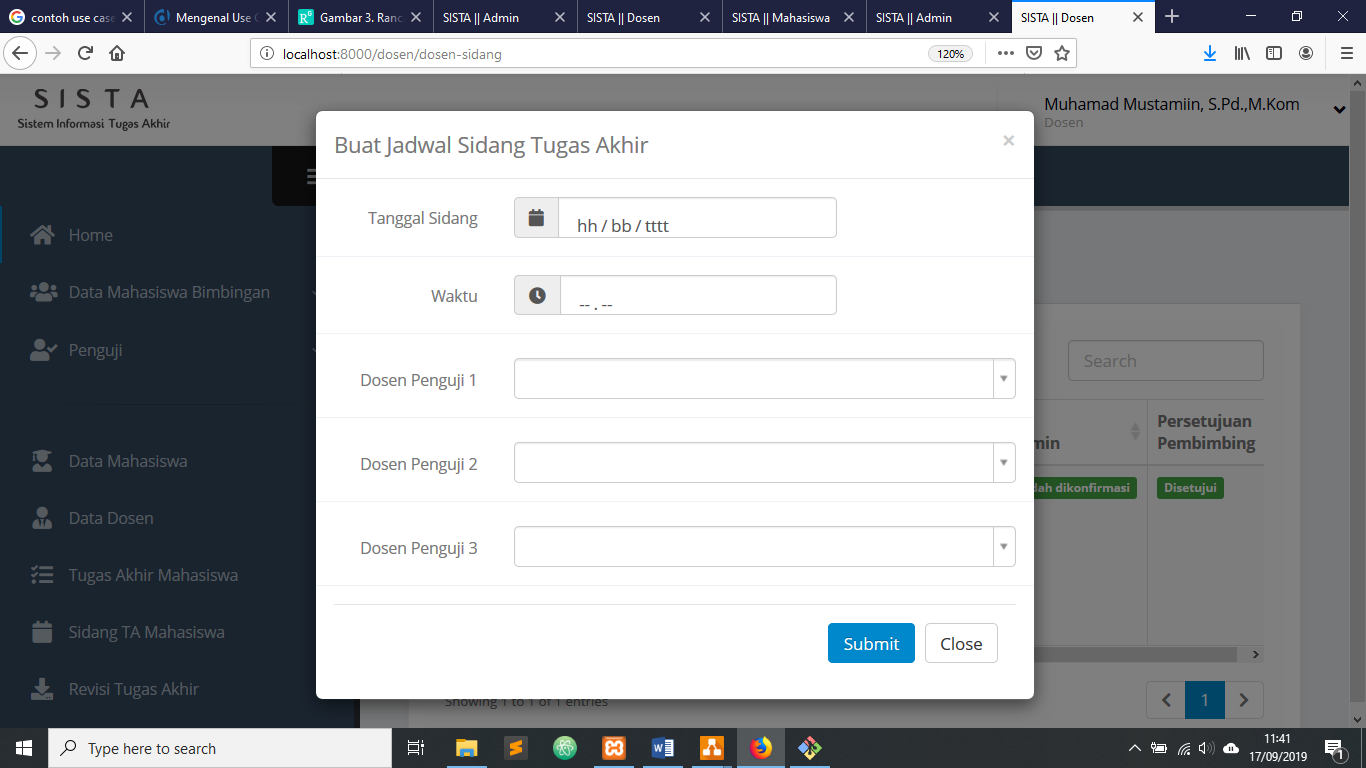
Halaman persetujuan sidang TA merupakan sebuah halaman dimana dosen pembimbing menyetujui pengajuan sidang TA dari mahasiswa bimbingannya. Adapun hasil tampilan halaman persetujuan TA dapat dilihat pada Gambar 4.21.



Gambar 4. 21 Tampilan Persetujuan Sidang TA

1. **Tampilan Tambah Jadwal Sidang TA**

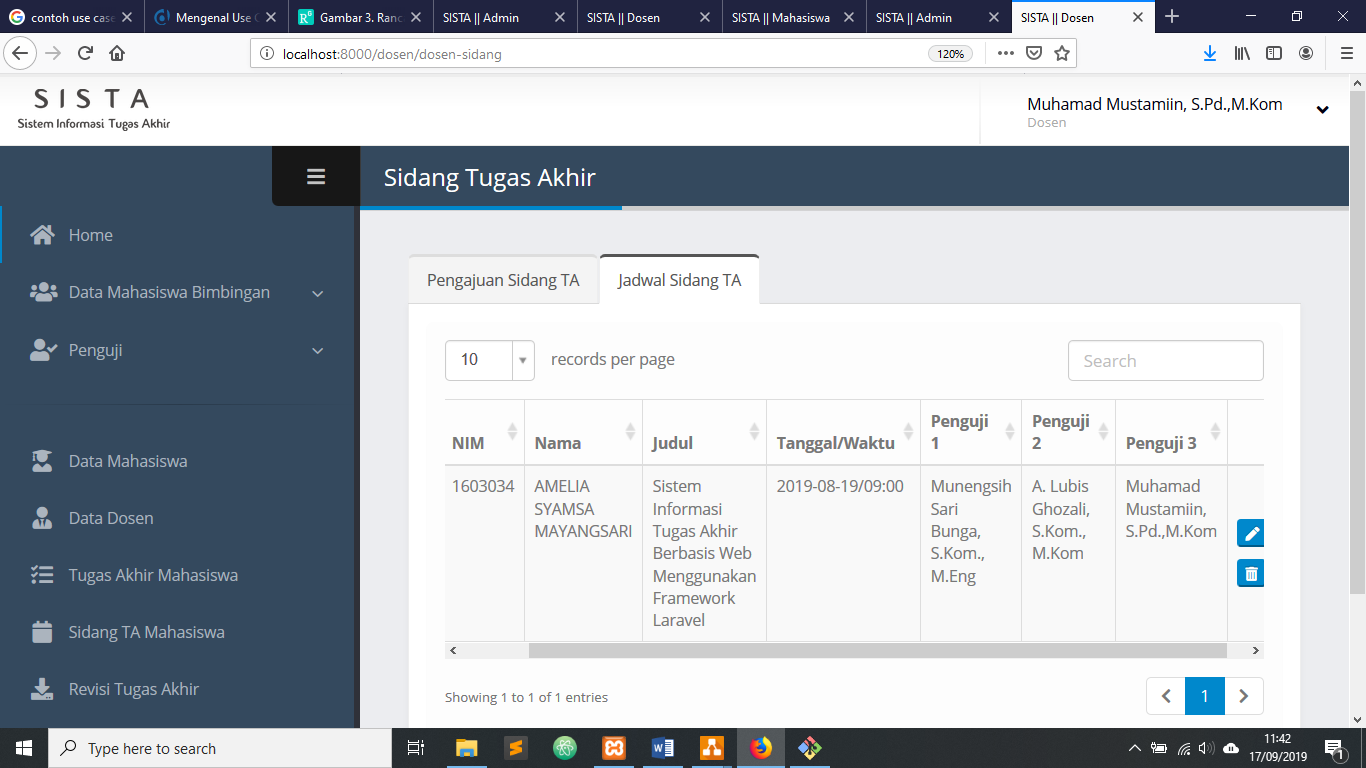
Halaman penentuan pembimbing adalah sebuah halaman dimana koordinator TA dapat menentukan pembimbing untuk setiap mahasiswa yang melakukan pengajuan tugas akhir. Adapun hasil tampilan halaman penentuan pembimbing dapat dilihat pada Gambar 4.22.



Gambar 4. 22 Tampilan Tambah Jadwal Sidang TA

1. **Tampilan List Jadwal Sidang TA**

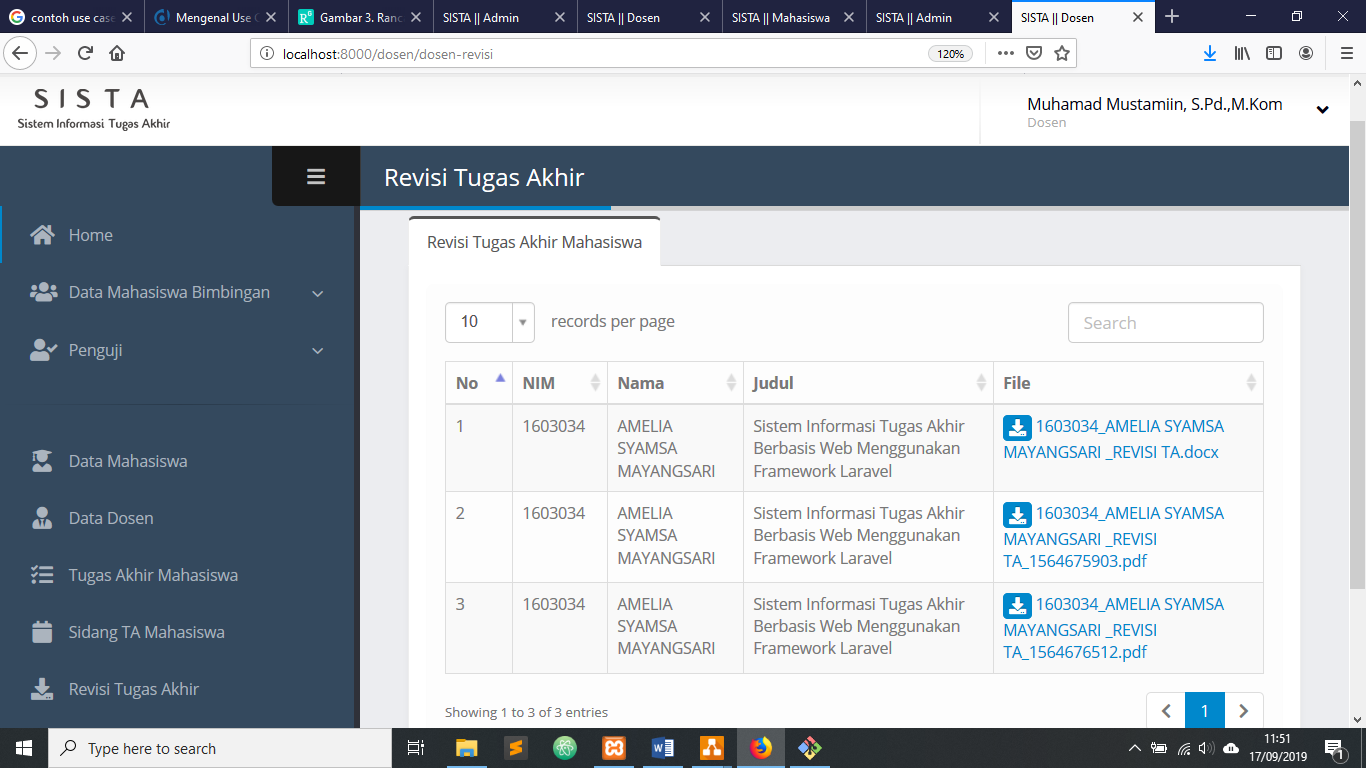
Halaman list jadwal sidang TA adalah suatu halaman dimana koordinator dapat melihat jadwal sidang TA seluruh dari mahasiswa, dosen pembimbing dapat melihat jadwal sidangTA dari mahasiswa bimbingannya, wali dosen dapat melihat jadwal sidangTA dari mahasiswa kelasnya. Selain itu ketua jurusan, koordinator TA, dosen pembimbing, dan wali dosen dapat mengunduh file laporan TA revisi tersebut. Adapun hasil tampilan halaman list jadwal sidang TA dapat dilihat pada Gambar 4.23.



