**LAPORAN AKHIR TUGAS BESAR MANAJEMEN BASIS DATA**

**Website Database Administrasi Pengerjaan Tugas Akhir Program Studi Teknologi Informasi**

*Dosen Pengampu:*

**Dedy Arisandi , S.T.,M.Kom.**

**DISUSUN OLEH :**

**KOM B – KELOMPOK 5**

Karina Angela Tobing 211402041

Gery Jonathan Manurung 211402137

Sarmida Uli Sinaga 211402071

Ridho P. Sibuea 211402014

Ferdy Fraddly Aldy Nainggolan 211402026

Yoel Mulya Raja Siregar 211402149

**UNIVERSITAS SUMATERA UTARA**

**FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI**

**PROGRAM STUDI TEKNOLOGI INFORMASI**

**T.A. 2022**

# KATA PENGANTAR

Puji dan syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, karena oleh karena berkat dan karunianya kami dapat menyelesaikan laporan akhir Tugas Besar yang berjudul “Website Database Administrasi Pengerjaan Tugas Akhir Prodi Teknologi Informasi Universitas Sumatera Utara” untuk memenuhi tugas besar mata kuliah Manajemen Sistem Basis Data.

Tidak lupa kami mengucapkan terima kasih kepada dosen mata kuliah Sistem Basis Data, Pak Dedy Arisandi, ST., M.Kom dan pendamping Laboratorium Praktikum Sistem Basis Data, Andrea Vicalina serta rekan – rekan yang telah membantu dan membimbing kami dalam pengerjaan tugas besar ini. Kami juga mengucapkan terima kasih kepada pihak-pihak yang terlibat dalam pembuatan laporan akhir ini.

Kami memohon maaf apabila dalam laporan ini terdapat kekurangan dan kesalahan dalam hal penyusunan maupun kosakata yang tidak digunakan secara tepat. Oleh karena itu kritik dan saran yang membangun sangat kami harapkan demi kesempurnaan tugas besar ini.

Medan,03 Desember 2022

Kelompok 5

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc124001947)

[DAFTAR ISI ii](#_Toc124001948)

[BAB I 1](#_Toc124001949)

[PENDAHULUAN 1](#_Toc124001950)

[**1.1 Latar Belakang** 1](#_Toc124001951)

[**1.2 Rumusan Masalah** 1](#_Toc124001952)

[**1.3 Batasan Masalah** 2](#_Toc124001953)

[**1.4 Tujuan** 2](#_Toc124001954)

[**1.5 Manfaat** 2](#_Toc124001955)

[BAB II 4](#_Toc124001956)

[ANALISA DAN PERANCANGAN 4](#_Toc124001957)

[**2.1 User Requirement** 4](#_Toc124001958)

[**2.2 Dokumen Dasar** 7](#_Toc124001959)

[**2.3 Flowchart Pengajuan Judul DAN Seminar Proposal Tugas Akhir (TA)** 25](#_Toc124001960)

[**2.4 Flowchart Seminar Hasil dan Sidang Meja Hijau** 27](#_Toc124001961)

[**2.5 Use case diagram** 28](#_Toc124001962)

[**2.6 Daftar Tabel dan View** 31](#_Toc124001963)

[**2.7 Relasi Antar Tabel** 45](#_Toc124001964)

[**2.8 Daftar Trigger** 45](#_Toc124001965)

[**2.9 Daftar Prosedur** 46](#_Toc124001966)

[**2.10 Daftar User dan Privileges** 60](#_Toc124001967)

[BAB III 62](#_Toc124001968)

[IMPLEMENTASI SISTEM 62](#_Toc124001969)

[**3.1 Kebutuhan Perangkat Keras dan Lunak** 62](#_Toc124001970)

[**3.2 Fitur , Subfitur dan Keterangan** 62](#_Toc124001971)

[**3.3 Implementasi Sistem** 65](#_Toc124001972)

[BAB IV 76](#_Toc124001973)

[PENUTUP 76](#_Toc124001974)

[**4.1 Kesimpulan** 76](#_Toc124001975)

[**4.2 Saran** 76](#_Toc124001976)

[DAFTAR PUSTAKA 77](#_Toc124001977)

# BAB I

# PENDAHULUAN

## **1.1 Latar Belakang**

Menjadi salah satu universitas terbaik di Indonesia , Universitas Sumatera Utara memiliki visi menjadi perguruan tinggi yang memiliki keunggulan akademik sebagai barometer kemajuan ilmu pengetahuan yang mampu bersaing dalam tataran global. Visi ini kemudian ditranslasikan ke dalam misi dan diwujudkan dalam struktur kurikulum pembelajaran. Kurikulum pembelajaran setiap program studi di universitas ini berbeda-beda. Namun, untuk menyelesaikan masa pembelajaran, skripsi diselenggarakan untuk mengevaluasi penguasaan mahasiswa terhadap kompetensi lulusan setiap program studinya.

Kegiatan tugas akhir yang diikuti oleh mahasiswa di Universitas Sumatera Utara setiap semesternya selama ini dilakukan tidak praktis dan kurang fleksibel. Hal ini dikarenakan beberapa kegiatan administrasi masih dilakukan secara manual. Data yang masih bersifat manual atau analog membuat pengolahan, analisis dan evaluasi data menjadi sulit untuk dilakukan. Kesulitan ini lebih dirasakan pada saat monitoring/evaluasi pembelajaran maupun akreditasi seperti tenggat pengumpulan berkas, tanggal penentuan sidang, jumlah mahasiswa bimbingan dosen tersebut, dan lain-lain , apalagi karena selama ini kampus masih menggunakan e-learning sebagai media untuk pengerjaan tugas akhir mahasiswa . Hal ini dirasa sangat tidak efektif karena data-data dari setiap mahasiswa harus di sortir secara manual satu per satu .Untuk itu database ini dihadirkan agar mampu menjadi wadah bagi pengerjaan tugas akhir mahasiswa dari proses awal hingga wisuda , diharapkan dengan adanya database ini proses pengumpulan dokumen dari mahasiswa akhir bisa dilakukan dengan lebih singkat dan mempercepat proses dari Tugas Akhir mahasiswa Teknologi Informasi.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan rumusan masalah dalam penulisan ini adalah sebagai berikut.

1. Pengerjaan tugas akhir mahasiswa masih di lakukan secara manual
2. Berkas yang berisiko rusak atau hilang jika pengerjaan tugas akhir dilakukan dengan manual
3. Mahasiswa yang ingin mengerjakan tugas akhir masih harus menggunakan e-learning , sehingga data ataupun berkas mahasiswa susah di kumpulkan

## **1.3 Batasan Masalah**

Berdasarkan rumusan masalah yang telah dipaparkan, batasan masalah dalam penulisan ini adalah sebagai berikut.

1. Seminar proposal, seminar hasil, dan sidang meja hijau dilakukan di luar web.
2. Bimbingan skripsi antara mahasiswa dan dosen dilakukan di luar aplikasi , namun akan ada catatan setiap bimbingan bersama dengan dosen.
3. Pengelolaan database administrasi skripsi hanya mencakup mahasiswa tingkat akhir program studi Teknologi Informasi saja.

## **1.4 Tujuan**

Adapun tujuan dari pembuatan website pengerjaan tugas akhir berbasis aplikasi ini adalah sebagai berikut,

1. Mempermudah proses pengerjaan tugas akhir untuk mahasiswa akhir Teknologi Informasi.
2. Meminimalisir kerusakan atau kehilangan berkas selama kegiatan berlangsung.
3. Meningkatkan kualitas dan sistem program studi sehingga dapat memenuhi visi dan misi program studi.
4. Memenuhi tugas besar mata kuliah Manajemen Sistem Basis Data

## **1.5 Manfaat**

Adapun manfaat dengan adanya sistem ini adalah sebagai berikut :

1. Bagi Program Studi
   * Meningkatkan kualitas program studi sehingga dapat memenuhi visi dan misi program studi.
2. Bagi Masyarakat Teknologi Informasi
   * Menciptakan kepraktisan dan kemudahan bagi masyarakat Teknologi Informasi yang mengerjakan tugas akhir
   * Menghindari terjadinya kerusakan dan kehilangan file selama masa kegiatan berlangsung
3. Bagi kelompok 5 (developer)
   * Meningkatkan keterampilan dan skill mahasiswa dalam mengembangkan aplikasi berbasis web yang mengimplementasikan DBMS.
   * Meletakkan hasil akhir website sebagai bentuk potofolio jika aplikasi ini kelak digunakan seperti tujuan pembuatannya.

# BAB II

# ANALISA DAN PERANCANGAN

## **2.1 User Requirement**

User requirement merupakan hal utama dalam membangun website. Analisa dan rancangan dari user requirement biasanya menganalisis role yang digunakan dan perannya dalam sistem tersebut. Dalam sistem ini, user membutuhkan beberapa proses yang perlu dilakukan seperti elimination, simplification, integration, dan automation untuk meningkatkan kinerja dari proses setiap user.

1. Mahasiswa
   1. Mahasiswa dapat mencetak dan mengunduh form/file yang dibutuhkan selama masa penyelesaian skripsi, seperti: formulir calon pembimbing, lembar kendali seminar proposal, hasil, dan sidang meja hijau, dan sebagainya.
   2. Mahasiswa dapat melihat jadwal seminar proposal, jadwal seminar hasil, dan jadwal sidang meja hijau.
   3. Mahasiswa dapat mendaftarkan dan memperbaharui judul skripsinya.
   4. Mahasiswa dapat menambahkan foto profil dan memperbaharui data dirinya.
2. Dosen
   1. Dosen dapat melihat data mahasiswa bimbingannya
   2. Dosen dapat melakukan pencarian mahasiswa bimbingan berdasarkan nama, angkatan, progress skripsi, dan sebagainya.
   3. Dosen dapat mencetak dan mengunduh lembar kendali bimbingan mahasiswa
   4. Dosen dapat melihat progress skripsi mahasiswa bimbingannya
   5. Dosen dapat melihat jadwal seminar/sidang mahasiswa bimbingannya.
   6. Dosen dapat memberikan nilai saat seminar dimana dosen tersebut berperan sebagai dosen pembimbing/pembanding dalam seminar tersebut
   7. Dosen dapat memberikan catatan bimbingan bagi mahasiswa bimbingannya
3. Kepala lab
   1. Dapat melihat list mahasiswa yang mengambil peminatan sesuai bidang yang diawasinya
   2. Dapat memberikan nilai kepada hasil executive summary yang diajukan kepadanya
4. Staff /Admin

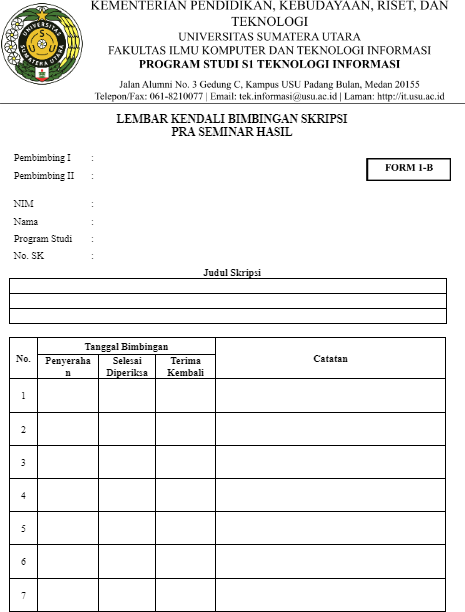
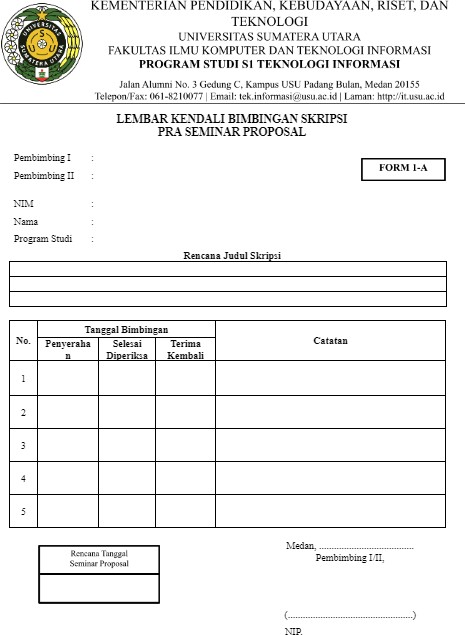
Admin terdiri dari admin dan super admin. Requirement pada admin sama dengan super admin, namun yang membedakan adalah super admin dapat menambahkan, admin baru, mengedit dan menghapus data semua admin , sedangkan admin biasa tidak bisa melakukan hal tersebut . Berikut merupakanuser requirement admin,