Gambar 4. 23 Tampilan List Jadwal Sidang TA

1. **Tampilan List Revisi Lapoaran TA**

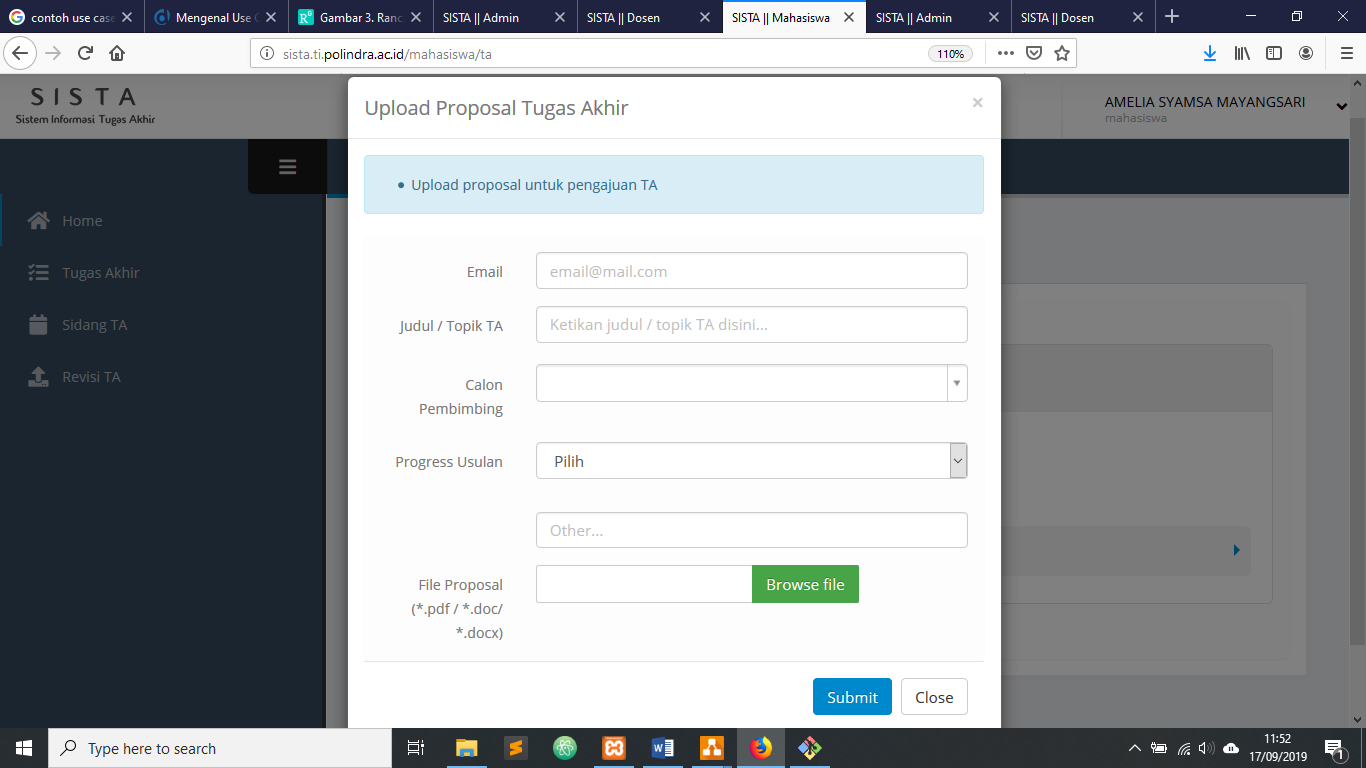
Halaman list revisi lapoaranTA adalah suatu halaman dimana ketua jurusan dan koordinator TA dapat melihat data revisi lapoaranTA seluruh dari mahasiswa, dosen pembimbing dapat melihat data revisi lapoaranTA dari mahasiswa bimbingannya, wali dosen dapat melihat data revisi lapoaranTA dari mahasiswa kelasnya. Selain itu ketua jurusan, koordinator TA, dosen pembimbing, dan wali dosen dapat mengunduh file laporan TA revisi tersebut. Adapun hasil tampilan halaman list revisi lapoaranTA dapat dilihat pada Gambar 4.24.



Gambar 4. 24 Tampilan List Revisi Laporan TA

1. **Tampilan Tambah Pengajuan TA**

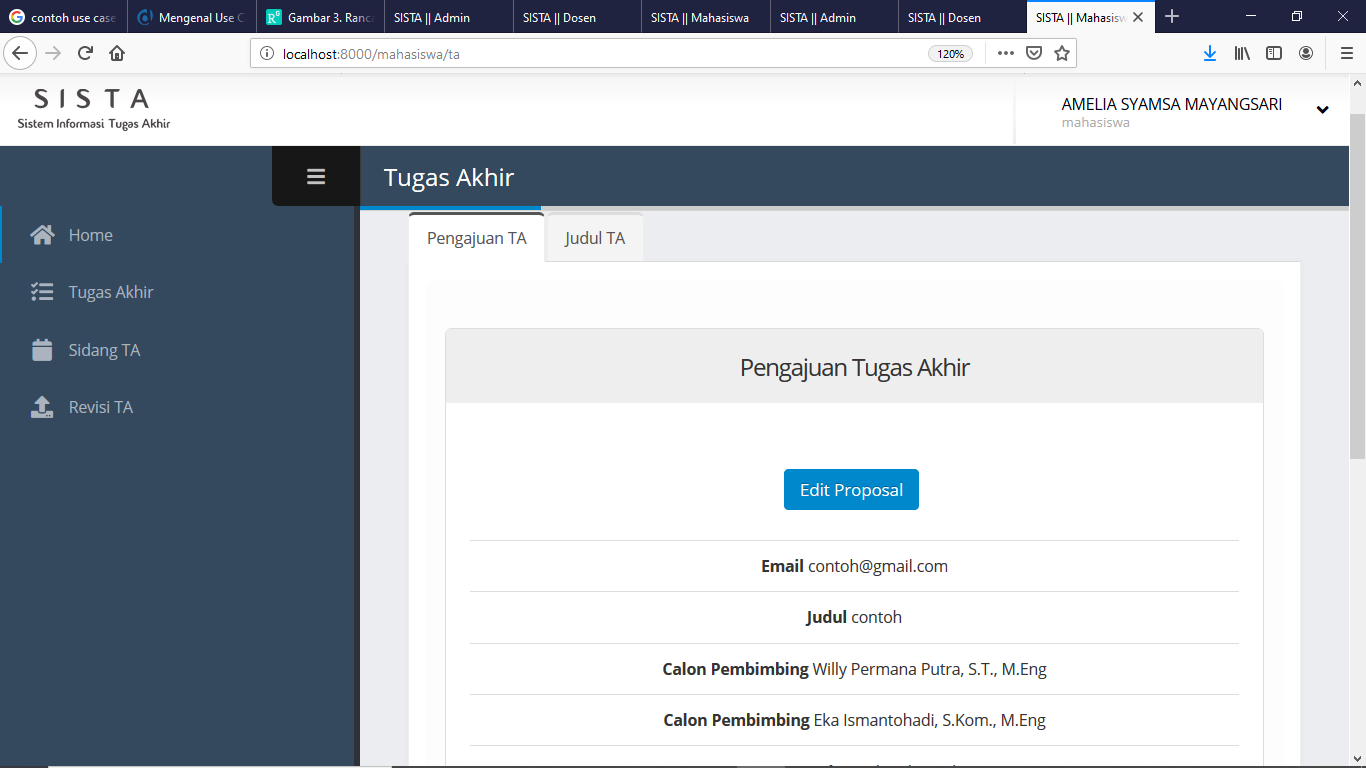
Halaman tambah pengajuan TA adalah suatu halaman dimana mahasiswa dapat melakukan pengajuan TA dengan mengisi form serta mengunggah proposal TA. Adapun hasil tampilan tambah pengajuanTA dapat dilihat pada Gambar 4.25.



Gambar 4. 25 Tampilan Tambah Pengajuan TA

1. **Tampilan Pengajuan TA Mahasiswa**

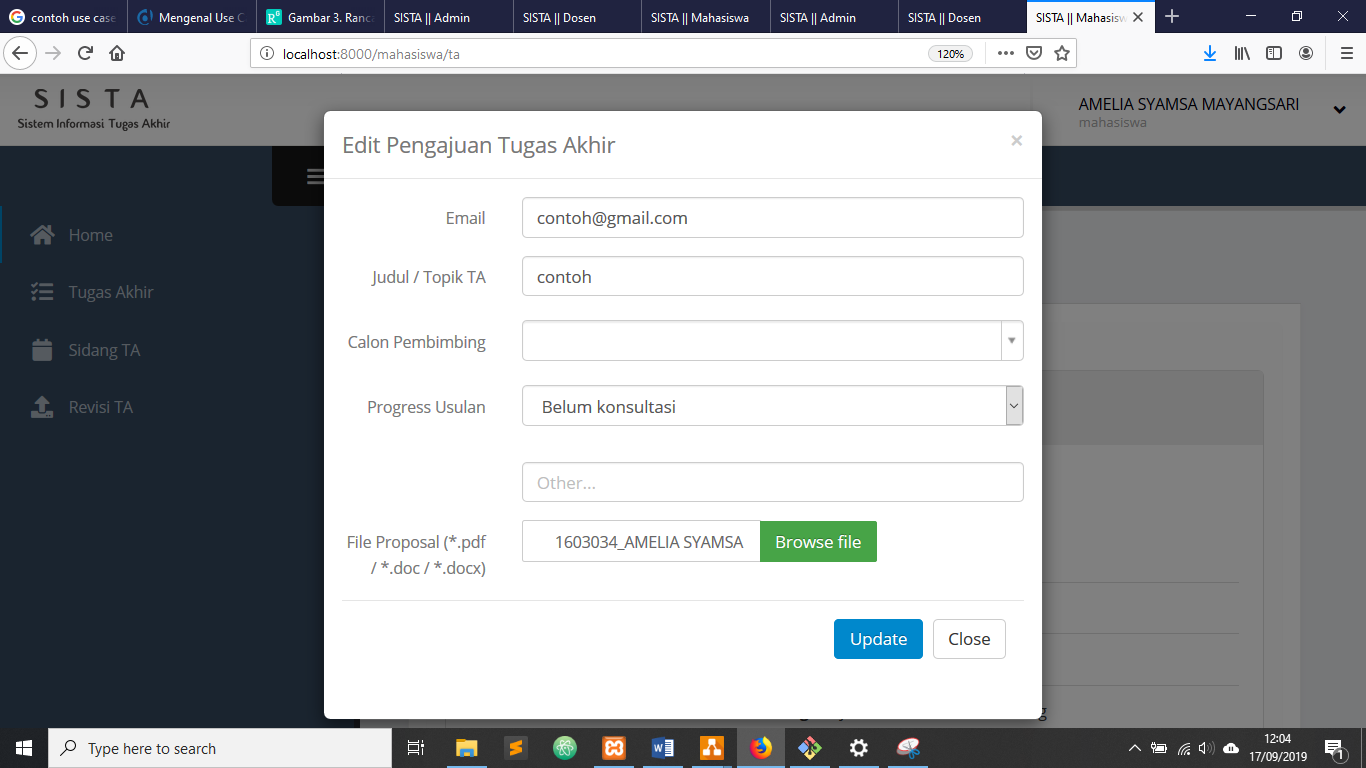
Halaman pengajuan mahasiswa merupakan halaman dimana mahasiswa dapat melihat data pengajuan TA yang telah dibuat sebelumnya. Mahasiswa juga dapat mengunduh kempali *file* yang telah unggah. Adapun hasil tampilan pengajuanTA dapat dilihat pada Gambar 4.26.



Gambar 4. 26 Tampilan Tambah Pengajuan TA

1. **Tampilan Ubah Pengajuan TA**

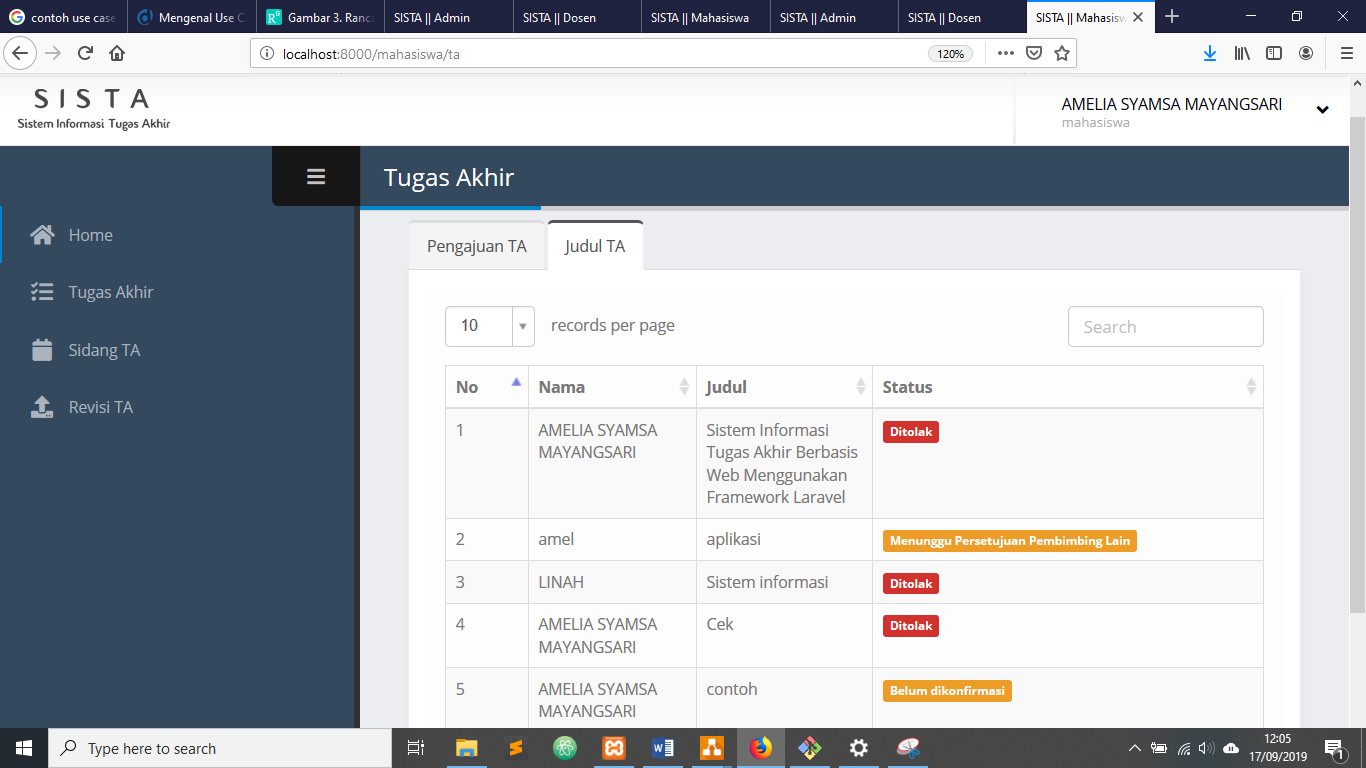
Halaman ubah pengajuan TA adalah halaman dimana mahasiswa dapat mengubah data pengjuan TA yang telah dibuat. Adapun hasil tampilan ubah pengajuanTA dapat dilihat pada Gambar 4.27.