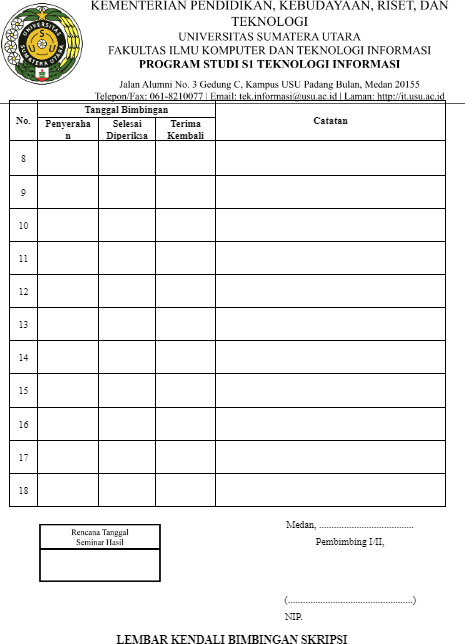
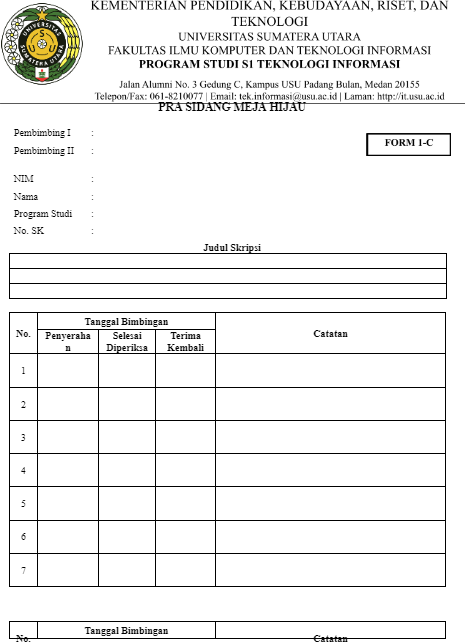
* 1. Admin dapat melihat data mahasiswa bimbingan dan dosen pembimbingnya
  2. Admin dapat melihat data mahasiswa dan dosen pembimbingnya
  3. Admin dapat mencari mahasiswa TA
  4. Admin dapat menambah, mengubah, dan menghapus dosen pembimbing dan judul skripsi mahasiswa.
  5. Admin dapat mencetak dan mengunduh form/file yang dibutuhkan mahasiswa bimbingan seperti: form persetujuan, lembar kendali seminar proposal, hasil, dan sidang meja hijau, dsb.
  6. Admin dapat memverifikasi mahasiswa bimbingan lulus dalam seminar/sidangnya
  7. Admin dapat menambah akun mahasiswa, dosen, admin, dan prodi.
  8. Admin dapat mengubah dan menghapus data mahasiswa dan dosen (pembimbing dan penguji).
  9. Admin dapat menambah, mengubah, dan menghapus jadwal seminar proposal, seminar hasil, dan sidang meja hijau mahasiswa
  10. Admin dapat melihat dan mencari daftar riwayat aktivitas admin
  11. Admin dapat menambahkan foto profil dan memperbaharui data dirinya

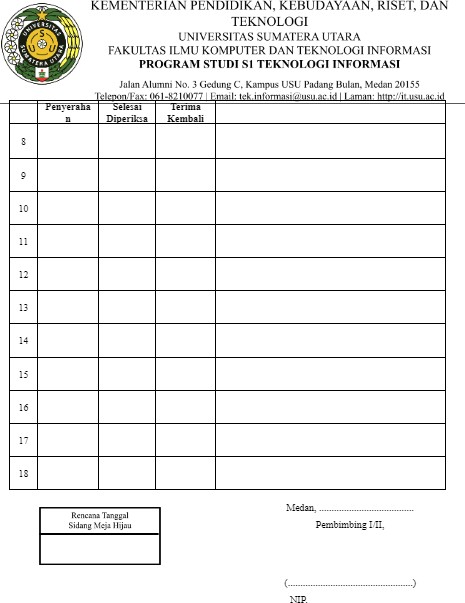
## **2.2 Dokumen Dasar**

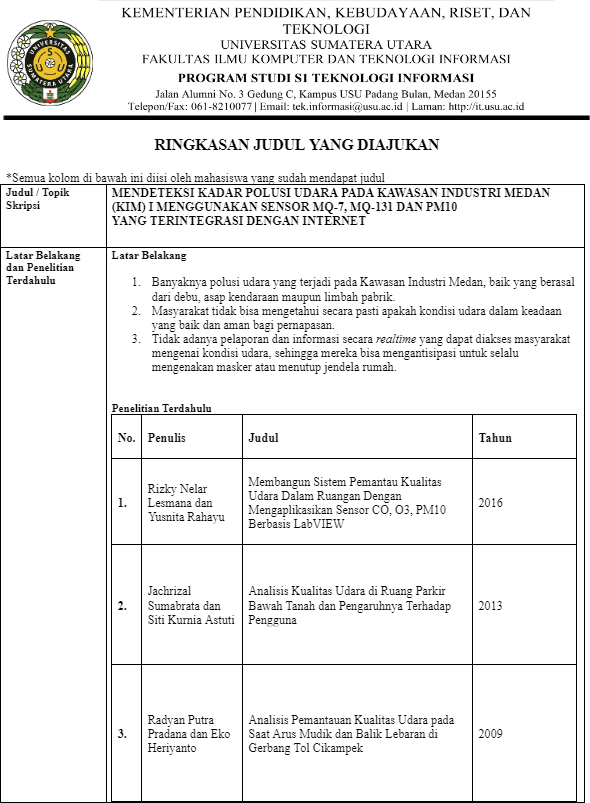
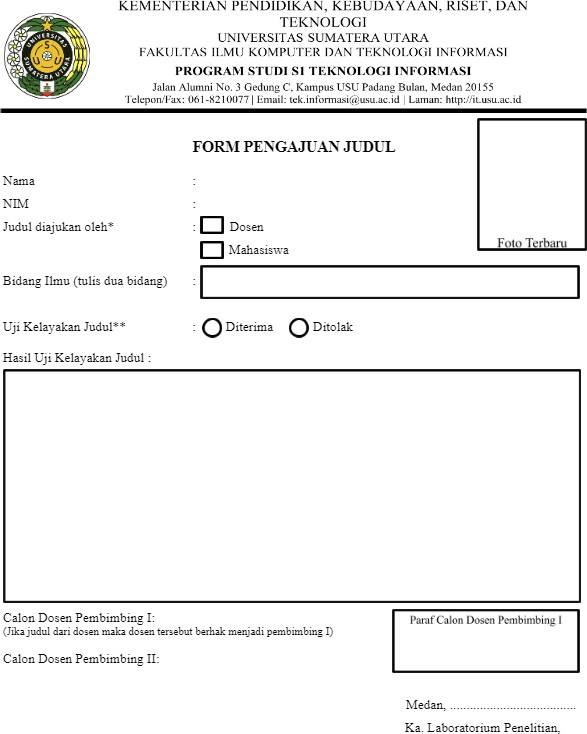
Adapun beberapa dokumen yang kami peroleh sebagai referensi dalam pengerjaan “Sistem Aministrasi Skripsi” adalah berdasarkan sumber yang kami dapat dari Bapak Dedy Arisandi, ST., M.Kom .

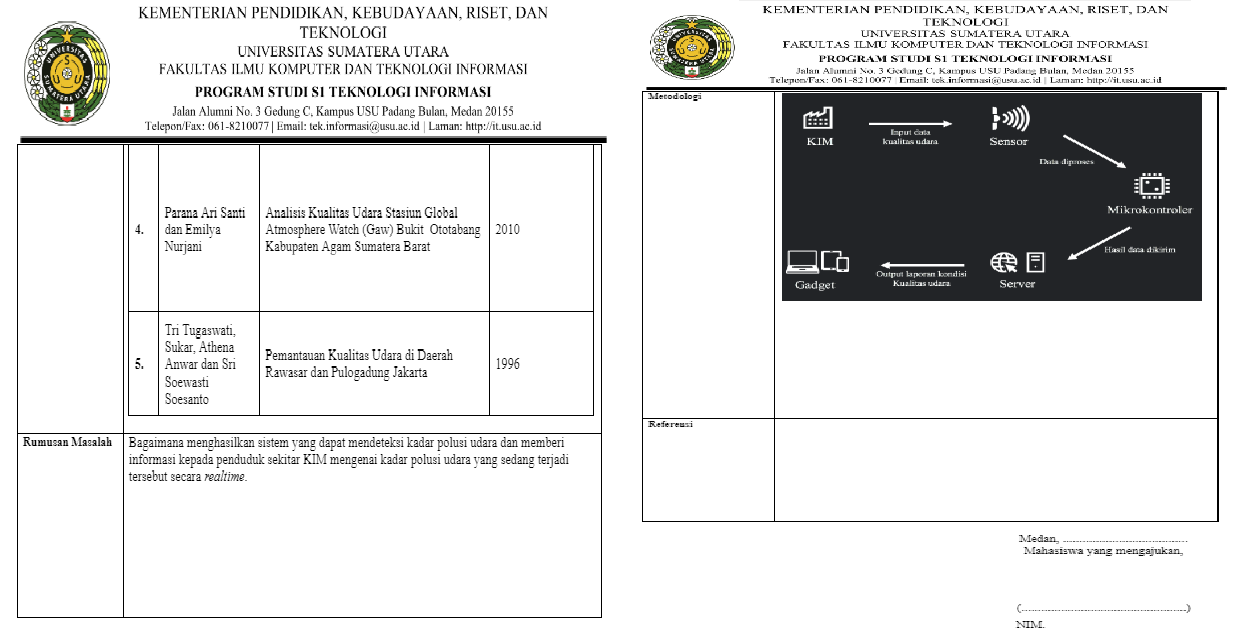
1. Lembar Kendali Proposal – Form 1A



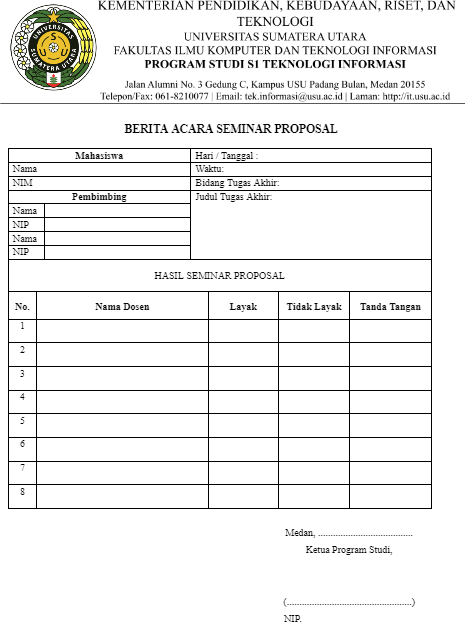




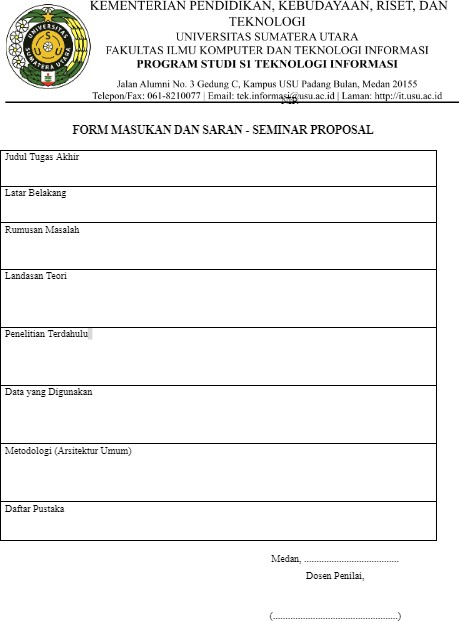
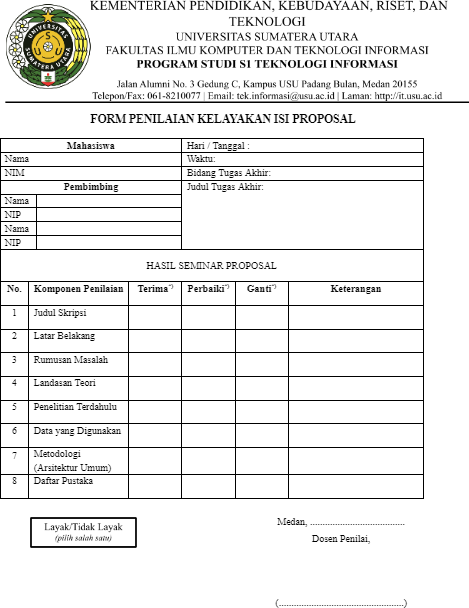
1. Form Pengajuan Judul – Form 2A