Gambar 4. 27 Tampilan Ubah Pengajuan TA

1. **Tampilan List Judul TA**

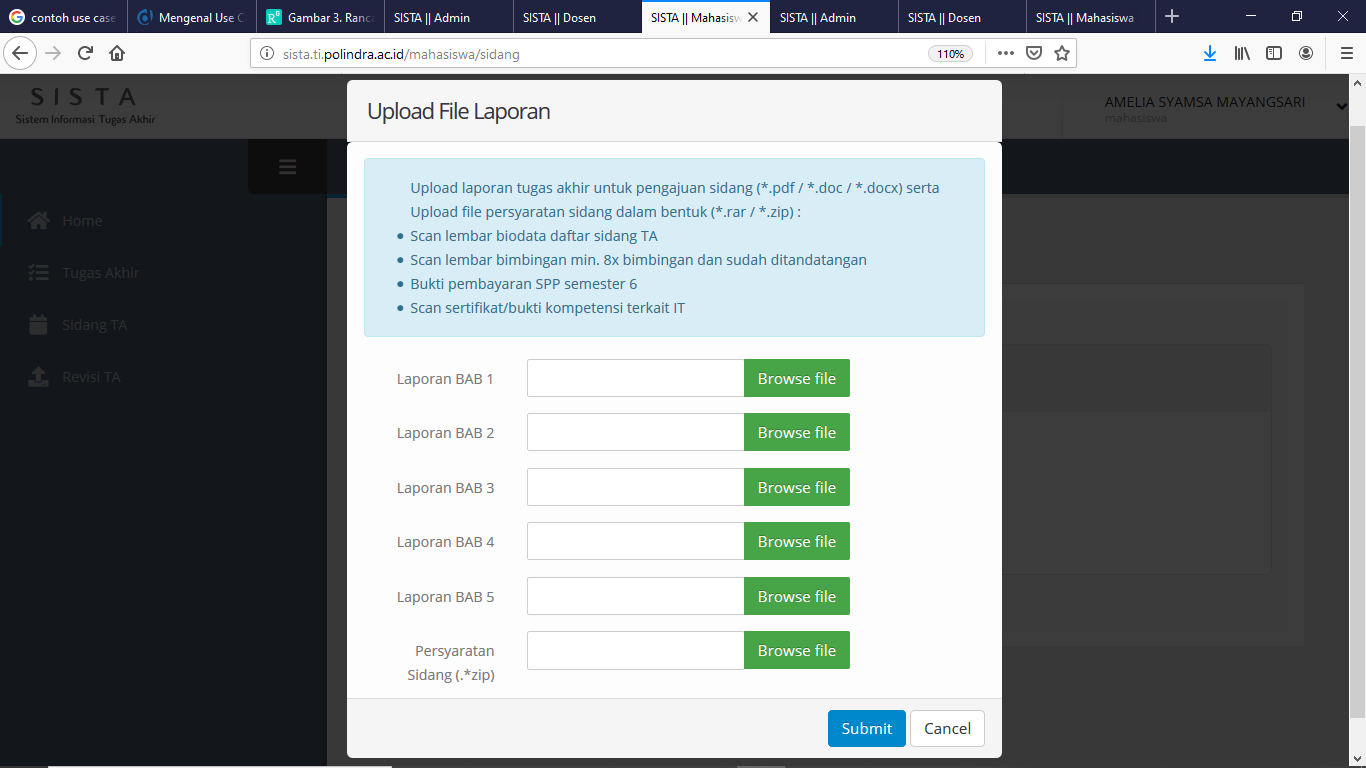
Halaman list judul TA mahasiswa merupakan halaman dimana mahasiswa dapat melihat list judul TA yang telah diajukan oleh mahasiswa lainnya. Adapun hasil tampilan list judulTA dapat dilihat pada Gambar 4.28.



Gambar 4. 28 Tampilan List Judul TA

1. **Tampilan Tambah Pengajuan Sidang TA**

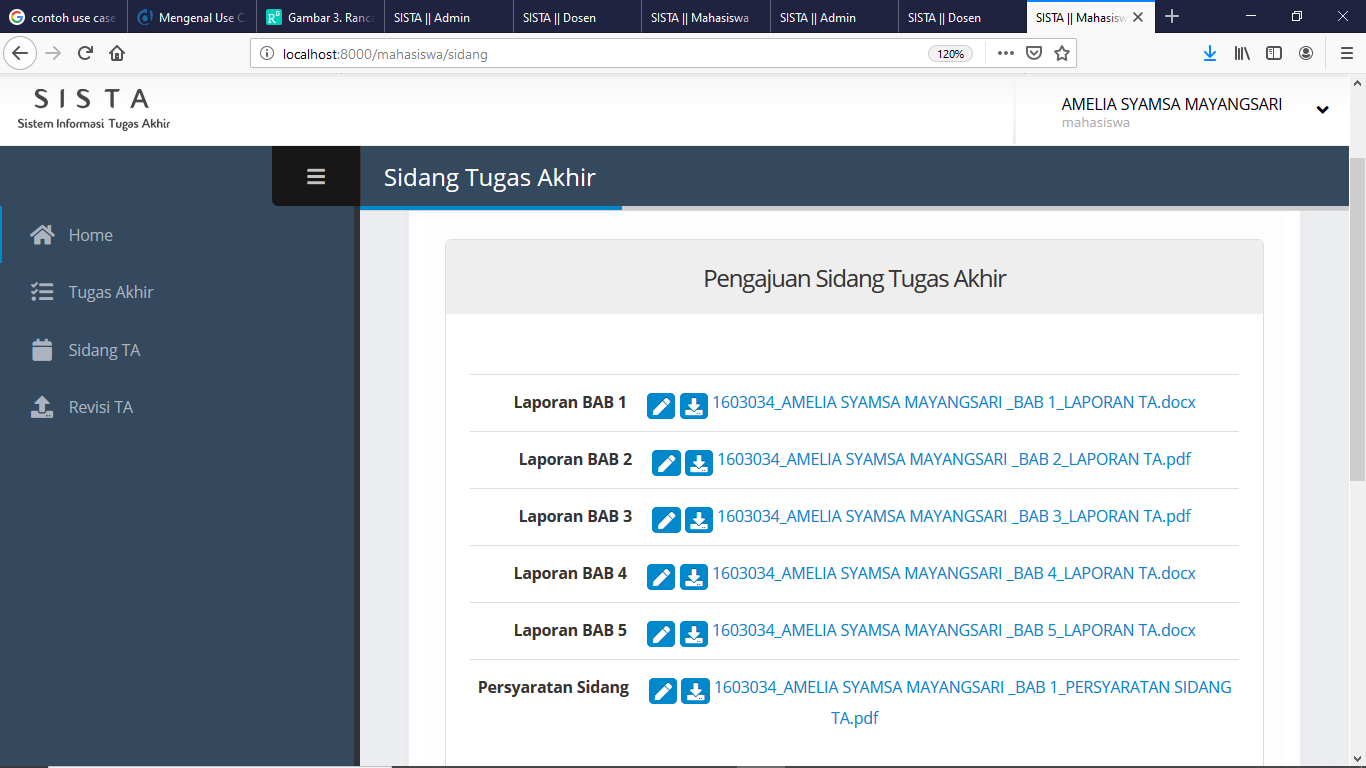
Halaman tambah pengajuan sidang TA adalah suatu halaman dimana mahasiswa dapat melakukan pengajuan sidang TA dengan mengisi form serta mengunggah *file* laporan TA per bab mulai dari bab 1 sampai bab 5 serta *file* persyaratan sidang. Adapun hasil tampilan tambah pengajuanTA dapat dilihat pada Gambar 4.29.

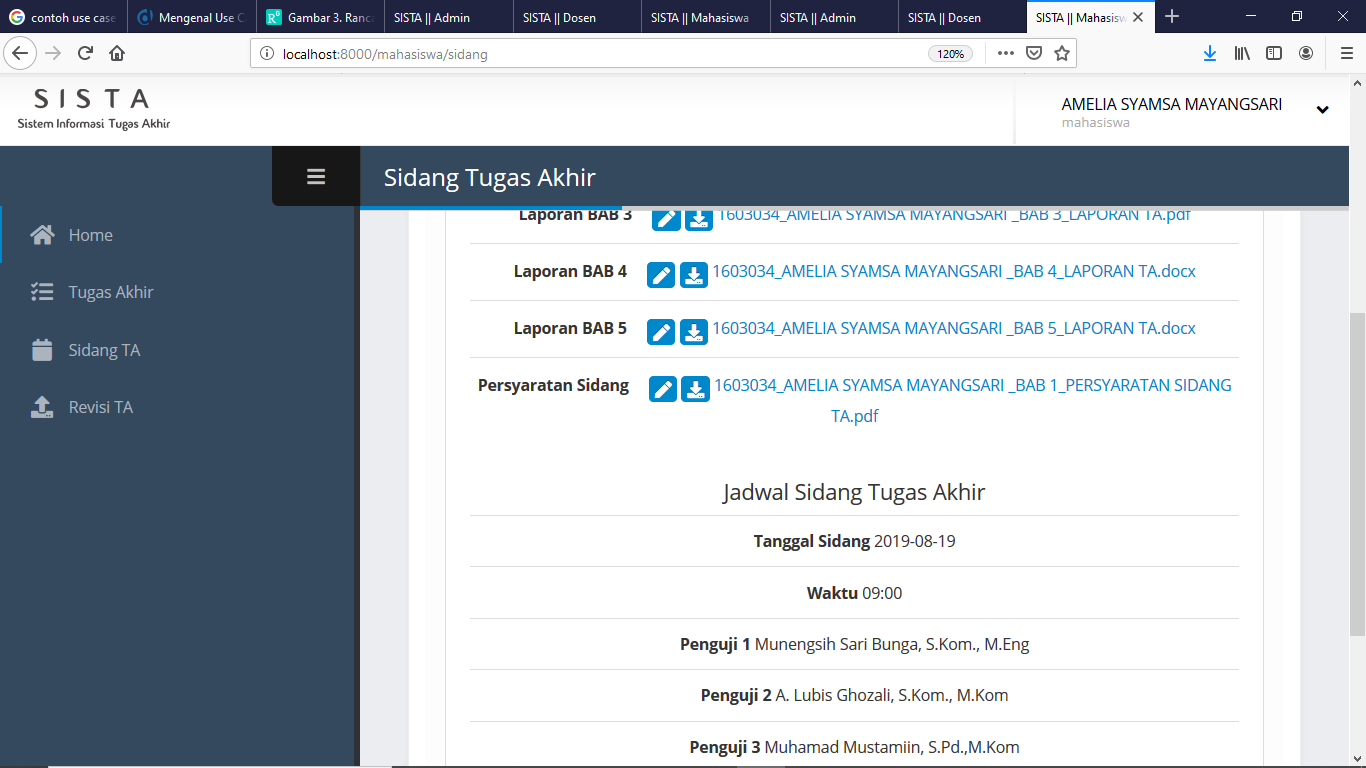


Gambar 4. 29 Tampilan Tambah Pengajuan Sidang TA

1. **Tampilan Pengajuan Sidang dan Jadwal Sidang TA**

Halaman pengajuan mahasiswa merupakan halaman dimana mahasiswa dapat melihat data pengajuan sidang TA yang telah dibuat sebelumnya. Mahasiswa juga dapat mengunduh kempali *file* yang telah unggah. Adapun hasil tampilan pengajuan sidangTA dapat dilihat pada Gambar 4.30.