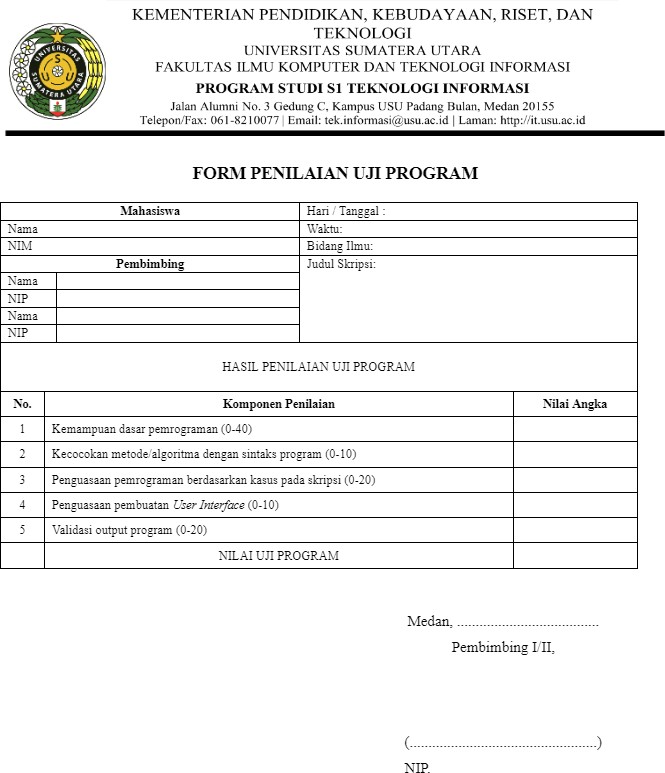


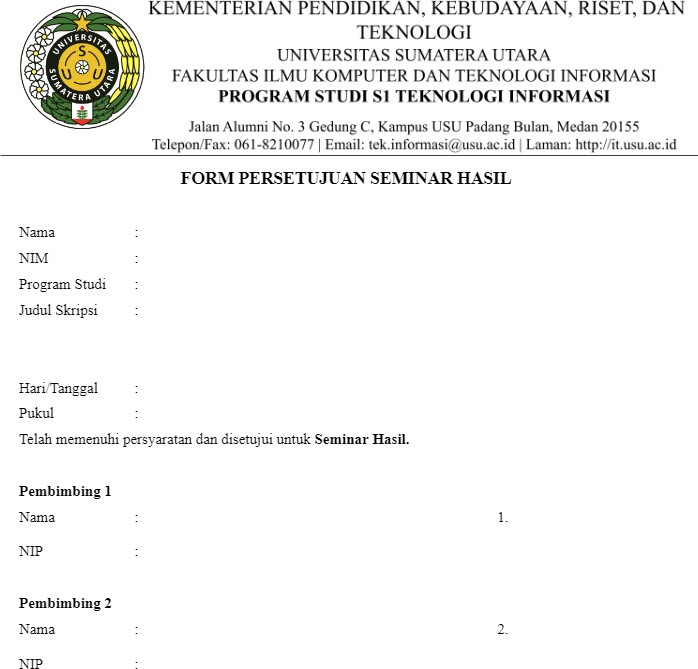
1. Form berita acara seminar proposal

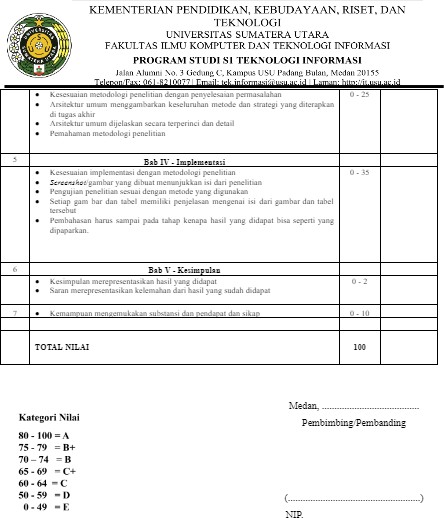
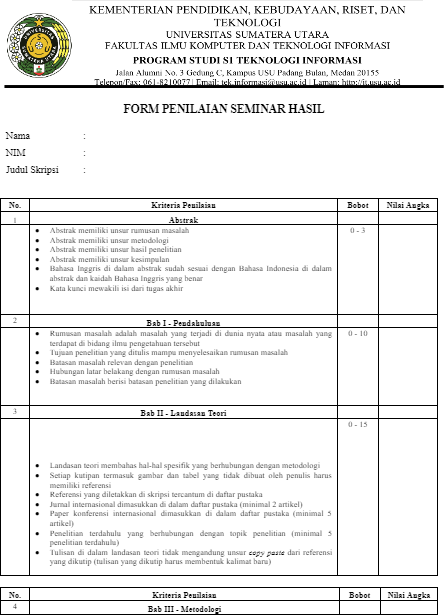


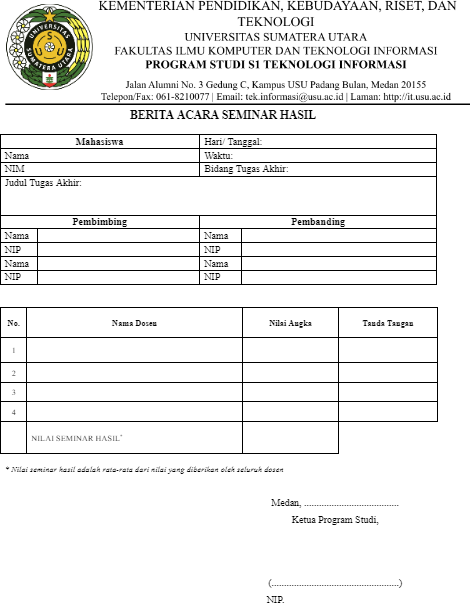
1. Form penilaian kelayakan isi proposal



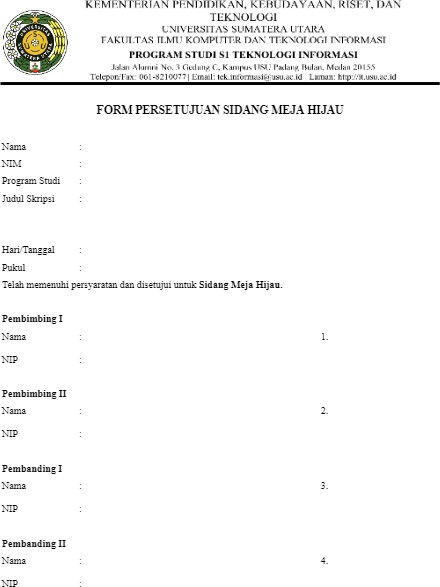
1. Form Penilaian Uji Program – Form 2c
2. Form Persetujuan Seminar Hasil – Form 2D



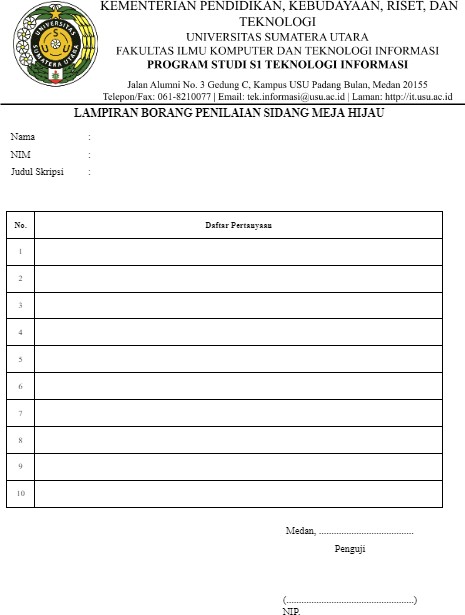
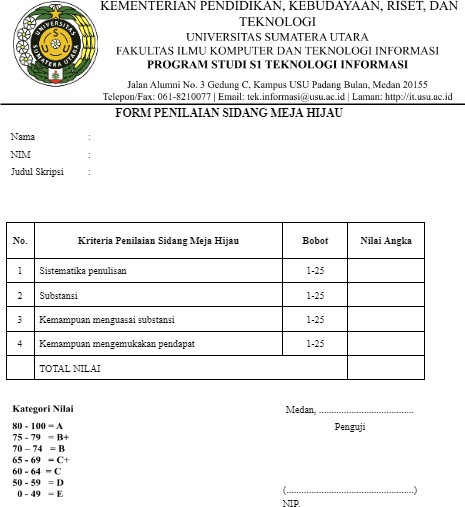
1. Form Penilaian Seminar Hasil – Form 2E
2. Berita Acara Seminar Hasil – Form 2f

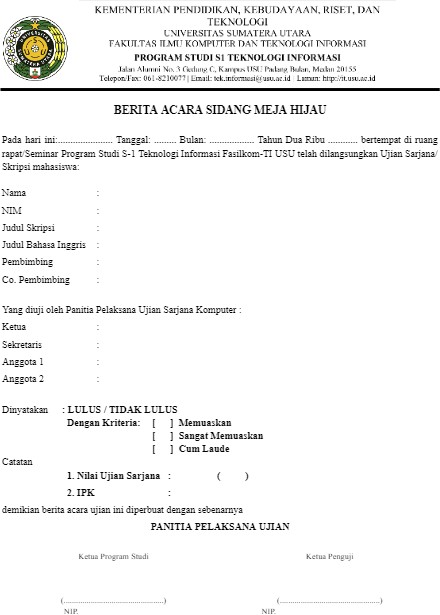
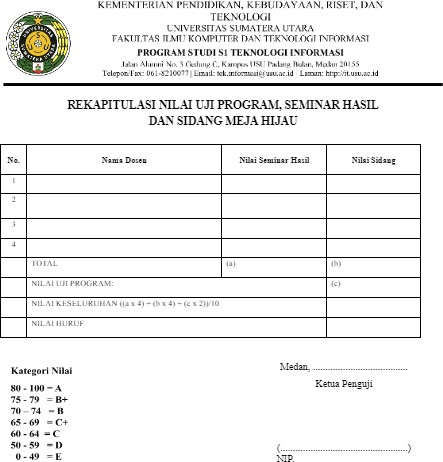


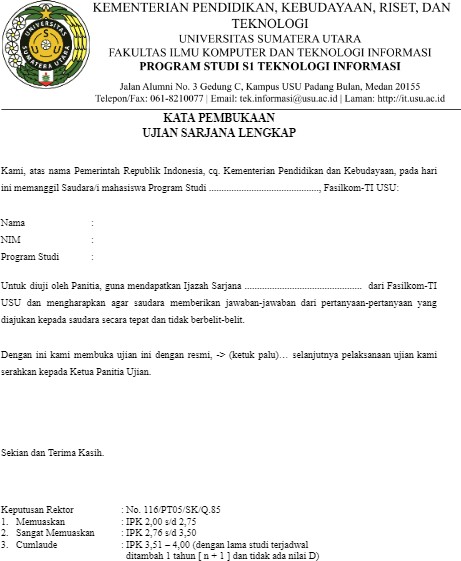
1. Form Persetujuan Sidang Meja Hijau



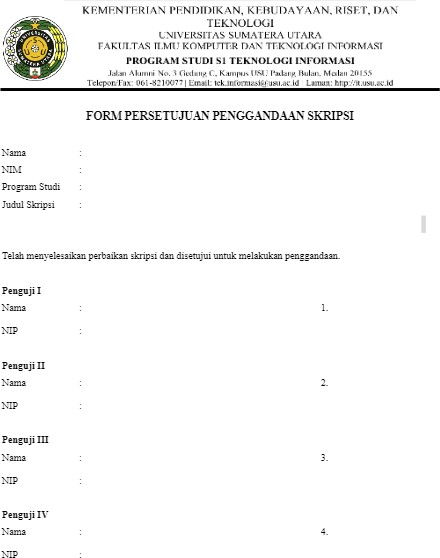
1. Form Penilaian Sidang Meja Hijau – Form 2H



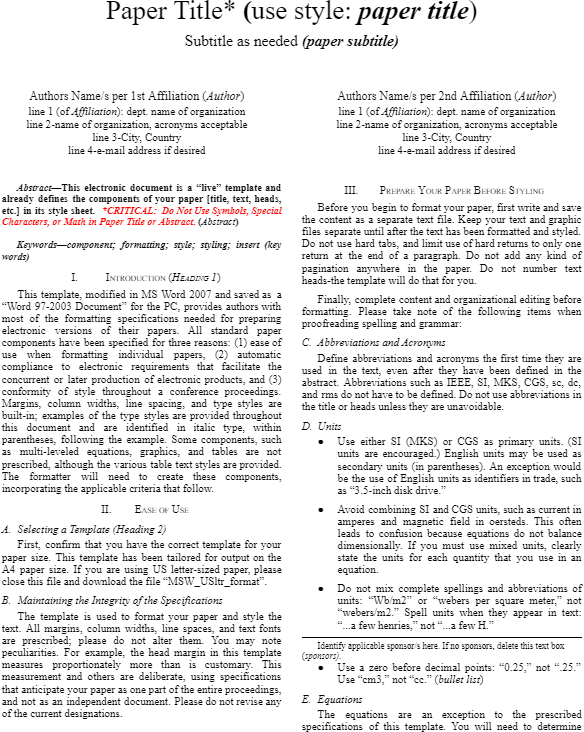
1. Berita Acara Sidang Meja Hijau – Form 2I
2. Kata Pengantar Sidang – Form 2J

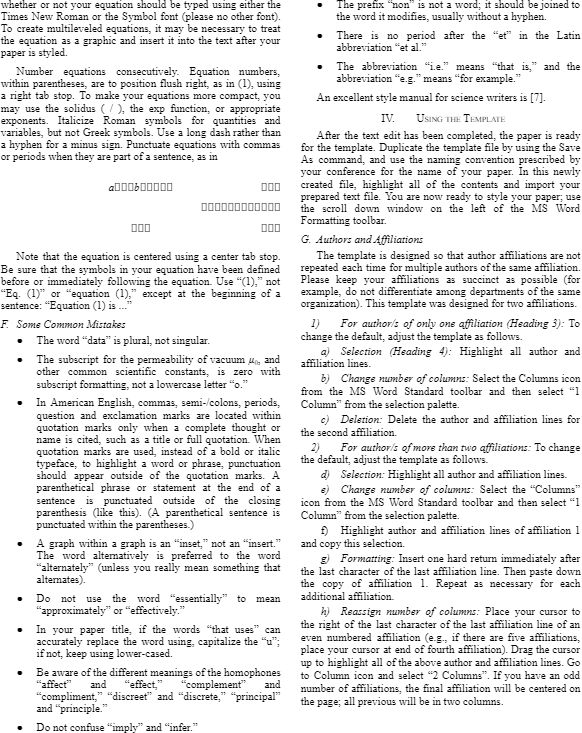
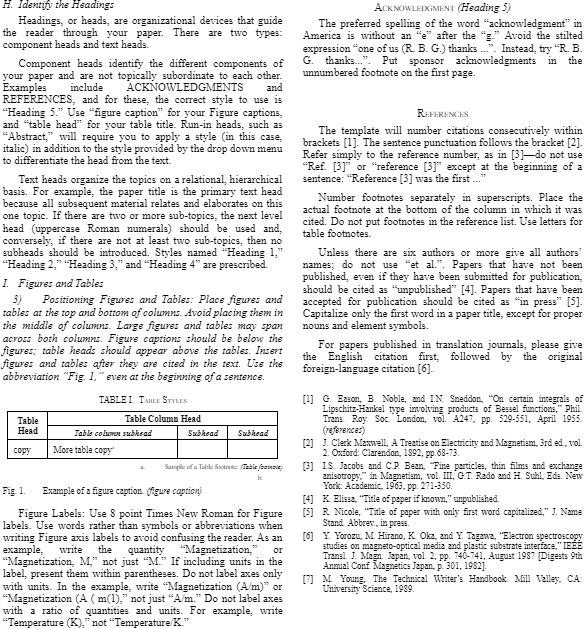