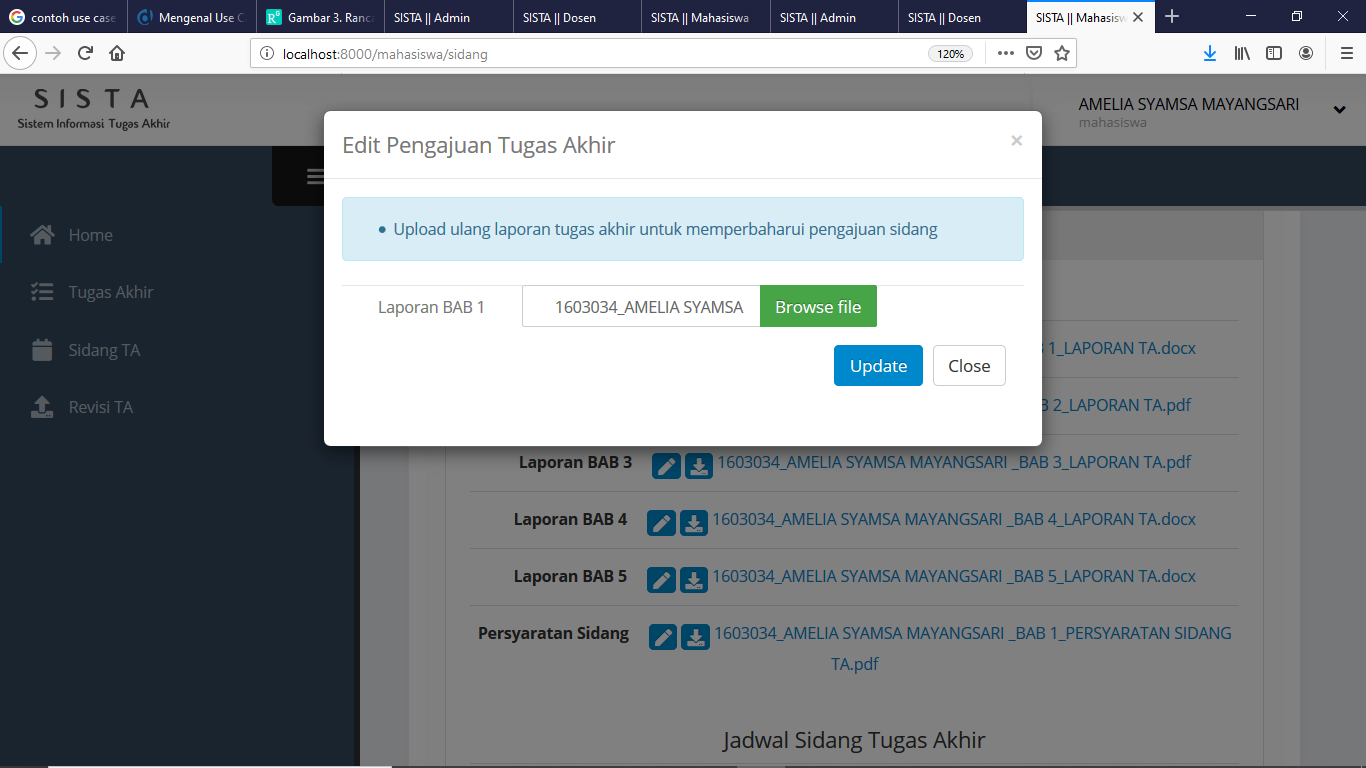




Gambar 4. 30 Tampilan Tambah Pengajuan Sidang dan Jadwal Sidang TA

1. **Tampilan Ubah Pengajuan Sidang TA**

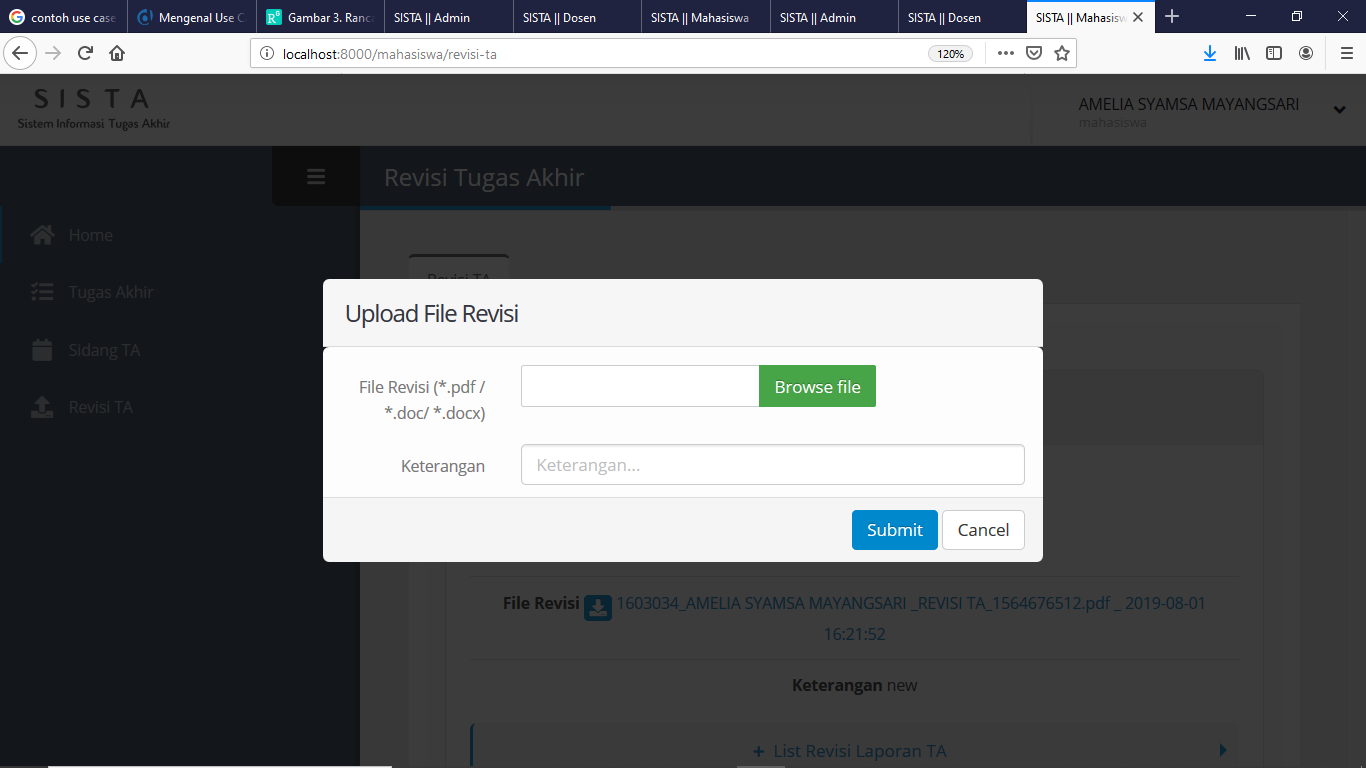
Halaman ubah pengajuan sidang TA adalah halaman dimana mahasiswa dapat mengubah data pengajuan sidang TA yang telah dibuat. Adapun hasil tampilan ubah pengajuansidangTA dapat dilihat pada Gambar 4.31.



Gambar 4. 31 Tampilan Ubah Pengajuan Sidang TA

1. **Tampilan Tambah Revisi TA**

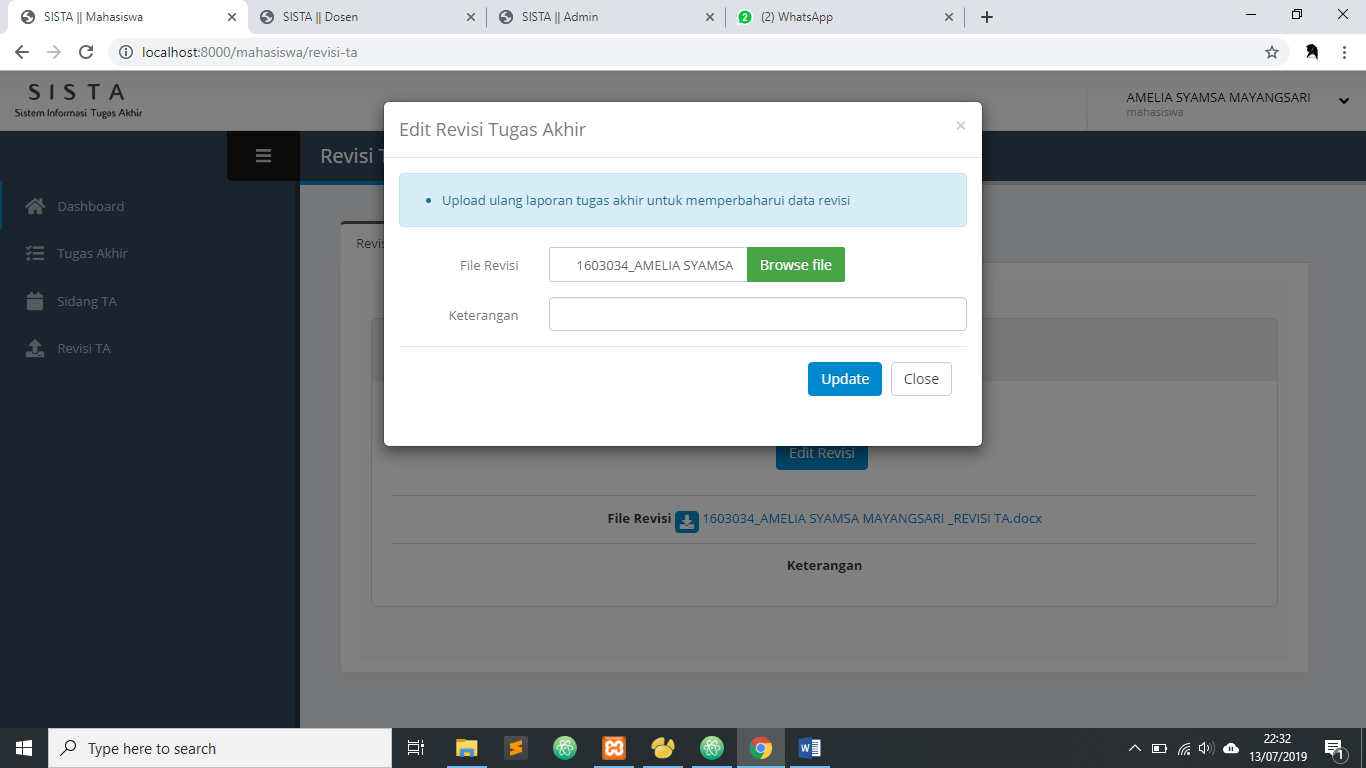
Halaman tambah revisi laporan TA adalah suatu halaman dimana mahasiswa dapat mengunggah *file* laporan TA yang telah diperbaiki. Adapun hasil tampilan tambah revisi laporanTA dapat dilihat pada Gambar 4.32.



Gambar 4. 32 Tampilan Tambah Revisi Laporan TA

1. **Tampilan Ubah Revisi TA**

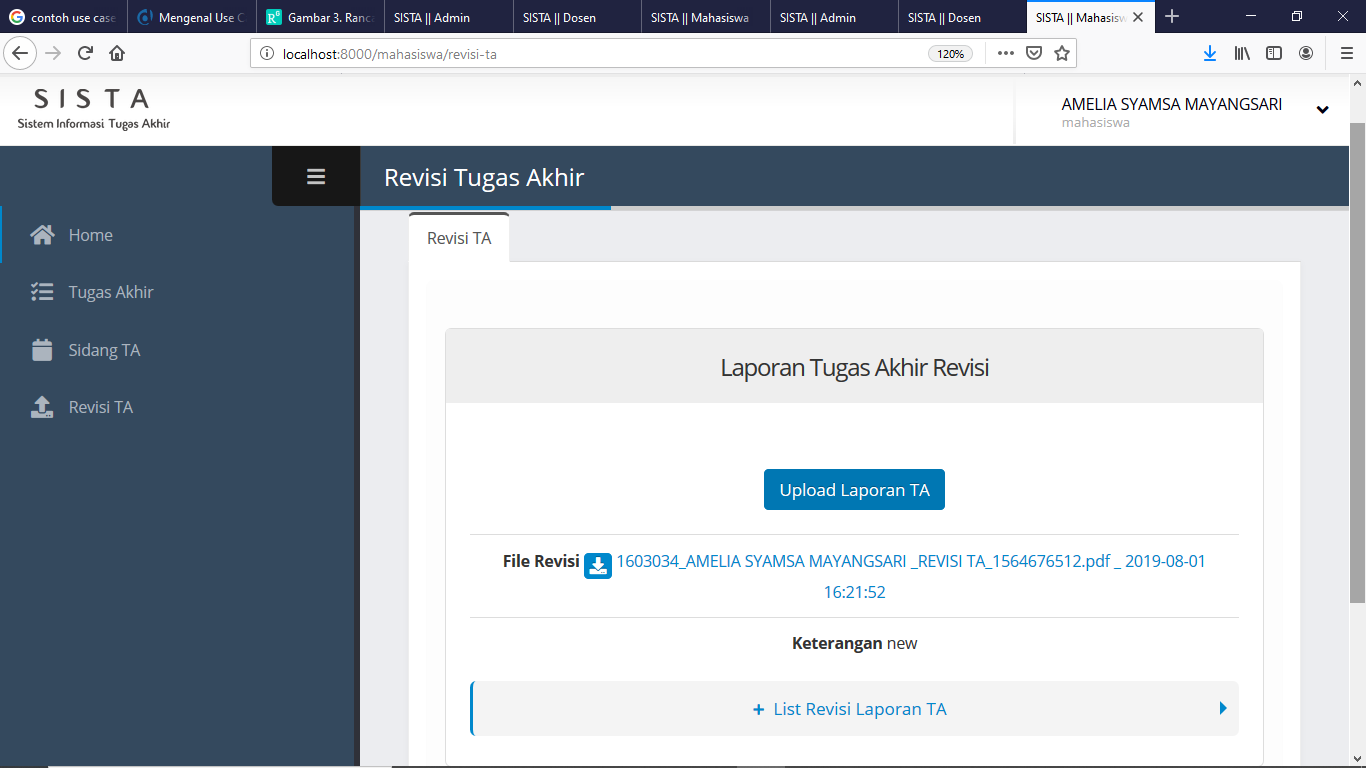
Halaman ubah revisi laporan TA adalah halaman dimana mahasiswa dapat mengubah *file* revisi laporan TA yang telah dibuat. Adapun hasil tampilan ubah revisi laporanTA dapat dilihat pada Gambar 4.33.