1. Form Persetujuan Penggandaan Skripsi – Form 2K

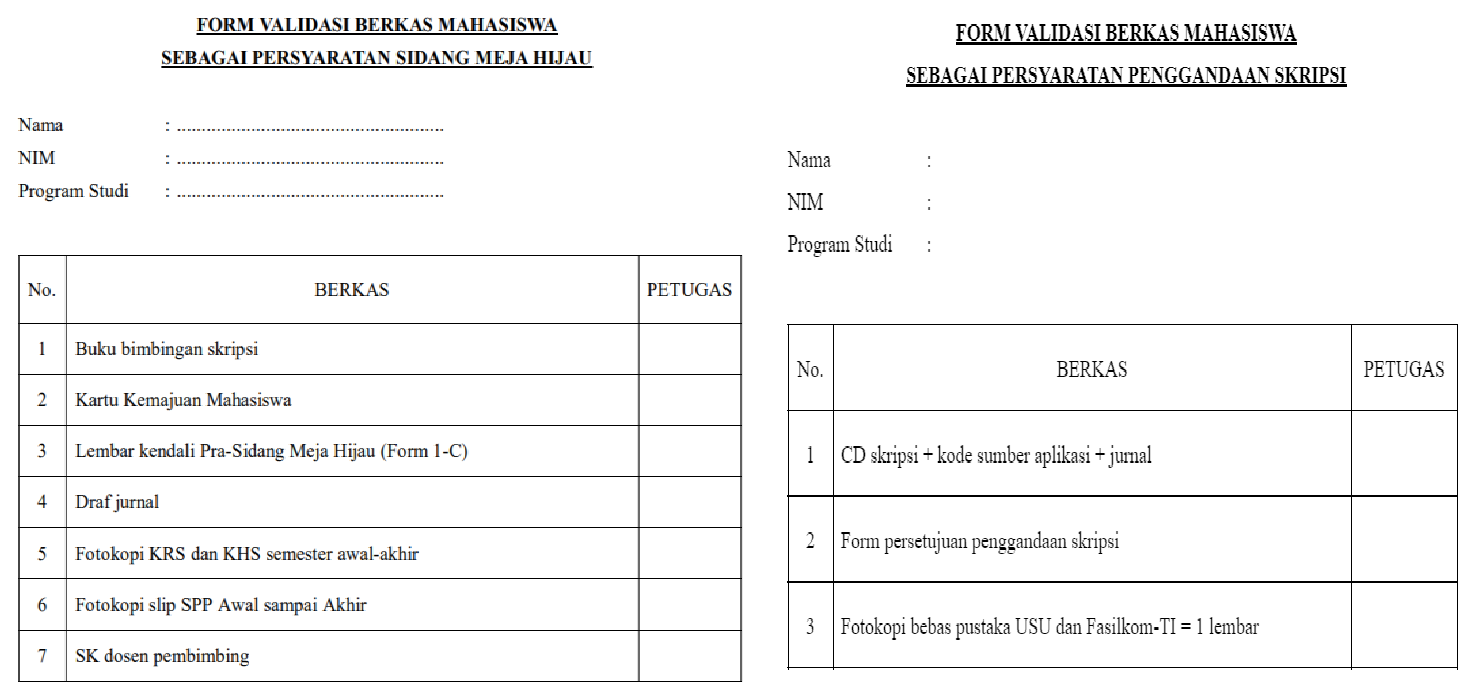
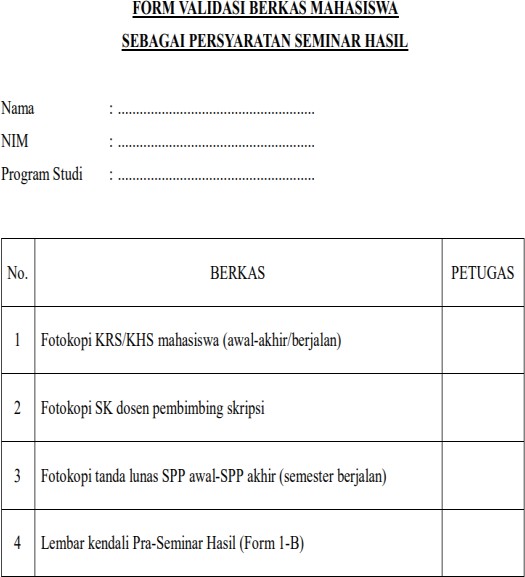
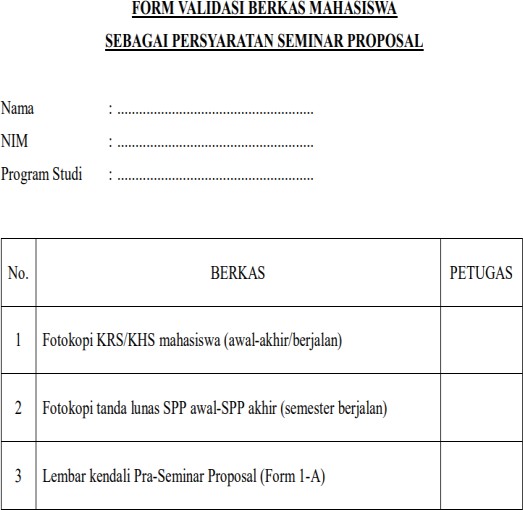


1. Form Jurnal



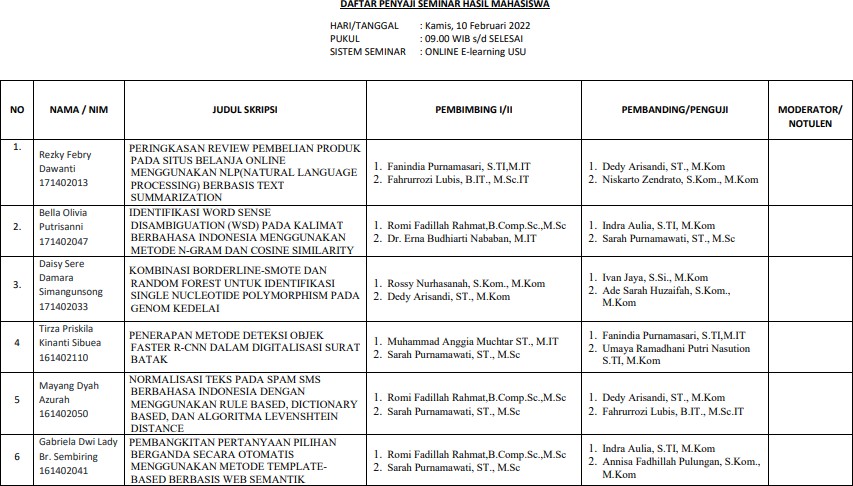
1. Form Validasi Berkas Mahasiswa – Form 3A

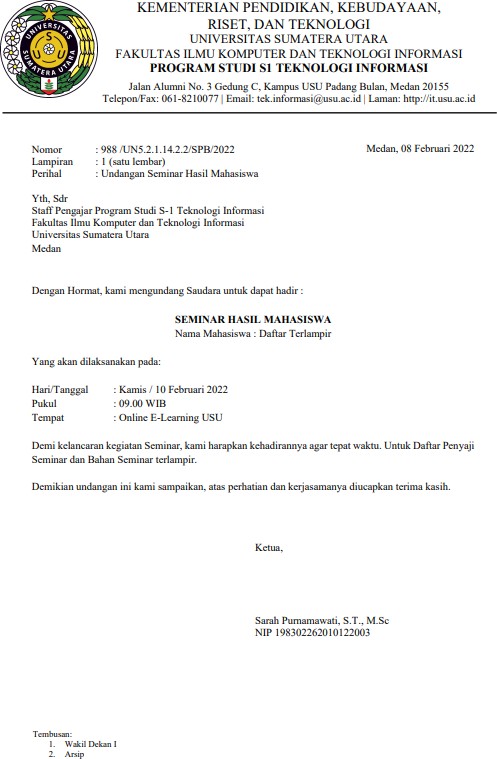


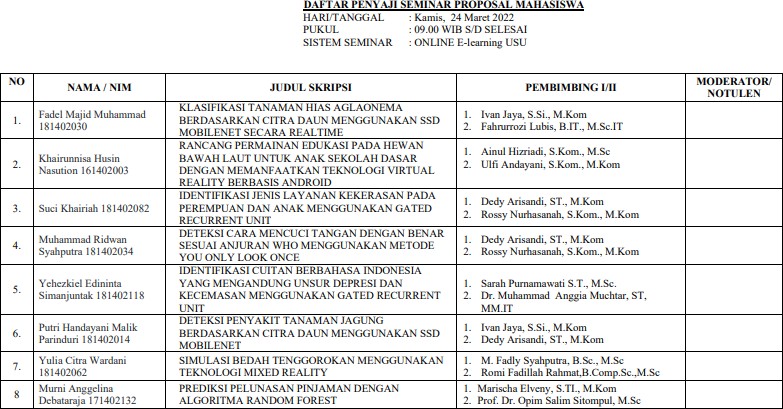
1. Form Calon Pembimbing



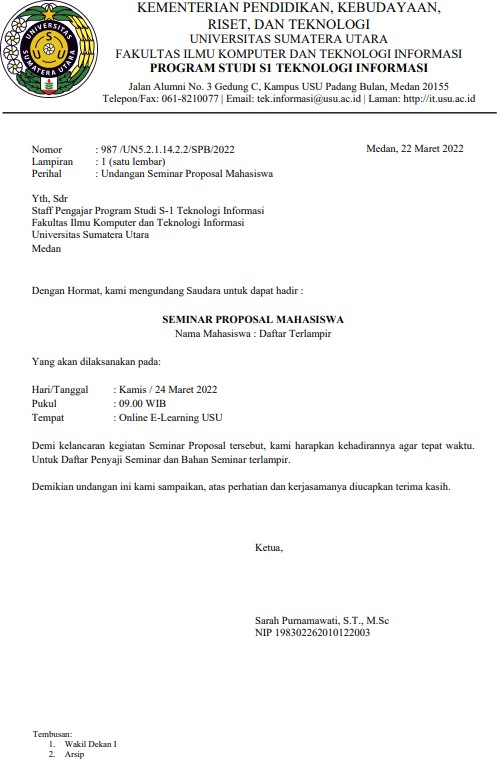
1. Contoh Seminar Hasil – Daftar Peserta



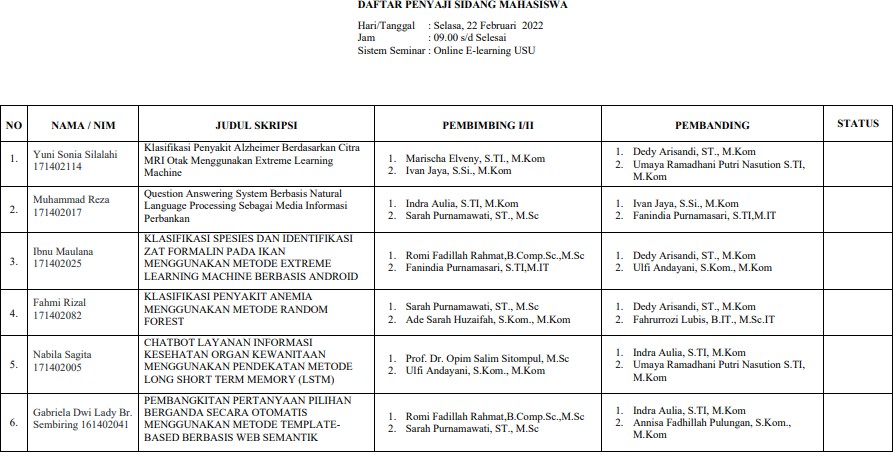
1. Contoh Seminar Hasil – Undangan
2. Contoh Seminar Proposal – Daftar Peserta

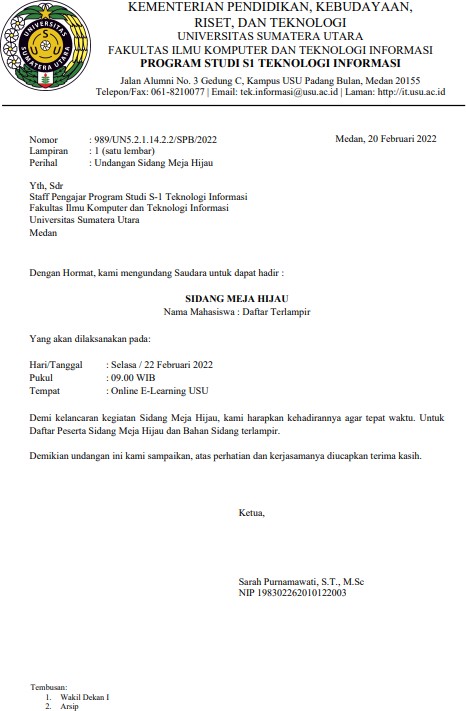


1. Contoh Seminar Proposal

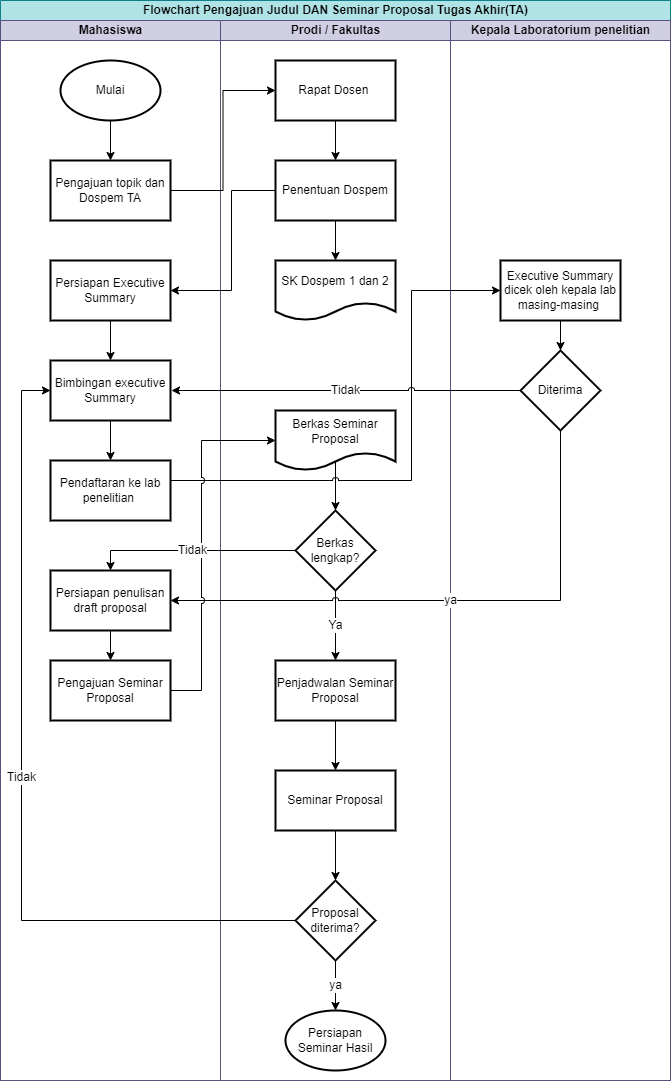


1. Contoh Sidang Meja Hijau – Daftar Peserta

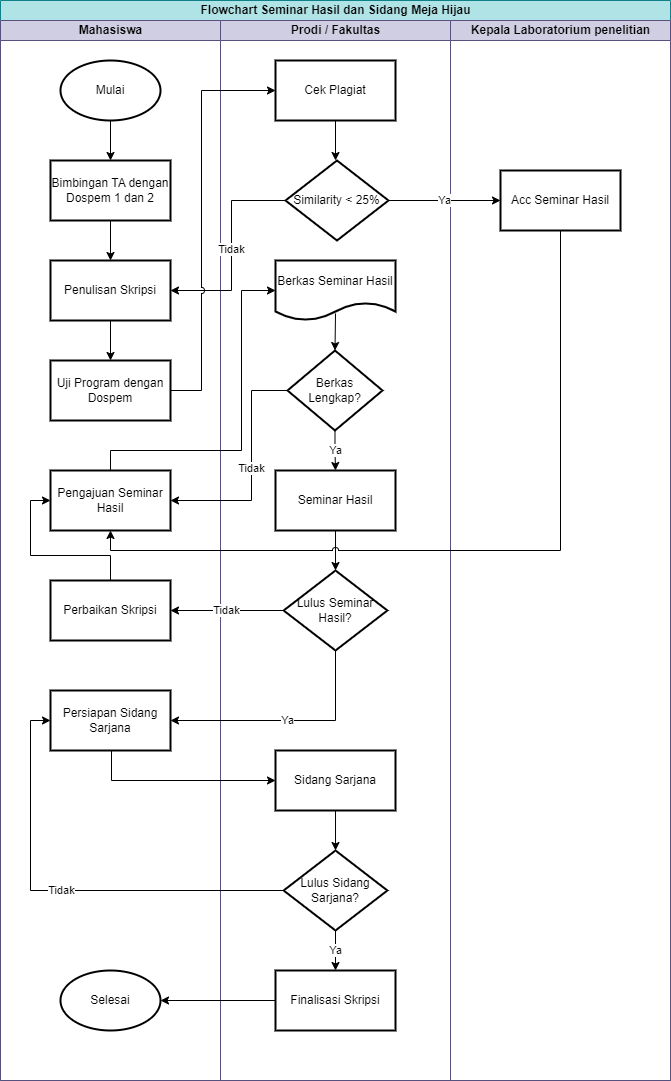


1. Contoh Sidang Meja Hijau – Undangan

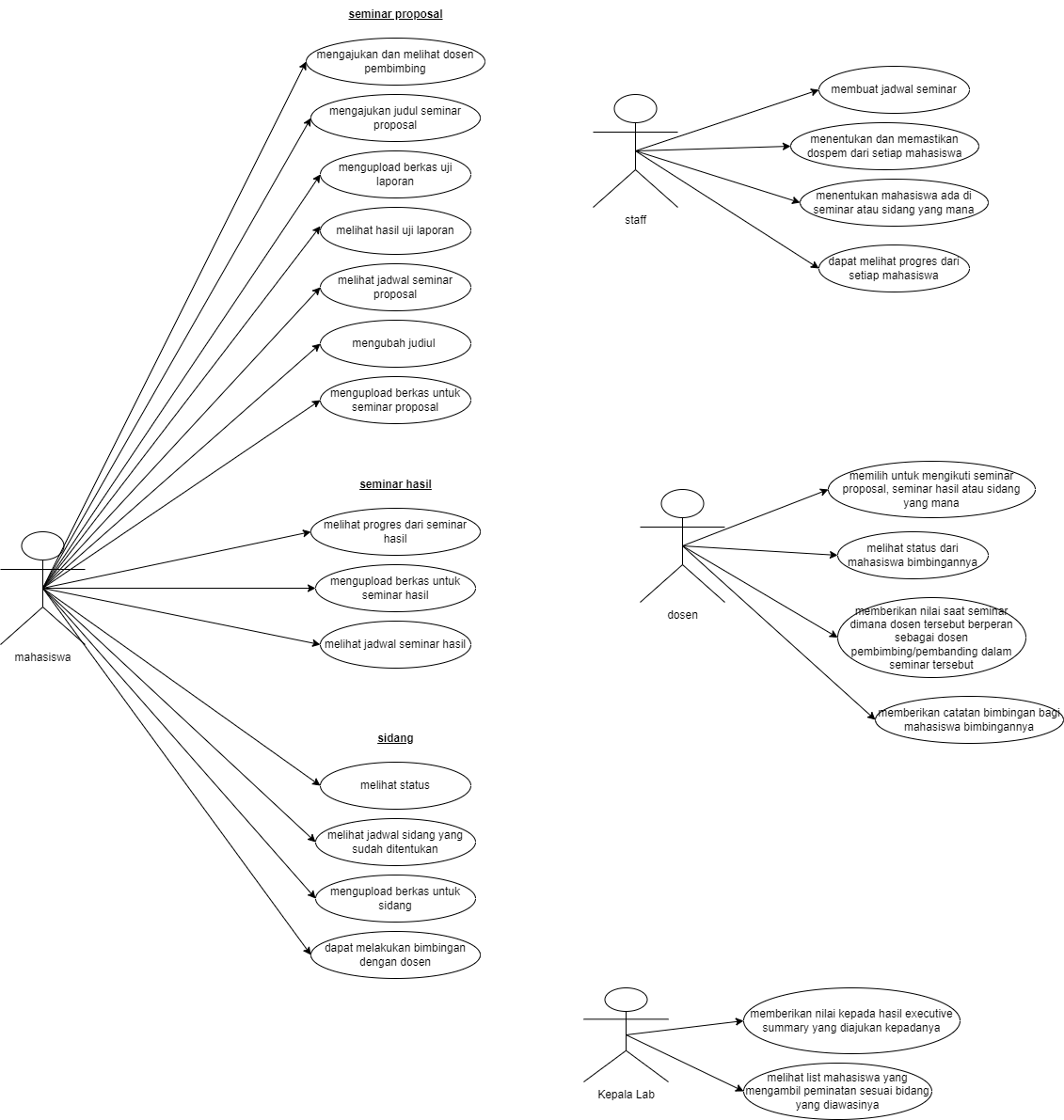
## **2.3 Flowchart Pengajuan Judul DAN Seminar Proposal Tugas Akhir (TA)**



## **2.4 Flowchart Seminar Hasil dan Sidang Meja Hijau**



## **2.5 Use case diagram**



Pada use case diagram diatas, terdapat 4 aktor yaitu mahasiswa, staff, dan dosen dan kepala lab. Masing-masing aktor tersebut berinteraksi dengan use case yang ada dalam sistem.

a) Login

Use case login berinteraksi dengan ketiga aktor yaitu mahasiswa, staff , kepala lab dan dosen dimana use case tersebut sekaligus memiliki hubungan include dengan use case verifikasi kredensial. Hal ini berarti bahwa keempat aktor yang ingin masuk ke sistem harus memberikan verifikasi kredensial berupa email dan password dan kemudian sistem akan melakukan verifikasi kredensial dengan data yang terdapat database. Apabila kredensial yang diberikan benar, maka keempat aktor akan di alihkan ke halaman dashboard sesuai level user yang ada.

b) Mencetak dan mengunduh berkas

Use case ini berinteraksi dengan keempat aktor yang ada. Use case ini ditujukan sebagai fitur yang berfungsi untuk mempermudah aktor dalam penyampaian berkas sehingga terciptanya kepraktisan dari sistem ini.

c) Melihat Jadwal

Use case ini berinteraksi dengan dua aktor yang ada, yaitu mahasiswa dan dosen. Use case ini ditujukan untuk mempermudah aktor dalam segi komunikasi jadwal.

d) Melihat daftar peserta

Use case ini berinteraksi dengan tiga aktor yang ada, yaitu admin, dosen, dan kaprodi. Use case ini berfungsi untuk melihat daftar mahasiswa yang sedang menjalankan skripsinya.

e) Memberikan konfirmasi seminar/sidang mahasiswa

Use case ini hanya berinteraksi dengan satu aktor saja, yaitu admin dan memiliki hubungan include dengan use case validasi data. Kata konfirmasi merujuk pada validasi kelayakan mahasiswa atau seminar/sidang yang dijalankannya. Ketika admin ingin mengonfirmasi mahasiswa tersebut, maka sistem akan langsung melakukan validasi data yang diberikan pada sistem dan mengubah status mahasiswa tersebut.

f) Manajemen user dan manajemen jadwal

Sama halnya dengan use case memberikan konfirmasi seminar/sidang mahasiswa, use case ini memiliki aktor dan include yang sama. Kedua use case ini hanya membedakan pengelolaan user dan jadwal saja. Use case ini berfungsi untuk menambahkan, mengubah, serta menghapus data sebelumnya. Ketika admin ingin melakukan tiga aktivitas tersebut, maka sistem akan langsung melakukan validasi data yang diberikan pada sistem terkait tipe dan ukuran data.

g) Menginput nilai

Use case ini hanya berinteraksi dengan satu aktor saja, yaitu admin dan memiliki hubungan include dengan use case validasi data. Pada use case ini admin dapat menginputkan nilai mahasiswa bimbingan. Inputan akan melakukakan validasi data yang diberikan pada sistem terkait tipe dan ukuram data.

h) Status

Use case ini berfungsi untuk melihat status mahasiswa yang sedang mengikuti kegiatan tugas akhir.