Gambar 4. 33 Tampilan Ubah Revisi TA

1. **Tampilan Revisi TA**

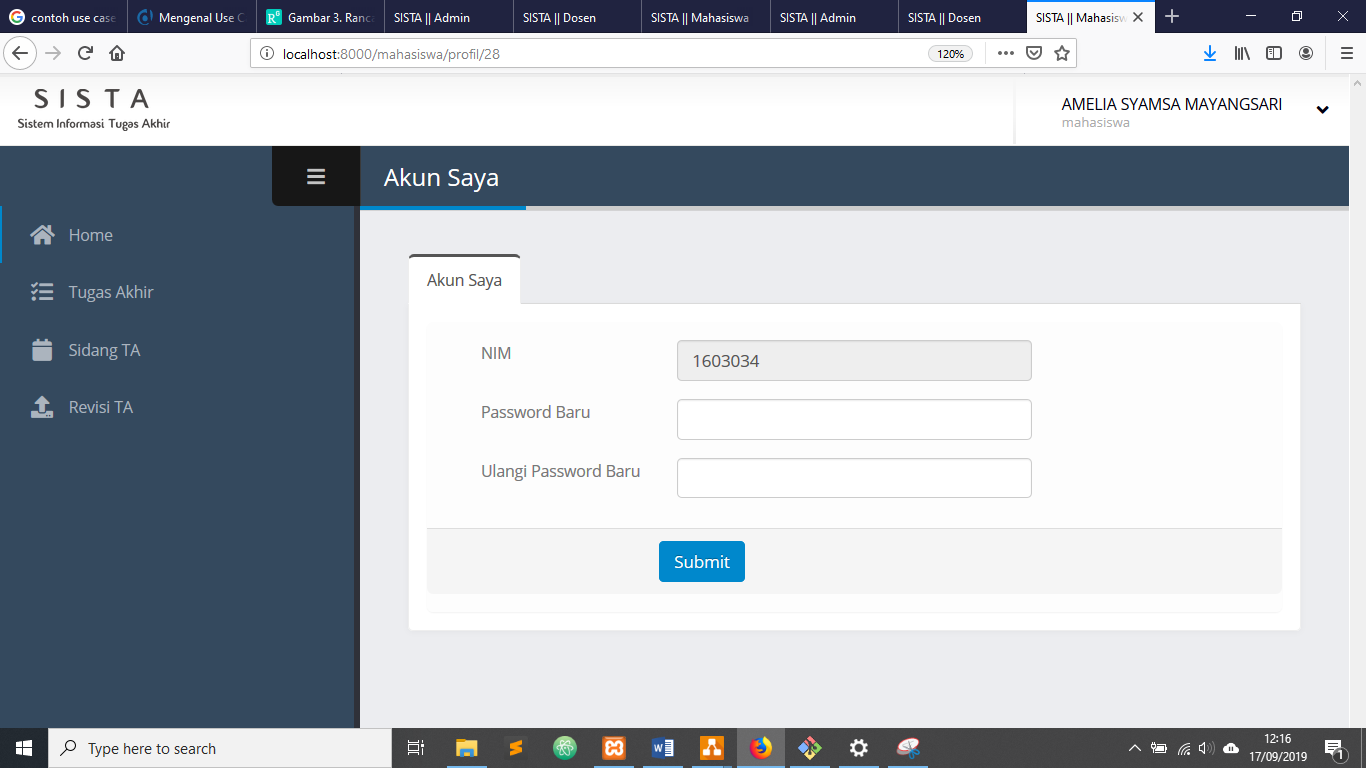
Halaman revisi laporan TA merupakan halaman dimana mahasiswa dapat melihat data revisi laporan TA yang telah dibuat sebelumnya. Mahasiswa juga dapat mengunduh kempali *file* yang telah unggah. Adapun hasil tampilan revisi laporanTA dapat dilihat pada Gambar 4.34.



Gambar 4. 34 Tampilan Revisi TA

1. **Tampilan Halaman Akun**

Halaman akun merupakan suatu halaman dimana *user* dapat melakukan perubahan data pada akun. Adapun rancangan halaman akun dapat dilihat pada Gambar 4.35.



Gambar 4. 35 Tampilan Halaman Akun

## **Pengujian**

Penulis melakukan pengujian sistem menggunakan kuisioner. Kuisioner tersebut terdiri dari 7 buah pertanyaan yang diberikan kepada 1 orang penguji sebagai administrator, 1 orang penguji sebagai dosen, dan 18 orang penguji sebagai mahasiswa dengan pertanyaan yang berbeda. Berikut adalah rekapitulasi pengujian kuisioner para penguji.

* + - 1. **Pertanyaan Kuisioner**

Keterangan:

A : Sangat Baik C : Cukup

B : Baik D : Kurang

Tabel 4. 16 Pertanyaan Kuisioner untuk Mahasiswa

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Penilaian** | | | |
| **A** | **B** | **C** | **D** |
| 1. | Bagaimana pendapat anda, apakah *website* ini cukup mudah digunakan? |  |  |  |  |
| 2. | Bagaimana pendapat anda tentang desain antarmuka dari *website* ini? |  |  |  |  |
| 3. | Apakah *website* ini dapat membantu anda dalam melakukan pengajuan tugas akhir? |  |  |  |  |
| 4. | Apakah *website* ini dapat membantu anda dalam melakukan pengajuan sidang tugas akhir? |  |  |  |  |
| 5. | Apakah semua fungsi pada website SISTA untuk mahasiswa dapat berjalan dengan baik? |  |  |  |  |
| 6. | Apakah SISTA bermanfaat bagi anda sebagai mahasiswa? |  |  |  |  |
| 7. | Apakah sistem ini mudah untuk dipahami? |  |  |  |  |

Tabel 4. 17 Pertanyaan Kuisioner untuk Administrator

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Penilaian** | | | |
| **A** | **B** | **C** | **D** |
| 1. | Bagaimana pendapat anda, apakah *website* ini cukup mudah digunakan? |  |  |  |  |
| 2. | Bagaimana pendapat anda tentang desain antarmuka dari *website* ini? |  |  |  |  |
| 3. | Bagaimana pendapat anda, apakah form input data mahasiswa dan dosen mudah digunakan? |  |  |  |  |
| 4. | Apakah *website* ini dapat membantu anda dalam mengkonfirmasi pengajuan sidang tugas akhir? |  |  |  |  |
| 5. | Apakah semua fungsi pada website SISTA untuk administrator dapat berjalan dengan  baik? |  |  |  |  |
| 6. | Apakah SISTA bermanfaat bagi anda sebagai administrator? |  |  |  |  |
| 7. | Apakah sistem ini mudah untuk dipahami? |  |  |  |  |

Tabel 4. 18 Pertanyaan Kuisioner untuk Dosen

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Pertanyaan** | **Penilaian** | | | |
| **A** | **B** | **C** | **D** |
| 1. | Bagaimana pendapat anda, apakah *website* ini cukup mudah digunakan? |  |  |  |  |
| 2. | Bagaimana pendapat anda tentang desain antarmuka dari *website* ini? |  |  |  |  |
| 3. | Apakah *website* ini dapat membantu anda dalam melakukan persetujuan tugas akhir? |  |  |  |  |
| 4. | Apakah *website* ini dapat membantu anda dalam melakukan persetujuan sidang tugas akhir? |  |  |  |  |
| 5. | Apakah semua fungsi pada website SISTA untuk dosen dapat berjalan dengan baik? |  |  |  |  |
| 6. | Apakah SISTA bermanfaat bagi anda sebagai dosen  ? |  |  |  |  |
| 7. | Apakah sistem ini mudah untuk dipahami? |  |  |  |  |

**4.2.2.2 Hasil Jawaban Kuesioner**

Tabel 4. 19 Hasil Jawaban Kuisioner

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Nama** | **Jawaban** | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|  | Wulan Dila Akpa | A | B | A | B | A | B | A |
|  | Ikhwanul Faqih | A | B | A | B | A | B | A |
|  | Hayatun Nupus | B | B | B | B | B | B | B |
|  | Sentanu Praja Wisesa | A | A | A | A | B | B | B |
|  | Widayanti | A | A | A | A | B | A | B |
|  | Febi Sandriya | A | A | B | B | A | A | A |
|  | Widi Indriyani | A | B | A | B | A | B | A |
|  | Ade Kartono | A | A | A | A | A | A | A |
|  | Nunung Nurhayati | A | B | A | B | A | B | A |
|  | Nezla Rustianingsih | B | B | A | B | A | A | A |
|  | Samsul Hadi | A | B | A | A | A | A | A |
|  | Nugraha Dwi Putra | A | A | A | A | A | A | A |
|  | Mohammad Rizaldi Afriza | A | B | B | A | C | A | A |
|  | Livia Neta | A | B | A | B | B | C | A |
|  | Triyani Ramadhani | A | B | A | B | A | B | A |
|  | Ahmad Pahruroji | B | B | A | B | B | A | B |
|  | Khairul Anwar | A | B | A | B | A | B | A |
|  | Ambar Nur Qori | B | A | A | B | A | B | B |
|  | Sonia Juliyanti | B | A | A | B | B | B | B |
|  | Muhamad Mustamiin | B | B | A | A | A | B | C |

Tabel 4. 20 Rekap Hasil Jawaban Kuisioner

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Sangat Baik** | **Baik** | **Cukup** | **Kurang** |
| 1. | 14 | 6 | 0 | 0 |
| 2. | 7 | 13 | 0 | 0 |
| 3. | 17 | 3 | 0 | 0 |
| 4. | 7 | 13 | 0 | 0 |
| 5. | 13 | 6 | 1 | 0 |
| 6. | 8 | 11 | 1 | 0 |
| 7. | 11 | 8 | 1 | 0 |

Dari hasil jawaban pengujian dengan kuisioner tersebut akan dihitung dengan menggunakan skala likert sebagai berikut:

Total responden = 20 Orang

Skor nilai:

A (Sangat baik) = 4

B (Baik) = 3

C (Cukup) = 2

D (Kurang) = 1

Skor tertinggi = 4 x 20 = 80

Skor terendah = 1 x 20 = 20

Skor kriteria = Jumlah responden x Skor nilai (4.1)

Rating Scale

76% - 100% = Sangat baik

50% - 75% = Baik

26% - 50% = Cukup

0% - 25% = Kurang

Presentase persetujuan

*p* = x 100% (4.2)

1. Pertanyaan Pertama

Sangat baik = *p* = x 100% = 70%

Baik = *p* = x 100% = 22.5%

Cukup = *p* = x 100% = 0%

Kurang = *p* = x 100% = 0% +

Jumlah = 92.5%

Dari hasil pengujian dengan kuisioner untuk pertanyaan pertama diperoleh presentase 92.5% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Kedua

Sangat baik = *p* = x 100% = 35%

Baik = *p* = x 100% = 48.75%

Cukup = *p* = x 100% = 0%

Kurang = *p* = x 100% = 0% +

Jumlah = 87.75%

Dari hasil pengujian dengan kuisioner untuk pertanyaan kedua diperoleh presentase 87.75% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Ketiga

Sangat baik = *p* = x 100% = 85%

Baik = *p* = x 100% = 11.25%

Cukup = *p* = x 100% = 0%

Kurang = *p* = x 100% = 0% +

Jumlah = 96.25%

Dari hasil pengujian dengan kuisioner untuk pertanyaan ketiga diperoleh presentase 96.25% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Keempat

Sangat baik = *p* = x 100% = 35%

Baik = *p* = x 100% = 48.75%

Cukup = *p* = x 100% = 0%

Kurang = *p* = x 100% = 0% +

Jumlah = 83.75%

Dari hasil pengujian dengan kuisioner untuk pertanyaan keempat diperoleh presentase 83.75% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Kelima

Sangat baik = *p* = x 100% = 65%

Baik = *p* = x 100% = 22.5%

Cukup = *p* = x 100% = 0%

Kurang = *p* = x 100% = 0% +

Jumlah = 87.5%

Dari hasil pengujian dengan kuisioner untuk pertanyaan kelima diperoleh presentase 87.5% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Keenam

Sangat baik = *p* = x 100% = 40%

Baik = *p* = x 100% = 41.25%

Cukup = *p* = x 100% = 2.5%

Kurang = *p* = x 100% = 0% +

Jumlah = 87.75%

Dari hasil pengujian dengan kuisioner untuk pertanyaan keenam diperoleh presentase 87.75% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

1. Pertanyaan Ketujuh

Sangat baik = *p* = x 100% = 55%

Baik = *p* = x 100% = 30%

Cukup = *p* = x 100% = 2.5%

Kurang = *p* = x 100% = 0% +

Jumlah = 87.5%

Dari hasil pengujian dengan kuisioner untuk pertanyaan ketujuh diperoleh presentase 87.5% dan pertanyaan tersebut masuk ke dalam kategori sangat baik.