## **2.6 Daftar Tabel dan View**

* Jumlah tabel utama : 18
* Jumlah tabel log : 9
* Jumlah view : 10

1. Tabel Utama

1. Tabel banding\_semhas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_semhas\_mhs | bigint | 20 | MUL |
| nidn | varchar | 255 | MUL |
| id\_semhas | bigint | 20 | MUL |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel banding\_sempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_sempro\_mhs | bigint | 20 | MUL |
| nidn | varchar | 255 | MUL |
| id\_sempro | bigint | 20 | MUL |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel banding\_sidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_sidang\_mhs | bigint | 20 | MUL |
| nidn | varchar | 255 | MUL |
| id\_sidang | bigint | 20 | MUL |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel bimbingans

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| nim | bigint | 20 | MUL |
| nidn | varchar | 255 | MUL |
| catatan | text | 20 |  |
| nilai | double |  |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel dosens

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| nidn | bigint | 255 | PRI |
| nama | varchar | 255 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel dospems

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| nim | bigint | 20 | MUL |
| nidn | varchar | 255 | MUL |
| statusdp | enum | ('dp1','dp2') |  |
| accdp | enum | ('tolak','terima','pending') |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel dospem\_semhas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_dospem | bigint | 20 | MUL |
| id\_semhas | bigint | 20 | MUL |
| accdp | enum | ('terima','pending') |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel dospem\_sempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_dospem | bigint | 20 | MUL |
| id\_sempro | bigint | 20 | MUL |
| accdp | enum | ('terima','pending') |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel dospem\_sidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| id\_dospem | bigint | 20 | MUL |
| id\_sidang | bigint | 20 | MUL |
| accdp | enum | ('terima','pending') |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel mahasiswas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| nim | varchar | 255 | PRI |
| nama | varchar | 255 |  |
| status | varchar | 255 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel pasca\_sidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel prasemhas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| nim | varchar | 255 | MUL |
| judul | varchar | 255 |  |
| berkas\_semhas | varchar | 255 |  |
| status | enum | ('terima','pending','tolak') |  |
| id\_semhas | bigint | 20 | MUL |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel prasidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| nim | varchar | 255 | MUL |
| judul | varchar | 255 |  |
| berkas\_sidang | varchar | 255 |  |
| status | enum | ('terima','pending','tolak') |  |
| id\_sidang | bigint | 20 | MUL |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel prasempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| nim | varchar | 255 | MUL |
| bidang | varchar | 255 |  |
| judul | varchar | 255 |  |
| diajukan\_oleh | enum | ('dosen','mahasiswa','bersama') |  |
| berkas\_uji | varchar | 255 |  |
| nilai\_uji | double | 4,2 |  |
| catatan\_uji | text |  |  |
| berkas\_sempro | varchar | 255 |  |
| status | enum | ('terima','pending','tolak') |  |
| id\_sempro | bigint | 20 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel semhas



1. Tabel sempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| tanggal | date |  |  |
| jam | time |  |  |
| tempat | varchar | 255 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel sidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| tanggal | date |  |  |
| jam | time |  |  |
| tempat | varchar | 255 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel users

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 | PRI |
| no\_induk | varchar | 255 | MUL |
| email | varchar | 255 | UNI |
| password | varchar | 255 |  |
| status | varchar | 255 |  |
| remember\_token | varchar | 100 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

2. Tabel Log

1. Tabel log\_banding\_semhas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| id\_prasemhas | bigint | 20 |  |
| nidn | varchar | 255 |  |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_banding\_sempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| id\_prasempro | bigint | 20 |  |
| nidn | varchar | 255 |  |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_banding\_sidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| id\_prasidang | bigint | 20 |  |
| nidn | varchar | 255 |  |
| nilai | double | 4,2 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_dosens

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| nidn | varchar | 255 |  |
| nama | varchar | 255 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_mahasiswas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| nim | Varchar | 255 |  |
| nidn | varchar | 255 |  |
| nama | varchar | 255 |  |
| status | int | 11 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_prasemhas

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| nim | varchar | 255 |  |
| judul | varchar | 255 |  |
| nilai\_banding | double | 4,2 |  |
| berkas\_semhas | varchar | 255 |  |
| status | enum | ('terima','pending','tolak',’lulus’) |  |
| id\_semhas | bigint | 20 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_prasempros

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| nim | varchar | 255 |  |
| bidang | varchar | 255 |  |
| judul | double | 4,2 |  |
| diajukan\_oleh | enum | (‘dosen’,’nahasiswa’,’bersama’) |  |
| berkas\_uji | varchar | 255 |  |
| nilai\_uji | double | 4,2 |  |
| nilai\_banding | double | 4,2 |  |
| catatan\_uji | text |  |  |
| berkas\_sempro | varchar | 255 |  |
| status | enum | (‘terima’,’pending’,’tolak’,’lulus’) |  |
| id\_sempro | bigint | 20 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_prasidangs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| nim | varchar | 255 |  |
| judul | varchar | 255 |  |
| nilai\_banding | double | 4,2 |  |
| berkas\_sidang | varchar | 255 |  |
| status | enum | ('terima','pending','tolak',’lulus’) |  |
| id\_sidang | bigint | 20 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

1. Tabel log\_users

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| no | int | 11 | PRI |
| log\_type | varchar | 255 |  |
| user | varchar | 255 |  |
| id | int | 11 |  |
| noInduk | varchar | 255 |  |
| email | varchar | 255 |  |
| password | varchar | 255 |  |
| status | varchar | 255 |  |
| remember\_token | varchar | 100 |  |
| created\_at | timestamp |  |  |
| updated\_at | timestamp |  |  |

3. View

1. View akundsn

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| email | varchar | 255 |  |
| nidn | varchar | 255 |  |
| nama | varchar | 255 |  |

1. View akunmhs

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| email | varchar | 255 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| status | varchar | 255 |  |

1. View mhsdospem

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| email | varchar | 255 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| status | varchar | 255 |  |
| nidn | varchar | 255 |  |
| statusdp | enum | ('dp1,'dp2') |  |
| accdp | enum | ('tolak','terima','pending') |  |
| namadsn | varchar | 255 |  |

1. View mhs\_dsn\_bim

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| nidn | varchar | 255 |  |
| statusdp | enum | ('dp1','dp2') |  |
| namadsn | varchar | 255 |  |
| catatan | enum | ('dp1,'dp2') |  |
| created\_at | timestamp |  |  |

1. View semhas\_jadwal

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| id\_semhas | bigint | 20 |  |
| judul | varchar | 255 |  |
| tanggal | date |  |  |
| jam | time |  |  |
| tempat | varchar | 255 |  |

1. View sempro\_jadwal

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| id\_sempro | bigint | 20 |  |
| judul | varchar | 255 |  |
| tanggal | date |  |  |
| jam | time |  |  |
| tempat | varchar | 255 |  |

1. View sidang\_jadwal

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| id | bigint | 20 |  |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| id\_sidang | bigint | 20 |  |
| judul | varchar | 255 |  |
| tanggal | date |  |  |
| jam | time |  |  |
| tempat | varchar | 255 |  |

1. View semhas\_dsn

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| statusmhs | int | 11 |  |
| nidn | varchar | 255 |  |
| namadsn | varchar | 255 |  |
| id | bigint | 20 |  |
| judul | varchar | 255 |  |
| berkas\_semhas | varchar | 255 |  |
| status | enum | ‘terima’,’pending’,’tolak’,’lulus’ |  |
| id\_semhas | bigint | 20 |  |

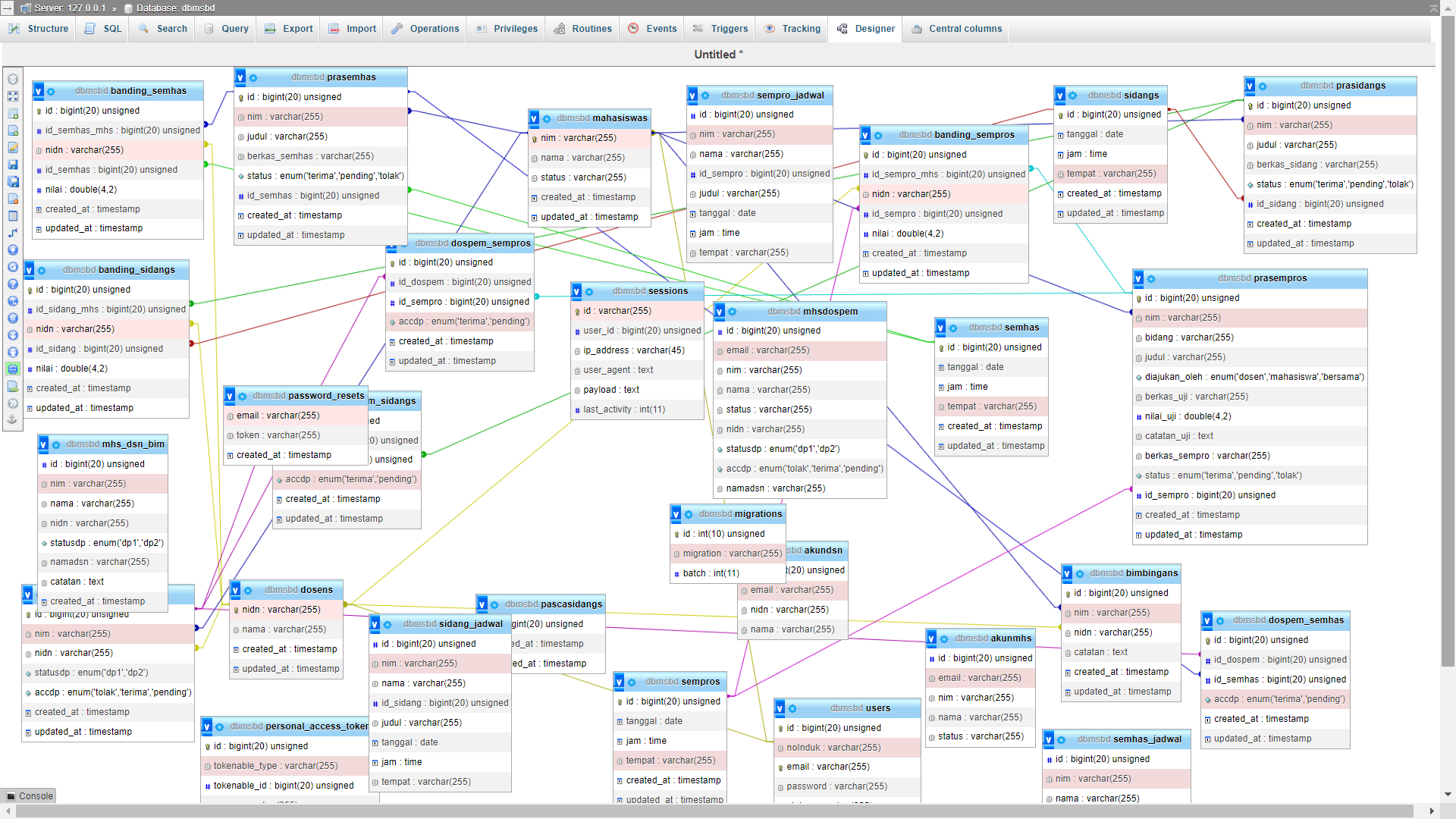
1. View sempro\_dsn

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| statusmhs | int | 11 |  |
| nidn | varchar | 255 |  |
| namadsn | varchar | 255 |  |
| id | bigint | 20 |  |
| judul | varchar | 255 |  |
| berkas\_sempro | varchar | 255 |  |
| status | enum | ‘terima’,’pending’,’tolak’,’lulus’ |  |
| id\_sempro | bigint | 20 |  |

1. View sidang\_dsn

|  |  |  |  |
| --- | --- | --- | --- |
| Field | Type | Length | Keterangan |
| nim | varchar | 255 |  |
| nama | varchar | 255 |  |
| statusmhs | int | 11 |  |
| nidn | varchar | 255 |  |
| namadsn | varchar | 255 |  |
| id | bigint | 20 |  |
| judul | varchar | 255 |  |
| berkas\_sidang | varchar | 255 |  |
| status | enum | ‘terima’,’pending’,’tolak’,’lulus’ |  |
| id\_sidang | bigint | 20 |  |

## **2.7 Relasi Antar Tabel**



## **2.8 Daftar Trigger**

1. Trigger log\_banding\_semhas\_delete

*CREATE TRIGGER `log\_banding\_semhas\_delete` AFTER DELETE ON `banding\_semhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_semhas(log\_type, user, id, id\_prasemhas, nidn, nilai, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.id\_prasemhas, old.nidn, old.nilai, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_banding\_semhas\_insert

*CREATE TRIGGER `log\_banding\_semhas\_insert` AFTER INSERT ON `banding\_semhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_semhas(log\_type, user, id, id\_prasemhas, nidn, nilai, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.id\_prasemhas, new.nidn, new.nilai, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_banding \_semhas\_update

*CREATE TRIGGER `log\_banding\_semhas\_update` AFTER UPDATE ON `banding\_semhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_semhas(log\_type, user, id, id\_prasemhas, nidn, nilai, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.id\_prasemhas, old.nidn, old.nilai, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_banding\_sempros\_delete

*CREATE TRIGGER `log\_banding\_sempros\_delete` AFTER DELETE ON `banding\_sempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sempros(log\_type, user, id, id\_prasempro, nidn, nilai, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.id\_prasempro, old.nidn, old.nilai, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_banding\_sempros\_insert

*CREATE TRIGGER `log\_banding\_sempros\_insert` AFTER INSERT ON `banding\_sempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sempros(log\_type, user, id, id\_prasempro, nidn, nilai, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.id\_prasempro, new.nidn, new.nilai, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_banding\_sempros\_update

*CREATE TRIGGER `log\_banding\_sempros\_update` AFTER UPDATE ON `banding\_sempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sempros(log\_type, user, id, id\_prasempro, nidn, nilai, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.id\_prasempro, old.nidn, old.nilai, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_banding\_sidangs\_delete

*CREATE TRIGGER `log\_banding\_sidangs\_delete` AFTER DELETE ON `banding\_sidangs`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sidangs(log\_type, user, id, id\_prasidang, nidn, nilai, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.id\_prasidang, old.nidn, old.nilai, old.created\_at, old.updated\_at); END*

1. Trigger log\_banding\_sidangs\_update

*CREATE TRIGGER `log\_banding\_sidangs\_update` AFTER UPDATE ON `banding\_sidangs`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sidangs(log\_type, user, id, id\_prasidang, nidn, nilai, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.id\_prasidang, old.nidn, old.nilai, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_banding\_sempros\_insert

*CREATE TRIGGER `log\_banding\_sempros\_insert` AFTER INSERT ON `banding\_sempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_banding\_sempros(log\_type, user, id, id\_prasempro, nidn, nilai, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.id\_prasempro, new.nidn, new.nilai, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_dosens\_delete

*CREATE TRIGGER `log\_dosens\_delete` AFTER DELETE ON `dosens`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_dosens(log\_type, user, nidn, nama, created\_at, updated\_at)*

*VALUES('delete', USER(), old.nidn, old.nama, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_dosens\_insert

*CREATE TRIGGER `log\_dosens\_insert` AFTER INSERT ON `dosens`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_dosens(log\_type, user, nidn, nama, created\_at, updated\_at)*

*VALUES('insert', USER(), new.nidn, new.nama, new.created\_at, new.updated\_at); END*

1. Trigger log\_dosens\_update

*CREATE TRIGGER `log\_dosens\_update` AFTER UPDATE ON `dosens`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_dosens(log\_type, user, nidn, nama, created\_at, updated\_at)*

*VALUES('update', USER(), old.nidn, old.nama, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_mahasiswas\_delete

*CREATE TRIGGER `log\_mahasiswas\_delete` AFTER DELETE ON `mahasiswas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_mahasiswas(log\_type, user, nim, nama, status, created\_at, updated\_at)*

*VALUES('delete', USER(), old.nim, old.nama, old.status, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_mahasiswas\_insert

*CREATE TRIGGER `log\_mahasiswas\_insert` AFTER INSERT ON `mahasiswas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_mahasiswas(log\_type, user, nim, nama, status, created\_at, updated\_at)*

*VALUES('insert', USER(), new.nim, new.nama, new.status, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_mahasiswas\_update

*CREATE TRIGGER `log\_mahasiswas\_update` AFTER UPDATE ON `mahasiswas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_mahasiswas(log\_type, user, nim, nama, status, created\_at, updated\_at)*

*VALUES('update', USER(), old.nim, old.nama, old.status, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_prasemhas\_delete

*CREATE TRIGGER `log\_prasemhas\_delete` AFTER DELETE ON `prasemhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasemhas(log\_type, user, id, nim, judul, nilai\_banding, berkas\_semhas, status, id\_semhas, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.nim, old.judul, old.nilai\_banding, old.berkas\_semhas, old.status, old.id\_semhas, old.created\_at, old.updated\_at);*

1. Trigger log\_prasemhas\_insert

*CREATE TRIGGER `log\_prasemhas\_insert` AFTER INSERT ON `prasemhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasemhas(log\_type, user, id, nim, judul, nilai\_banding, berkas\_semhas, status, id\_semhas, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.nim, new.judul, new.nilai\_banding, new.berkas\_semhas, new.status, new.id\_semhas, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_prasemhas\_update

*CREATE TRIGGER `log\_prasemhas\_update` AFTER UPDATE ON `prasemhas`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasemhas(log\_type, user, id, nim, judul, nilai\_banding, berkas\_semhas, status, id\_semhas, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.nim, old.judul, old.nilai\_banding, old.berkas\_semhas, old.status, old.id\_semhas, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_prasempros\_delete

*CREATE TRIGGER `log\_prasempros\_delete` AFTER DELETE ON `prasempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasempros(log\_type, user, id, nim, bidang, judul, diajukan\_oleh, berkas\_uji, nilai\_uji, nilai\_banding, catatan\_uji, berkas\_sempro, status, id\_sempro, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.nim, old.bidang, old.judul, old.diajukan\_oleh, old.berkas\_uji, old.nilai\_uji, old.nilai\_banding, old.catatan\_uji, old.berkas\_sempro, old.status, old.id\_sempro, old.created\_at, old.updated\_at); END*

1. Trigger log\_prasempros\_insert

*CREATE TRIGGER `log\_prasempros\_insert` AFTER INSERT ON `prasempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasempros(log\_type, user, id, nim, bidang, judul, diajukan\_oleh, berkas\_uji, nilai\_uji, nilai\_banding, catatan\_uji, berkas\_sempro, status, id\_sempro, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.nim, new.bidang, new.judul, new.diajukan\_oleh, new.berkas\_uji, new.nilai\_uji, new.nilai\_banding, new.catatan\_uji, new.berkas\_sempro, new.status, new.id\_sempro, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_prasempros\_update

*CREATE TRIGGER `log\_prasempros\_update` AFTER UPDATE ON `prasempros`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasempros(log\_type, user, id, nim, bidang, judul, diajukan\_oleh, berkas\_uji, nilai\_uji, nilai\_banding, catatan\_uji, berkas\_sempro, status, id\_sempro, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.nim, old.bidang, old.judul, old.diajukan\_oleh, old.berkas\_uji, old.nilai\_uji, old.nilai\_banding, old.catatan\_uji, old.berkas\_sempro, old.status, old.id\_sempro, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_prasidangs\_delete

*CREATE TRIGGER `log\_prasidangs\_delete` AFTER DELETE ON `prasidangs`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasidangs(log\_type, user, id, nim, judul, nilai\_banding, berkas\_sidang, status, id\_sidang, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.nim, old.judul, old.nilai\_banding, old.berkas\_sidang, old.status, old.id\_sidang, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_prasidangs\_insert

*CREATE TRIGGER `log\_prasidangs\_insert` AFTER INSERT ON `prasidangs`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasidangs(log\_type, user, id, nim, judul, nilai\_banding, berkas\_sidang, status, id\_sidang, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.nim, new.judul, new.nilai\_banding, new.berkas\_sidang, new.status, new.id\_sidang, new.created\_at, new.updated\_at);*

*END*

1. Trigger log\_prasidangs\_update

*CREATE TRIGGER `log\_prasidangs\_update` AFTER UPDATE ON `prasidangs`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_prasidangs(log\_type, user, id, nim, judul, nilai\_banding, berkas\_sidang, status, id\_sidang, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.nim, old.judul, old.nilai\_banding, old.berkas\_sidang, old.status, old.id\_sidang, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_users\_delete

*CREATE TRIGGER `log\_users\_delete` AFTER DELETE ON `users`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_users(log\_type, user, id, noInduk, email, password, status, created\_at, updated\_at)*

*VALUES('delete', USER(), old.id, old.noInduk, old.email, old.password, old.status, old.created\_at, old.updated\_at);*

*END*

1. Trigger log\_users\_insert

*CREATE TRIGGER `log\_users\_insert` AFTER INSERT ON `users`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_users(log\_type, user, id, noInduk, email, password, status, created\_at, updated\_at)*

*VALUES('insert', USER(), new.id, new.noInduk, new.email, new.password, new.status, new.created\_at, new.updated\_at); END*

1. Trigger log users\_update

*CREATE TRIGGER `log\_users\_update` AFTER UPDATE ON `users`*

*FOR EACH ROW BEGIN*

*INSERT INTO log\_users(log\_type, user, id, noInduk, email, password, status, created\_at, updated\_at)*

*VALUES('update', USER(), old.id, old.noInduk, old.email, old.password, old.status, old.created\_at, old.updated\_at);*

*END*

1. Trigger trg\_accPrasemhas

*CREATE TRIGGER `trg\_accPrasemhas` AFTER UPDATE ON `dospem\_semhas`*

*FOR EACH ROW BEGIN*

*DECLARE hasil VARCHAR(10) DEFAULT "";*

*SET hasil = (SELECT COUNT(\*) FROM dospem\_semhas*

*WHERE accdp = "terima" AND id\_semhas = old.id\_semhas);*

*IF(hasil = "2") THEN*

*UPDATE prasemhas*

*SET status = "terima", updated\_at = now()*

*WHERE id = old.id\_semhas;*

*UPDATE mahasiswas*

*SET status = 9*

*WHERE nim = (SELECT nim FROM prasemhas WHERE id = old.id\_semhas);*

*END IF;*

*END*

1. Trigger trg\_accPrasempro

*CREATE TRIGGER `trg\_accPrasempro` AFTER UPDATE ON `dospem\_sempros`*

*FOR EACH ROW BEGIN*

*DECLARE hasil VARCHAR(10) DEFAULT "";*

*SET hasil = (SELECT COUNT(\*) FROM dospem\_sempros*

*WHERE accdp = "terima" AND id\_sempro = old.id\_sempro);*

*IF(hasil = "2") THEN*

*UPDATE prasempros*

*SET status = "terima", updated\_at = now()*

*WHERE id = old.id\_sempro;*

*END IF;*

*END*

1. Trigger trg\_accPrasidang

*CREATE TRIGGER `trg\_accPrasidang` AFTER UPDATE ON `dospem\_sidangs`*

*FOR EACH ROW BEGIN*

*DECLARE hasil VARCHAR(10) DEFAULT "";*

*SET hasil = (SELECT COUNT(\*) FROM dospem\_sidangs*

*WHERE accdp = "terima" AND id\_sidang = old.id\_sidang);*

*IF(hasil = "2") THEN*

*UPDATE prasidangs*

*SET status = "terima", updated\_at = now()*

*WHERE id = old.id\_sidang;*

*UPDATE mahasiswas*

*SET status = 13*

*WHERE nim = (SELECT nim FROM prasidangs WHERE id = old.id\_sidang);*

*END IF;*

*END*

1. Trigger trg\_check\_dospems\_create\_prasempros

*CREATE TRIGGER `trg\_check\_dospems\_create\_prasempros` BEFORE INSERT ON `prasempros`*