## **Rekapitulasi Hasil Perhitungan Kuesioner**

Dari hasil pengujian dengan kuesioner untuk semua pertanyaan dapat diperoleh rata – rata presentase seperti berikut :

92.5% + 87.75% + 96.25% + 83.75% + 87.5% + 87.75% + 87.5% = 89%

7

Dari hasil perhitungan jawaban kuesioner dapat disimpulkan bahwa rata-rata presentase dengan jumlah rata – rata presentase adalah 89%. Sistem ini secara keseluruhan masuk kedalam kategori sangat baik untuk gunakan.

## **Kesimpulan Pengujian**

Berdasarkan hasil pengujian yang dilakukan oleh penulis dengan kuisioner maka dapat diambil kesimpulan bahwa aplikasi yang dibuat sudah berjalan dengan baik. Pada dasarnya hasil pengujian aplikasi sesuai harapan dan membantu administrator, koordinator TA, mahasiswa, dosen pembimbing, ketua jurusan, dan wali dosen dalam mengkoordinasi proses pelakanaan tugas akhir. Hasil pengujian tersebut menyatakan bahwa tidak terdapat kesalahan – kesalahan/error, sehingga dapat dikatakan bahwa aplikasi sistem informasi tugas akhir berbasis web ini sudah amat baik digunakan oleh pengguna.

## **Perbaikan Sistem**

Tidak menutup kemungkinan aplikasi yang telah berhasil dibuat saat ini terhindar dari kesalahan yang disebabkan oleh sistem maupun *human error* dimasa yang akan datang, maka dari itu diperlukan suatu perawatan dan perbaikan sehingga aplikasi dapat berjalan dengan lebih baik dari sebelumya.

# **BAB V**

# **PENUTUP**

## **Kesimpulan**

Kesimpulan yang didapat dari penelitian ini setelah melakukan tahapan mengumpulkan data, menganalisis kebutuhan, mengembangkan dan melakukan pengujian sistem yang dihasilkan adalah sebagai berikut.

1. Aplikasi sistem informasi tugas akhir berbasis web ini dirancang menggunakan UML (*Unified Modelling Language*), *flowchart*, dan perancangan antarmuka (*mockup*) untuk memberikan gambaran mengenai alur sistem.
2. Aplikasi sistem informasi tugas akhir berbasis web ini dibuat menggunakan bahasa pemrograman PHP, framework laravel, dan MySQL sebagai pengolah basis data.
3. Berdasarkan hasil kuisioner dengan presentase 92.5%, aplikasi sistem informasi tugas akhir berbasis web ini mudah digunakan oleh administrator, koordinator TA, mahasiswa, dosen pembimbing, ketua jurusan, dan wali dosen dalam mengkoordinasi proses pelakanaan tugas akhir seperti pengajuan TA, pengajuan sidang TA, judul TA, dan jadwal sidang TA.

## **Saran**

Dalam pembuatan aplikasi sistem informasi tugas akhir berbasis web ini tidak jauh dari kelemahan dan kekurangan pada sistem yang dibuat. Berdasarkan hasil dan pembahasan terdapat beberapa saran agar sistem yang dibuat lebih baik dan sempurna dan diharapkan memperhatikan saran berikut :

1. Dapat ditambahkan fitur untuk bimbingan tugas akhir antara dosen pembimbing dan mahasiswa.
2. Jadwal sidang tugas akhir dapat di *generate* secara otomatis agar sistem dapat lebih mudah digunakan.
3. Aplikasi sistem informasi tugas akhir ini dapat digunakan oleh semua jurusan di Politeknik Negeri Indramayu.
4. Berdasarkan hasil presentase kuesioner terkecil aplikasi ini sudah masuk kedalam kategori sangat baik untuk digunakan tetapi masih perlu dikembangkan lagi khususnya pada fitur sidang TA agar semakin baik dan mempermudah pengguna dalam melakukan pengajuan sidang TA, persetujuan sidang TA, dan konfirmasi sidang TA.

# **DAFTAR PUSTAKA**

Ahaddin, Fachrul. 2015. Apa itu Activity Diagram. https://www.dumetschool.com/blog/Apa-Itu-Activity-Diagram. Diakses pada tanggal 12 September 2019.

Andre. 2017. Belajar HTML Pengertian HTML. <https://www.duniailkom.com>.

Diakses pada tanggal 27 Januari 2019.

Anonim. 2019. https://laravel.com/. Diakses pada tanggal 12 September 2019.

Anonim. 2019. https://www.google.com/. Diakses pada tanggal 12 September 2019.

Ariata. 2018. Apa Itu HTML? Pemahaman Dasar Tentang Bahasa Markup Hypertext. <https://www.hostinger.co.id/tutorial/apa-itu-html/>. Diakses pada tanggal 21 Mei 2019.

Arif. 2018. Pengertian composer. <http://www.kursuswebsite.org/pengertian-composer/>. Diakses pada tanggal 21 Mei 2019.

Arifwicaksanaa. 2016. Pengertian Use Case. <https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf>. Diakses pada tanggal 20 Juli 2019.

Aritias Anggaeni, Puspita, Bambang Sujatmiko. 2013. Jurnal Manajemen Informasi. Jurnal Manajemen Informatika. Volume 02.

Budiaji, Weksi. 2013. Jurnal Ilmu Pertanian dan Perikanan Desember 2013. Volume 2. Halaman 129.

Cox, Dan. 2019. Server-Side PHP: Part 1: Server Output. https://videlais.com/2019/05/11/server-side-php-part-1-server-output/. Diakses pada tanggal 30 Agustus 2019.

Daniel, 2016. Pengertian Xampp dan Fungsinya. [http://www.kursus*website*.org/pengertian-xampp-dan-fungsi-nya/](http://www.kursuswebsite.org/pengertian-xampp-dan-fungsi-nya/). Diakses pada tanggal. Diakses pada tanggal 08 Agustus 2019*.*

Dharmawijaya, Handoko. 2018. Data dan Informasi. https://slideplayer.info/slide/12727094/. Diakses pada tanggal 30 Agustus 2019.

Dudung. 2019. Pengertian Komponen dan Fungsi Xampp Lengkap Dengan

Penjelasannya*.* <https://www.dosenpendidikan.com>. Diakses pada tanggal 27 Januari 2019.

Efendi, Ilham. 2014. Pengertian dan Tipe Diagram UML. <https://www.it-jurnal.com/pengertian-dan-tipe-diagram-uml/>. Diakses pada tanggal 19 Juli 2019.

Fadul, Fadullah. 2018. Belajar Bootstrap Untuk Pemula. Apa itu Bootstrap ?. https://www.apacara.com. Diakses pada tanggal 04 Mei 2019.

Febiyan, Arya. 2014. Apa itu composer. <https://www.dumetschool.com/blog/Apa-Itu-Composer>. Diakses pada tanggal 21 Mei 2019.

Fikri Setiadi, M. 2017. Kupan Tuntas Konsep Model Data Entity Relationship Diagram atau ERD. http://mfikri.com/artikel/kupas-tuntas-konsep-model-data-entity-relationship-diagram-atau-erd.html. Diakses pada tanggal 31 Agustus 2019.

Galandi, Fitho. 2016. Metode Waterfall : Definisi, Tahapan, Kelebihan dan Kekurangan.http://www.pengetahuandanteknologi.com/2016/09/metode-waterfall-definisi-tahapan.html. Diakses pada tanggal 23 Agustus 2019.

Ghulam Asshokin, Muhammad, Deddy Kusbianto Purwoko Aji, Ariadi Retno Trihayati Ririd. 2015. Jurnal Informatika Polinema. Volume 1.

Gumilar Pratama, Ega. 2017. Mengenal Sekilas Jenis – jenis Flowchart Pada Pemrograman. <https://www.codepolitan.com/mengenal-sekilas-jenis-jenis-flowchart-pada-pemrograman-589be38b9417f>. Diakses pada tanggal 20 Juli 2019.

Halwa, RF. 2018. Apa itu CSS (Cascading Style Sheets) ?. <https://www.apacara.com/tutorial/apa/apa-itu-css-cascading-style-sheets.html>. Diakses pada tanggal 21 Mei 2019.

Harefa, Efriaman. 2016. Pengertian Website.https://www.citratek.co.id*.* Diakses pada tanggal 02 Maret 2019.

Indra, Reza. 2016. Mengenal Atom, Text Editor yang Sangat Powerful. https://www.dumetschool.com. Diakses pada tanggal 23 April 2019.

Julianto, Heri. Apa itu ERD ?. 2016. https://www.academia.edu/29325646/Apa\_itu\_ERD. Diakses pada tanggal 31 Agustus 2019.

Kamaludin. 2012. Perancangan Arsitektur Perangkat Lunak Dengan Metode Uml Pada Disain Penampang Rangka Batang Baja.

Kurniawati, Peni. 2018. Pengujian Sistem. https://medium.com/skyshidigital/pengujian-sistem-52940ee98c77. Diakses pada tanggal 23 Agustus 2019.

Muhardian, Ahmad. 2016. Belajar HTML dari Nol: Pengenalan Dasar HTML untuk Pemula. https://www.petanikode.com/html-dasar/. Diakses pada tanggal 12 September 2019.

Natalia Tanuwijaya, Christina. Domain Class Diagram. 2016. <http://sis.binus.ac.id/2016/06/20/domain-class-diagram/>. Diakses pada tanggal 20 Juli 2019.

Noviana, Eddy, Otang Kurniaman, Muhammad Nailul Huda. 2018. [Pengembangan Aplikasi Bimbingan Tugas Akhir Mahasiswa Berbasis Website pada Program Studi Pendidikan Guru Sekolah Dasar FKIP Universitas Riau](https://www.neliti.com/publications/258208/pengembangan-aplikasi-bimbingan-tugas-akhir-mahasiswa-berbasis-website-pada-prog). Jurnal Primary Program Studi Pendidikan Guru Sekolah Dasar Fakultas Keguruan dan Ilmu Pendidikan Universitas Riau. Volume 7. Halaman 2.

Novitasari, Candra. 2018. Pengertian Class Diagram Contoh dan Simbolnya. <https://pelajarindo.com/pengertian-class-diagram-simbol/>. Diakses pada tanggal 20 Juli 2019.

Nur, Nahya. 2016. Grey Box Testing. http://share.its.ac.id/blog/index.php?entryid=1027. Diakses pada tanggal 23 Agustus 2019.

Osd, Dimas. 2015. Pengertian SI (Sistem Informasi)*.* https://www.kompasiana.com. Diakses pada tanggal 13 Maret 2019.

Putra, Achmad Irianto Eka. 2018. Apa itu Framework Laravel. https://medium.com/@ryanbekhen/apa-itu-framework-laravel-e3b53cb75536. Diakses pada tanggal 12 September 2019.

Rahman, Syakir. 2013. Mengenal apa itu CSS dan bagaimana menggunakannya. https://www.devaradise.com/id/2013/08/mengenal-apa-itu-css-dan-bagaimana-menggunakannya.html. Diakses pada tanggal 12 September 2019.

Rahmatullah Pratama, Aditiya. 2019. Belajar UML : Use Case Diagram. <https://www.codepolitan.com/mengenal-uml-diagram-use-case>. Diakses pada tanggal 20 Juli 2019

Rasyid, Fadjar Efendy. 2014. Bahasa Pemrograman Populer PHP.h[ttps://](ttps://www.ubaya.ac.id.)

www. ubaya.ac.id. Diakses pada tanggal 27 April 2019.

Rendra. 2018. Javascript adalah : Pengertian Javascript dan Dasar Penggunaan Javascript. https://www.webhostingterbaik.id. Dikakses pada tanggal 03 April 2019.

Rohman, Nur. 2019. Pengertian Kuesioner, Jenis dan Contohnya. https://akuntanonline.com/pengertian-kuesioner-jenis-dan-contohnya/. Diakses pada tanggal 12 September 2019.

Salamadian. 2017. Simbol Flowchart : Pengertian, Jenis, Fungsi dan Contohnya. https://salamadian.com/simbol-simbol-flowchart/. Diakses pada tanggal 30 Agustus 2019.

Sora. 2014. Pengertian Basis Data dan Sistem Basis Data http://www.pengertianku.net. Diakses pada tanggal 09 Maret 2019.

Shy-gan, Edy. Daftar Simbol UML. https://id.scribd.com/document/143412967/Daftar-Simbol-Uml. Diakses pada tanggal 30 Agustus 2019.

Ubaid Salim, Yusrah, Sri Widaningsih. 2017. e-Proceeding of Applied Science. Volume 3.

Winarso, Bambang. 2016. Apa Itu Google Chrome.https://dailysocial.id/post. Diakses pada tanggal 27 April 2019.

Yudhanto, Yudho & Helmi Adi Prasetyo. 2018. Panduan Mudah Belajar Framework Laravel. Jakarta: PT Alex Media Komputindo.

# **LAMPIRAN**

**LAMPIRAN 1 HASIL OBSERVASI**

Data Mahasiswa Tingkat 3 D3 Teknik Informatika

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **NIM** | **NAMA** | **KELAS** |
| 1 | 1603001 | ABDUL MUHALI | 16-D3TI-3A |
| 2 | 1603002 | ADE HILMI AZIS | 16-D3TI-3A |
| 3 | 1603004 | ADHITYA FATURROCHMAN | 16-D3TI-3A |
| 4 | 1603005 | CHRISTIAN EKA WIBISONO | 16-D3TI-3A |
| 5 | 1603006 | CICA SUSANTI | 16-D3TI-3A |
| 6 | 1603007 | DANIYAH HAYATI | 16-D3TI-3A |
| 7 | 1603008 | DENDY OCTAVIAN | 16-D3TI-3A |
| 8 | 1603009 | FADLI SEPTIAN NAWAWI | 16-D3TI-3A |
| 9 | 1603010 | FAHMI MUHAMMAD | 16-D3TI-3A |
| 10 | 1603011 | GHEBINUQI ZAHRA FIADNIN | 16-D3TI-3A |
| 11 | 1603012 | HAFIDZ AUGUSTIAN PRATAMA | 16-D3TI-3A |
| 12 | 1603013 | HAMDAN MAULANA | 16-D3TI-3A |
| 13 | 1603014 | KARLINA | 16-D3TI-3A |
| 14 | 1603016 | LINAH | 16-D3TI-3A |
| 15 | 1603017 | LISDA ISMAYANTI | 16-D3TI-3A |
| 16 | 1603018 | MAWAR UTARI | 16-D3TI-3A |
| 17 | 1603021 | NANDIA RAHADIAN NUGRAHA | 16-D3TI-3A |
| 18 | 1603023 | PANDI SAHARA | 16-D3TI-3A |
| 19 | 1603024 | REFI DWI YANTI | 16-D3TI-3A |
| 20 | 1603025 | RIYAN ADITIYA | 16-D3TI-3A |
| 21 | 1603026 | TITA ANISA | 16-D3TI-3A |
| 22 | 1603027 | UCI RAMADHANI | 16-D3TI-3A |
| 23 | 1603028 | UMYATI | 16-D3TI-3A |
| 24 | 1603029 | WULAN DILLA AKPA | 16-D3TI-3A |
| 25 | 1603030 | YANUAR ALI AZIZ | 16-D3TI-3A |
| 26 | 1603032 | AHMAD PAHRUROJI | 16-D3TI-3B |
| 27 | 1603033 | AMBAR NUR QORI | 16-D3TI-3B |
| 28 | 1603034 | AMELIA SYAMSA MAYANGSARI | 16-D3TI-3B |
| 29 | 1603035 | DEWI JAYANTI | 16-D3TI-3B |
| 30 | 1603036 | DHANDI TUHBAGUS KAESSAR | 16-D3TI-3B |
| 31 | 1603037 | DIAN RAHAYU | 16-D3TI-3B |
| 32 | 1603038 | DIKI DARMAWAN FITRIYADI | 16-D3TI-3B |
| 33 | 1603040 | FEBI SANDRIYA | 16-D3TI-3B |
| 34 | 1603041 | HAYATUN NUFUS | 16-D3TI-3B |
| 35 | 1603042 | HAYATUN NUPUS | 16-D3TI-3B |
| 36 | 1603043 | IKHWANUL FAQIH | 16-D3TI-3B |
| 37 | 1603044 | KHAIRUL ANWAR | 16-D3TI-3B |
| 38 | 1603045 | KHURI SYAHIDA RIYADI | 16-D3TI-3B |
| 39 | 1603046 | LIVIA NETA INARA | 16-D3TI-3B |
| 40 | 1603047 | LUFIANTI | 16-D3TI-3B |
| 41 | 1603048 | MOEHAMMAD RAIFALDHI | 16-D3TI-3B |
| 42 | 1603049 | MOHAMMAD RIZALDI AFRIZA | 16-D3TI-3B |
| 43 | 1603050 | MUHAMAD FAUZY | 16-D3TI-3B |
| 44 | 1603051 | NEZLA RUSTIANINGSIH | 16-D3TI-3B |
| 45 | 1603053 | NUGRAHA DWI PUTRA ARIFUDIN | 16-D3TI-3B |
| 46 | 1603054 | RIZKHAN AMRULLAH | 16-D3TI-3B |
| 47 | 1603055 | RIZKY AKBAR LAZUARDI | 16-D3TI-3B |
| 48 | 1603056 | SAMSUL HADI | 16-D3TI-3B |
| 49 | 1603057 | SENTANU PRAJA WISESA | 16-D3TI-3B |
| 50 | 1603059 | VIVIAN SAHRUL YUSUF | 16-D3TI-3B |
| 51 | 1603060 | WIDAYANTI | 16-D3TI-3B |
| 52 | 1603061 | ANNISA KARTIKA UTAMI | 16-D3TI-3C |
| 53 | 1603063 | ARI IRAWAN | 16-D3TI-3C |
| 54 | 1603064 | ARIE FAHMI LUTHFI | 16-D3TI-3C |
| 55 | 1603065 | DIKY ANWAR | 16-D3TI-3C |
| 56 | 1603066 | DINA MICELA | 16-D3TI-3C |
| 57 | 1603067 | DIYAH AYU AMALIYAH | 16-D3TI-3C |
| 58 | 1603068 | FIKIH NUR RAMADAN | 16-D3TI-3C |
| 59 | 1603069 | FIRDA FARADILA | 16-D3TI-3C |
| 60 | 1603070 | HASRI HABBEANA PERTIWI | 16-D3TI-3C |
| 61 | 1603071 | IMAM MUHAYAT | 16-D3TI-3C |
| 62 | 1603072 | INNEKE WIDIANTI | 16-D3TI-3C |
| 63 | 1603073 | IRIN WINDIYATI | 16-D3TI-3C |
| 64 | 1603074 | KIKIS MAULANA | 16-D3TI-3C |
| 65 | 1603075 | LULU MUKHOYYAROH | 16-D3TI-3C |
| 66 | 1603076 | LUQMANUL HAKIM | 16-D3TI-3C |
| 67 | 1603078 | MUHAMMAD BIMO ABDITAMA | 16-D3TI-3C |
| 68 | 1603079 | MOCHAMAD FAIZAL | 16-D3TI-3C |
| 69 | 1603080 | NOUFAL IBRAHIM | 16-D3TI-3C |
| 70 | 1603081 | NUNUNG NURHAYATI | 16-D3TI-3C |
| 71 | 1603082 | RIZALUDDIN SIDQI BAIHAQI | 16-D3TI-3C |
| 72 | 1603083 | SHAFA DHIYANTI | 16-D3TI-3C |
| 73 | 1603085 | STEPHAN DWIKI ALKINE | 16-D3TI-3C |
| 74 | 1603086 | TRIYANI RAMADHANI | 16-D3TI-3C |
| 75 | 1603087 | WIDI INDRIYANI | 16-D3TI-3C |
| 76 | 1603088 | WIDIANA SILVI | 16-D3TI-3C |
| 77 | 1603089 | YOSHIE PANGESTU | 16-D3TI-3C |
| 78 | 1503029 | ADE KARTONO | 16-D3TI-3C |
| 79 | 1603091 | ASTRI ALISAH | 16-D3TI-3D |
| 80 | 1603092 | BAHRAINSYAH OKSAREINALDI | 16-D3TI-3D |
| 81 | 1603093 | BAYU RASUKMA RAGA | 16-D3TI-3D |
| 82 | 1603094 | DIYANTI | 16-D3TI-3D |
| 83 | 1603095 | DODI HIDAYAT | 16-D3TI-3D |
| 84 | 1603096 | EKO PRASTYO | 16-D3TI-3D |
| 85 | 1603097 | FIRMANSYAH | 16-D3TI-3D |
| 86 | 1603098 | FIRSTI AULYA KHUSNUL KHOTIMAH | 16-D3TI-3D |
| 87 | 1603099 | HILMY LAZUARDI | 16-D3TI-3D |
| 88 | 1603100 | ISMATUL MAULA | 16-D3TI-3D |
| 89 | 1603101 | JAKARIA | 16-D3TI-3D |
| 90 | 1603102 | JENIA ADELLIA PUSPITA | 16-D3TI-3D |
| 91 | 1603103 | KORIAH | 16-D3TI-3D |
| 92 | 1603104 | LIA NUR FADILAH | 16-D3TI-3D |
| 93 | 1603105 | LUVI HAERUNISAH | 16-D3TI-3D |
| 94 | 1603106 | MAULANA AHMAD QUSYAERI | 16-D3TI-3D |
| 95 | 1603107 | MUHAMMAD FAUJI AL FARIZ | 16-D3TI-3D |
| 96 | 1603109 | NUR INAYATUN MAHMUDA | 16-D3TI-3D |
| 97 | 1603111 | RIZKY ALIEF SATRIA | 16-D3TI-3D |
| 98 | 1603112 | SUJIYANTO | 16-D3TI-3D |
| 99 | 1603113 | SYAHRUL GUNAWAN | 16-D3TI-3D |
| 100 | 1603114 | SYAHRUL ROMADHONI | 16-D3TI-3D |
| 101 | 1603115 | SYAMSUL FAHRI HADI | 16-D3TI-3D |
| 102 | 1603116 | TUTI NURAFNI AMALIA | 16-D3TI-3D |
| 103 | 1603118 | WULAN DIANI | 16-D3TI-3D |

Data Dosen Jurusan Teknik Informatika

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 9098630 | A. Sumarudin, S.Pd., M.T., M.Sc | Dosen |
| 2 | 8048123 | Eka Ismantohadi, S.Kom., M.Eng | Dosen & Sekretaris Jurusan |
| 3 | 8098642 | Willy Permana Putra, S.T., M.Eng | Dosen & Ka. Laboratorium |
| 4 | 8098543 | Munengsih Sari Bunga, S.Kom., M.Eng | Dosen |
| 5 | 8096544 | Ir. Muh. Lukman Sifa | Dosen |
| 6 | 8098145 | Darsih, S.Kom., M.Kom | Dosen |
| 7 | 9098631 | A. Lubis Ghozali, S.Kom., M.Kom | Dosen |
| 8 | 9058001 | Mohammad Yani, S.T., M.T., M.Sc | Dosen |
| 9 | 17039203 | Muhamad Mustamiin, S.Pd.,M.Kom | Dosen |
| 10 | 17099206 | Fachrul Pralienka Bani Muhamad, S.ST., M.Kom | Dosen |
| 11 | 17089004 | Adi Suheryadi, S.ST.,M.Kom | Dosen |
| 12 | 16039001 | Iryanto, S.Si., M.Si | Dosen & Ketua Jurusan |
| 13 | 16079312 | Alifia Puspaningrum, S.Pd., M.Kom. | Dosen |
| 14 | 8888888 | Esti Mulyani, S.Kom., M.Kom | Dosen |
| 15 | 9999999 | Kurnia Adi Cahyanto, S.T., M.Kom | Dosen |
| 16 | 1111111 | Muh. Anis Al Hilmi, S.Si., M.T | Dosen |

**LAMPIRAN 2 KODE PROGRAM**

Kode Program Import Data Mahasiswa *User* Administrator

|  |
| --- |
| public function store(Request $request){  //dd($request);  $messages = [  'required' => ':attribute wajib diisi',  'numeric' => ':attribute harus angka',  'max' => ':attribute maksimal :max karakter',  ];  $this->validate($request, [  'nim' => 'required|unique:data\_mahasiswa|numeric',  'nama' => 'required|max:100',  'kelas' => 'required|max:15',  ], $messages);  $mhs = DataMahasiswa::create([  'nim' => $request->nim,  'nama' => $request->nama,  'kelas' => $request->kelas,  ]);  $akun\_mhs = AkunMahasiswa::create([  'nim' => $request->nim,  'password' => $request->nim,  'status' => '1',  ]);  return !($mhs && $akun\_mhs || null)  ? redirect()->back()->with('errors', 'Gagal tambah mahasiswa')->withInput($request->all())  : redirect()->route('admin.mahasiswa')->with('success', 'Berhasil tambah data mahasiswa');  } |

Kode Program Pengajuan Tugas Akhir *User* Mahasiswa

|  |
| --- |
| public function store(Request $request){  $mhs\_id = Auth::guard('mahasiswa')->user()->id;  $mahasiswa = DataMahasiswa::with('AkunMahasiswa')->find($mhs\_id);  //dd($request);  $filenameWithExt = $request->file('file\_proposal')->getClientOriginalName();  // Get just filename  $filename = pathinfo($filenameWithExt, PATHINFO\_FILENAME);  // Get just ext  $extension = $request->file('file\_proposal')->getClientOriginalExtension();  //Filename to store  $fileNameToStore = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_PROPOSAL\_TA'.'.'.$extension;  if(is\_file('PROPOSAL/'.$fileNameToStore)){  Storage::delete($fileNameToStore);  return redirect()->back();  }  $isSaved = Storage::disk('local')->putFileAs('PROPOSAL', $request->file\_proposal, $fileNameToStore);  if(!$isSaved){  Storage::delete($fileNameToStore);  return redirect()->back()->with('warning', 'Gagal simpan file berkas')->withInput($request->all());  }  $messages = [  'required' => ':attribute wajib diisi',  'max' => ':attribute maksimal :max karakter',  ];  $this->validate($request, [  'email' => 'required',  'judul' => 'required|max:191',  'nama\_dosen' => 'required',  'progress' => 'required',  'file\_proposal' => 'required',  ], $messages);  $proposal = Proposal::create([  'mhs\_id' => $request->mhs\_id,  'dosen\_id' => $request->nama\_dosen,  'email' => $request->email,  'judul' => $request->judul,  'progress' => $request->progress,  'other' => $request->other,  'file' => $isSaved,  'status' => '0',  ]);  $judul = JudulTA::create([  'mhs\_id' => $request->mhs\_id,  'judul' => $request->judul,  ]);  return !($proposal && $judul || null)  ? redirect()->back()->with('errors', 'Gagal upload proposal')->withInput($request->all())  : redirect()->route('mhs.ta')->with('success', 'Berhasil upload proposal');  } |

Kode Program Pemilihan Pembimbingan *User* Koordinator

|  |
| --- |
| public function update(Request $request)  {  $messages = [  'required' => 'calon pembimbing wajib diisi',  ];  $this->validate($request, [  'dosen\_id' => 'required',  'dosens\_id' => 'required',  ], $messages);  $updateTAMahasiswa = Proposal::with('datadosen', 'mahasiswa')->findOrFail($request->id);  $updateTAMahasiswa->update($request->all());  return !($updateTAMahasiswa || null)  ? redirect()->back()->with('warning', 'Gagal ubah pengajuan tugas akhir mahasiswa')->withInput($request->all())  : redirect()->route('dosen.ta.mhs')->with('success', 'Berhasil ubah tugas akhir mahasiswa');  } |

Kode Program Setujui atau Tolak Pengajuan Tugas Akhir *User* Dosen

|  |
| --- |
| public function updatebimbingan(Request $request)  {  Proposal::with('datadosen', 'mahasiswa')->findOrFail($request->id);  if($request->status == 0){ //blm acc  $status\_fix = 1; //menunggu acc pembimbing lain  }else if($request->status == 1){  $status\_fix = 2; //acc  }else if($request->status == 2){  $status\_fix = 3; //ditolak  }else {  $status\_fix = 0;  }  $konfirmasi = Proposal::findOrFail($request->id)->where('id',$request->id)->update(['status' => $status\_fix]);  return back()->with('success', 'Berhasil setujui proposal TA');  } |

Kode Program Pengajuan Sidang TA *User* Mahasiswa

|  |
| --- |
| public function uploadLaporan(Request $request){  $mhs\_id = Auth::guard('mahasiswa')->user()->id;  $mahasiswa = DataMahasiswa::with('AkunMahasiswa')->find($mhs\_id);  //dd($request);  $messages = [  'required' => ':attribute wajib diisi',  ];  $this->validate($request, [  'proposal' => 'required',  'bab1' => 'required',  'bab2' => 'required',  'bab3' => 'required',  'bab4' => 'required',  'bab5' => 'required',  'file\_lain' => 'required',  ], $messages);  // BAB 1  $bab1\_extension = $request->file('bab1')->getClientOriginalExtension();  //Filename to store  $fileNameBab1 = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_BAB 1\_LAPORAN TA'.'.'.$bab1\_extension;  if(is\_file('LAPORAN/'.$fileNameBab1)){  Storage::delete($fileNameBab1);  }  $bab1isSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->bab1, $fileNameBab1);  if(!$bab1isSaved){  Storage::delete($fileNameBab1);  return redirect()->back()->with('warning', 'Gagal simpan file laporan tugas akhir')->withInput($request->all());  }  // BAB 2  $bab2\_extension = $request->file('bab2')->getClientOriginalExtension();  //Filename to store  $fileNameBab2 = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_BAB 2\_LAPORAN TA'.'.'.$bab2\_extension;  if(is\_file('LAPORAN/'.$fileNameBab2)){  Storage::delete($fileNameBab2);  }  $bab2isSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->bab2, $fileNameBab2);  if(!$bab2isSaved){  Storage::delete($fileNameBab2);  return redirect()->back()->with('warning', 'Gagal simpan file laporan tugas akhir')->withInput($request->all());  }  // BAB 3  $bab3\_extension = $request->file('bab3')->getClientOriginalExtension();  //Filename to store  $fileNameBab3 = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_BAB 3\_LAPORAN TA'.'.'.$bab3\_extension;  if(is\_file('LAPORAN/'.$fileNameBab3)){  Storage::delete($fileNameBab3);  }  $bab3isSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->bab3, $fileNameBab3);  if(!$bab3isSaved){  Storage::delete($fileNameBab3);  return redirect()->back()->with('warning', 'Gagal simpan file laporan tugas akhir')->withInput($request->all());  }  // BAB 5  $bab4\_extension = $request->file('bab4')->getClientOriginalExtension();  //Filename to store  $fileNameBab4 = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_BAB 4\_LAPORAN TA'.'.'.$bab4\_extension;  if(is\_file('LAPORAN/'.$fileNameBab4)){  Storage::delete($fileNameBab4);  }  $bab4isSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->bab4, $fileNameBab4);  if(!$bab4isSaved){  Storage::delete($fileNameBab4);  return redirect()->back()->with('warning', 'Gagal simpan file laporan tugas akhir')->withInput($request->all());  }  // BAB 5  $bab5\_extension = $request->file('bab5')->getClientOriginalExtension();  //Filename to store  $fileNameBab5 = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_BAB 5\_LAPORAN TA'.'.'.$bab5\_extension;  if(is\_file('LAPORAN/'.$fileNameBab5)){  Storage::delete($fileNameBab5);  }  $bab5isSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->bab5, $fileNameBab5);  if(!$bab5isSaved){  Storage::delete($fileNameBab5);  return redirect()->back()->with('warning', 'Gagal simpan file laporan tugas akhir')->withInput($request->all());  }  //File lain  $file\_lain\_extension = $request->file('file\_lain')->getClientOriginalExtension();  //Filename to store  $fileNameFileLain = $mahasiswa->nim.'\_'.$mahasiswa->nama.'\_PERSYARATAN\_SIDANG\_TA'.'.'.$file\_lain\_extension;  if(is\_file('LAPORAN/'.$fileNameFileLain)){  Storage::delete($fileNameFileLain);  }  $fileLainisSaved = Storage::disk('local')->putFileAs('LAPORAN', $request->file\_lain, $fileNameFileLain);  if(!$fileLainisSaved){  Storage::delete($fileNameFileLain);  return redirect()->back()->with('warning', 'Gagal simpan file persyaratan sidang tugas akhir')->withInput($request->all());  }  $sidang = SidangTA::create([  'mhs\_id' => $request->mhs\_id,  'proposal\_id' => $request->proposal,  'bab1' => $bab1isSaved,  'bab2' => $bab2isSaved,  'bab3' => $bab3isSaved,  'bab4' => $bab4isSaved,  'bab5' => $bab5isSaved,  'file\_lain' => $fileLainisSaved,  'status' => '0',  ]);  return !($sidang || null)  ? redirect()->back()->with('errors', 'Gagal melakukan pengajuan sidang tugas akhir')->withInput($request->all())  : redirect()->route('mhs.sidang')->with('success', 'Berhasil melakukan pengajuan sidang tugas akhir');  } |

Kode Program Setujui Sidang TA *User* Dosen

|  |
| --- |
| public function status(Request $request)  {  // dd($request->all());  if($request->status == 0){  $status\_fix = 1;  }else if($request->status == 1){  $status\_fix = 2;  }else {  $status\_fix = 0;  }  $konfirmasi = SidangTA::findOrFail($request->id)->where('id',$request->id)->update(['status' => $status\_fix]);  } |

Kode Program Konfirmasi Pengajuan Sidang TA *User* Administrator

|  |
| --- |
| public function konfirmasiSidang(Request $request)  {  // dd($request->all());  $konfirmasi = SidangTA::findOrFail($request->id);  $konfirmasi->update($request->all());  return back()->with('success', 'Berhasil konfirmasi pengajuan sidang TA');  } |

Kode Program Tambah Jadwal Sidang TA *User* Koordinator

|  |
| --- |
| public function store(Request $request){  // dd($request);  $messages = [  'required' => ':attribute wajib diisi',  ];  $this->validate($request, [  'tgl\_sidang'=> 'required',  'waktu' => 'required',  'penguji1' => 'required',  'penguji2' => 'required',  'penguji3' => 'required',  ], $messages);  $jadwal\_sidang = JadwalSidang::create([  'mhs\_id' => $request->mhs\_id,  'proposal\_id' => $request->proposal\_id,  'sidang\_id' => $request->id,  'tgl\_sidang' => $request->tgl\_sidang,  'waktu' => $request->waktu,  'penguji1' => $request->penguji1,  'penguji2' => $request->penguji2,  'penguji3' => $request->penguji3,  ]);  return !($jadwal\_sidang || null)  ? redirect()->back()->with('errors', 'Gagal tambah jadwal sidang tugas akhir')->withInput($request->all())  : redirect()->route('dosen.sidang')->with('success', 'Berhasil tambah jadwal sidang tugas akhir');  } |

**LAMPIRAN 4 BIODATA PENULIS**

**BIODATA PENULIS TUGAS AKHIR**

|  |  |  |
| --- | --- | --- |
| Nama | : Amelia Syamsa Mayangsari |  |
| Nomor Induk Mahasiswa | : 1603034 |
| Jenis Kelamin | : Perempuan |
| Tempat, Tanggal Lahir | : Indramayu, 10 Mei 1999 |
| Status | : Belum Menikah |
| Agama | : Islam |
| Asal Ijazah Sekolah | : Nama Sekolah Kota Sekolah | Tahun Ijazah |
| SD | :-SDN Bugis I -Indramayu | -2010 |
| SMP | :-MTs N Anjatan -Indramayu | -2013 |
| SMA | :-SMK Al – Huda -Indramayu | -2016 |
|  | Kedungwungu Anjatan |  |
| Program Studi/Jurusan | : Teknik Informatika |  |

Alamat Rumah Asal : Dusun Cilempung Desa Bugis, Kec. Anjatan, Kab. Indramayu, 45256.

No. Handphone : 082120940701

Email : ameliasyamsa1309@gmail.com

Nama Orang Tua : Muchidin - Ni Made Rai Sumartini

Pekerjaan Orang Tua : Wiraswasta

|  |  |
| --- | --- |
| Judul Tugas Akhir | : Sistem Informasi Tugas Akhir Berbasis Web  Menggunakan Framework Laravel. |

Pembimbing : 1. Eka Ismantohadi, S.Kom., M.Eng

2. Muhamad Mustamiin, S.Pd., M.Kom

Indramayu, 15 Agustus 2019

Penulis,

Amelia Syamsa Mayangsari

NIM.1603034