*FOR EACH ROW BEGIN*

*IF( (SELECT status FROM mahasiswas WHERE nim = new.nim) != 2 )*

*THEN*

*SIGNAL SQLSTATE '45000'*

*SET MESSAGE\_TEXT = "status dosen pembimbing belum diterima";*

*END IF;*

*END*

1. Trigger trg\_rerataNilaiSemhas

*CREATE TRIGGER `trg\_rerataNilaiSemhas` AFTER UPDATE ON `banding\_semhas`*

*FOR EACH ROW BEGIN*

*DECLARE hasil DOUBLE(4,2);*

*DECLARE jmlhsmua INTEGER DEFAULT 0;*

*DECLARE jmlhnotnull INTEGER DEFAULT 0;*

*DECLARE nims VARCHAR(255) DEFAULT '';*

*DECLARE juduls VARCHAR(255) DEFAULT '';*

*SET jmlhnotnull = (SELECT COUNT(\*) FROM banding\_semhas*

*WHERE id\_prasemhas = old.id\_prasemhas AND nilai IS NOT NULL);*

*SET jmlhsmua = (SELECT COUNT(\*) FROM banding\_semhas*

*WHERE id\_prasemhas = old.id\_prasemhas);*

*SET nims = (SELECT nim FROM prasemhas WHERE id = old.id\_prasemhas);*

*SET juduls = (SELECT judul FROM prasemhas WHERE id = old.id\_prasemhas);*

*IF(jmlhnotnull = jmlhsmua) THEN*

*SET hasil = (SELECT AVG(nilai) FROM banding\_semhas*

*WHERE id\_prasemhas = old.id\_prasemhas);*

*IF(hasil>=70) THEN*

*UPDATE prasemhas*

*SET nilai\_banding = hasil, status = "lulus", updated\_at = now()*

*WHERE id = old.id\_prasemhas;*

*CALL buatPrasidangs(nims, juduls);*

*ELSE*

*UPDATE prasemhas*

*SET nilai\_banding = hasil, status = "tolak", updated\_at = now()*

*WHERE id = old.id\_prasemhas;*

*END IF;*

*END IF;*

*END*

1. Trigger trg\_reratanilaiSempro

*CREATE TRIGGER `trg\_rerataNilaiSempro` AFTER UPDATE ON `banding\_sempros`*

*FOR EACH ROW BEGIN*

*DECLARE hasil DOUBLE(4,2);*

*DECLARE jmlhsmua INTEGER DEFAULT 0;*

*DECLARE jmlhnotnull INTEGER DEFAULT 0;*

*DECLARE nims VARCHAR(255) DEFAULT '';*

*DECLARE juduls VARCHAR(255) DEFAULT '';*

*SET jmlhnotnull = (SELECT COUNT(\*) FROM banding\_sempros*

*WHERE id\_prasempro = old.id\_prasempro AND nilai IS NOT NULL);*

*SET jmlhsmua = (SELECT COUNT(\*) FROM banding\_sempros*

*WHERE id\_prasempro = old.id\_prasempro);*

*SET nims = (SELECT nim FROM prasempros WHERE id = old.id\_prasempro);*

*SET juduls = (SELECT judul FROM prasempros WHERE id = old.id\_prasempro);*

*IF(jmlhnotnull = jmlhsmua) THEN*

*SET hasil = (SELECT AVG(nilai) FROM banding\_sempros*

*WHERE id\_prasempro = old.id\_prasempro);*

*IF(hasil>=70) THEN*

*UPDATE prasempros*

*SET nilai\_banding = hasil, status = "lulus", updated\_at = now()*

*WHERE id = old.id\_prasempro;*

*CALL buatPrasemhas(nims, juduls);*

*ELSE*

*UPDATE prasempros*

*SET nilai\_banding = hasil, status = "tolak", updated\_at = now()*

*WHERE id = old.id\_prasempro;*

*END IF;*

*END IF;*

*END*

1. Trigger trg\_rerataNilaiSidang

*CREATE TRIGGER `trg\_rerataNilaiSidang` AFTER UPDATE ON `banding\_sidangs`*

*FOR EACH ROW BEGIN*

*DECLARE hasil DOUBLE(4,2);*

*DECLARE jmlhsmua INTEGER DEFAULT 0;*

*DECLARE jmlhnotnull INTEGER DEFAULT 0;*

*DECLARE nims VARCHAR(255) DEFAULT '';*

*DECLARE juduls VARCHAR(255) DEFAULT '';*

*SET jmlhnotnull = (SELECT COUNT(\*) FROM banding\_sidangs*

*WHERE id\_prasidang = old.id\_prasidang AND nilai IS NOT NULL);*

*SET jmlhsmua = (SELECT COUNT(\*) FROM banding\_sidangs*

*WHERE id\_prasidang = old.id\_prasidang);*

*SET nims = (SELECT nim FROM prasidangs WHERE id = old.id\_prasidang);*

*SET juduls = (SELECT judul FROM prasidangs WHERE id = old.id\_prasidang);*

*IF(jmlhnotnull = jmlhsmua) THEN*

*SET hasil = (SELECT AVG(nilai) FROM banding\_sidangs*

*WHERE id\_prasidang = old.id\_prasidang);*

*IF(hasil>=70) THEN*

*UPDATE prasidangs*

*SET nilai\_banding = hasil, status = "lulus", updated\_at = now()*

*WHERE id = old.id\_prasidang;*

*UPDATE mahasiswas*

*SET status = "15"*

*WHERE `nim` = (SELECT nim FROM prasidangs WHERE id = old.id\_prasidang);*

*ELSE*

*UPDATE prasidangs*

*SET nilai\_banding = hasil, status = "tolak", updated\_at = now()*

*WHERE id = old.id\_prasidang;*

*END IF;*

*END IF;*

*END*

1. Trigger trg\_updatestat\_10

*CREATE TRIGGER `trg\_updatestat\_10` AFTER UPDATE ON `prasemhas`*

*FOR EACH ROW BEGIN*

*IF(new.id\_semhas IS NOT NULL)*

*THEN*

*UPDATE mahasiswas*

*SET status = 10*

*WHERE nim = new.nim;*

*END IF;*

*END*

1. Trigger trg\_updatestat\_11

*CREATE TRIGGER `trg\_updatestat\_11` AFTER INSERT ON `prasidangs`*

*FOR EACH ROW BEGIN*

*UPDATE mahasiswas*

*SET status = 11*

*WHERE nim = new.nim;*

*END*

1. Trigger trg\_updatestat\_14

*CREATE TRIGGER `trg\_updatestat\_14` AFTER UPDATE ON `prasidangs`*

*FOR EACH ROW BEGIN*

*IF(new.id\_sidang IS NOT NULL)*

*THEN*

*UPDATE mahasiswas*

*SET status = 14*

*WHERE nim = new.nim;*

*END IF;*

*END*

1. Trigger trg\_updatestat\_2

*CREATE TRIGGER `trg\_updatestat\_2` AFTER UPDATE ON `dospems`*

*FOR EACH ROW BEGIN*

*DECLARE accDospem INTEGER DEFAULT 0;*

*SET accDospem = (SELECT COUNT(\*) FROM dospems*

*WHERE nim = new.nim AND accdp ='terima');*

*IF( accDospem = 2)*

*THEN*

*UPDATE mahasiswas*

*SET status = 2*

*WHERE nim = new.nim;*

*END IF;*

1. Trigger trg\_updatestat\_3

*CREATE TRIGGER `trg\_updatestat\_3` AFTER INSERT ON `prasempros`*

*FOR EACH ROW BEGIN*

*UPDATE mahasiswas*

*SET status = 3*

*WHERE nim = new.nim;*

*END*

1. Trigger trg\_updatestat\_4\_exum

*CREATE TRIGGER `trg\_updatestat\_4\_exum` AFTER UPDATE ON `prasempros`*

*FOR EACH ROW BEGIN*

*DECLARE hasil VARCHAR(10) DEFAULT "";*

*SET hasil = (SELECT COUNT(\*) FROM dospem\_sempros*

*WHERE accdp = "terima" AND id\_sempro = old.id);*

*IF(hasil = "2" AND (SELECT nilai\_uji FROM prasempros WHERE id = old.id) IS NOT NULL) THEN*

*UPDATE mahasiswas*

*SET status = 4*

*WHERE nim = old.nim;*

*END IF;*

*END*

1. Trigger trg\_updatestat\_4\_judul

*CREATE TRIGGER `trg\_updatestat\_4\_judul` AFTER UPDATE ON `dospem\_sempros`*

*FOR EACH ROW BEGIN*

*DECLARE hasil VARCHAR(10) DEFAULT "";*

*SET hasil = (SELECT COUNT(\*) FROM dospem\_sempros*

*WHERE accdp = "terima" AND id\_sempro = old.id\_sempro);*

*IF(hasil = "2" AND (SELECT nilai\_uji FROM prasempros WHERE id = old.id\_sempro) IS NOT NULL) THEN*

*UPDATE mahasiswas*

*SET status = 4*

*WHERE nim = (SELECT nim FROM prasempros WHERE id = old.id\_sempro);*

*END IF;*

*END*

1. Trigger trg\_update\_stat\_6

*CREATE TRIGGER `trg\_updatestat\_6` AFTER UPDATE ON `prasempros`*

*FOR EACH ROW BEGIN*

*IF( new.id\_sempro IS NOT NULL)*

*THEN*

*UPDATE mahasiswas*

*SET status = 6*

*WHERE nim = new.nim;*

*END IF;*

*END*

1. Trigger trg\_updatestat\_7

*CREATE TRIGGER `trg\_updatestat\_7` AFTER INSERT ON `prasemhas`*

*FOR EACH ROW BEGIN*

*UPDATE mahasiswas*

*SET status = 7*

*WHERE nim = new.nim;*

*END*

## **2.9 Daftar Prosedur**

1. Procedure accDospems

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `accDospems`(IN `ninotem` VARCHAR(64), IN `nidnn` VARCHAR(64), IN `accdp` ENUM('tolak','terima','pending'))

BEGIN

UPDATE dospems

SET `accdp` = accdp, `updated\_at` = NOW()

WHERE `nim` = nimm AND `nidn` = nidnn;

END$$

DELIMITER ;

1. Procedure Prasempro

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `accPrasempro`(IN `iddospem` VARCHAR(5), IN `acc` ENUM('terima','pending'))

BEGIN

UPDATE dospem\_sempros

SET `accdp` = acc, `updated\_at` = NOW()

WHERE `id\_dospem` = iddospem;

END$$

DELIMITER ;

1. Procedure buatAkun

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `buatAkun`(IN `mail` VARCHAR(64), IN `password` VARCHAR(64), IN `status` ENUM("mahasiswa","dosen"), IN `induk` VARCHAR(64), IN `nama` VARCHAR(64))

BEGIN

IF(status = "mahasiswa") THEN

INSERT INTO users(`noInduk`, `password`, `email`, `status`)

VALUES (induk, password, mail , status);

INSERT INTO mahasiswas (`nim`, `nama`, `created\_at`, `updated\_at`)VALUES (induk, nama, now(), null);

ELSEIF(status = "dosen") THEN

INSERT INTO users(`noInduk`, `password`, `email`, `status`)

VALUES (induk, password, mail , status);

INSERT INTO dosens(`nidn`, `nama`,`created\_at`, `updated\_at`)

VALUES (induk, nama, now(), null);

END IF;

END$$

DELIMITER ;

1. Procedure buatDospems

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `buatDospems`(IN `nim` VARCHAR(64), IN `nidn` VARCHAR(64), IN `statusdp` ENUM("dp1","dp2"))

BEGIN

INSERT INTO dospems(`nim`, `nidn`, `statusdp`, `accdp`, `created\_at`)

VALUES (nim, nidn, statusdp , "pending", NOW());

END$$

DELIMITER ;

1. Procedure buatPrasempro

DELIMITER $$

CREATE DEFINER=`root`@`localhost` PROCEDURE `buatPrasempro`(IN `nims` VARCHAR(64), IN `bidangs` VARCHAR(64), IN `juduls` VARCHAR(255), IN `ajukans` ENUM("mahasiswa","dosen","bersama"), IN `berkas\_ujis` VARCHAR(255))

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE praid INTEGER DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id FROM dospems WHERE nim = nims;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET \_continue =1;

INSERT INTO prasempros(`nim`, `bidang`, `judul`, `diajukan\_oleh`, `berkas\_uji`, `created\_at`) VALUES (nims, bidangs, juduls, ajukans, berkas\_ujis, NOW());

SET praid = LAST\_INSERT\_ID();

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

INSERT INTO dospem\_sempros(`id\_dospem`, `id\_sempro`, `accdp`, `created\_at`) VALUES (MyField, praid,"pending", NOW());

END LOOP getData;

CLOSE MyCursor;

END$$

DELIMITER ;

1. Procedure addSempro

BEGIN

UPDATE prasempros

SET id\_sempro = idsempro, `updated\_at` = NOW()

WHERE `id` = idprasem;

CALL procedure\_dospemApplyBandingSempro(idsempro);

END

1. Procedure addNilaiSempro

BEGIN

UPDATE banding\_sempros

SET `nilai` = nilais, `updated\_at` = NOW()

WHERE `id\_prasempro` = idprasem AND `nidn` = nidns;

END

1. Procedure applyBandingSempro

BEGIN

DECLARE nidns VARCHAR(64) DEFAULT "";

SET nidns = (SELECT nidn FROM dospems WHERE id = iddosen);

INSERT INTO banding\_sempros(`id\_prasempro`, `nidn`, `created\_at`) VALUES (idprasem, nidns, NOW());

END

1. Procedure buatPrasemhas

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE praid INTEGER DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id FROM dospems WHERE nim = nims;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET \_continue =1;

INSERT INTO prasemhas(`nim`, `judul`, `created\_at`) VALUES (nims, juduls , NOW());

SET praid = LAST\_INSERT\_ID();

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

INSERT INTO dospem\_semhas(`id\_dospem`, `id\_semhas`, `accdp`, `created\_at`) VALUES (MyField, praid,"pending", NOW());

END LOOP getData;

CLOSE MyCursor;

END

1. Procedure buatSempros

BEGIN

INSERT INTO sempros(`tanggal`, `jam`, `tempat`, `created\_at`)

VALUES (tgl, jams, place, NOW());

END

1. Procedure accPrasemhas

BEGIN

UPDATE dospem\_semhas

SET `accdp` = acc, `updated\_at` = NOW()

WHERE `id\_dospem` = iddospem;

END

1. Procedure accPrasidang

BEGIN

UPDATE dospem\_sidangs

SET `accdp` = acc, `updated\_at` = NOW()

WHERE `id\_dospem` = iddospem;

END

1. Procedure addSidang

BEGIN

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

UPDATE prasidangs

SET id\_sidang = idsidang, `updated\_at` = NOW()

WHERE `id` = idprasem;

CALL procedure\_dospemApplyBandingsidang(idprasem);

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure accPrasidang

BEGIN

UPDATE dospem\_sidangs

SET `accdp` = acc, `updated\_at` = NOW()

WHERE `id\_dospem` = iddospem;

END

1. Procedure addSemhas

BEGIN

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

UPDATE prasemhas

SET id\_semhas = idsemhas, `updated\_at` = NOW()

WHERE `id` = idprasem;

CALL procedure\_dospemApplyBandingSemhas(idprasem);

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure addSidang

BEGIN

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

UPDATE prasidangs

SET id\_sidang = idsidang, `updated\_at` = NOW()

WHERE `id` = idprasem;

CALL procedure\_dospemApplyBandingsidang(idprasem);

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure addnilaiSidang

BEGIN

UPDATE banding\_sidangs

SET `nilai` = nilais, `updated\_at` = NOW()

WHERE `id\_prasidang` = idprasem AND `nidn` = nidns;

END

1. Procedure addnilaiUji

BEGIN

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

IF(nilais>75)

THEN

UPDATE prasempros

SET `nilai\_uji` = nilais, status = "terima", `updated\_at` = NOW()

WHERE `id` = idprasem;

ELSE

UPDATE prasempros

SET `nilai\_uji` = nilais, status = "tolak", `updated\_at` = NOW()

WHERE `id` = idprasem;

END IF;

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure procedure\_dospemApplyBandingSempro

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id\_dospem FROM dospem\_sempros WHERE id\_sempro = idsempro;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET \_continue = TRUE;

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

CALL applyBandingSempro(idsempro, MyField);

END LOOP getData;

CLOSE MyCursor;

END

1. Procedure procedure\_dospemApplyBandingSemhas

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id\_dospem FROM dospem\_semhas WHERE id\_semhas = idsemhas;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET \_continue = TRUE;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

CALL applyBandingsemhas(idsemhas, MyField);

END LOOP getData;

CLOSE MyCursor;

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure procedure\_dospemApplyBandingSidang

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE `\_rollback` BOOL DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id\_dospem FROM dospem\_sidangs WHERE id\_sidang = idsidang;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET \_continue = TRUE;

DECLARE CONTINUE HANDLER FOR SQLEXCEPTION SET `\_rollback` = 1;

START TRANSACTION;

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

CALL applyBandingsidang(idsidang, MyField);

END LOOP getData;

CLOSE MyCursor;

IF `\_rollback` THEN

ROLLBACK;

ELSE

COMMIT;

END IF;

END

1. Procedure buatPrasidangs

BEGIN

DECLARE MyField VARCHAR(255) DEFAULT "";

DECLARE \_continue INTEGER DEFAULT 0;

DECLARE praid INTEGER DEFAULT 0;

DECLARE MyCursor CURSOR FOR

SELECT id FROM dospems WHERE nim = nims;

DECLARE CONTINUE HANDLER FOR NOT FOUND

SET \_continue =1;

-- UPDATE mahasiswas

-- SET status = "5"

-- WHERE `nim` = nims;

INSERT INTO prasidangs(`nim`, `judul`, `created\_at`) VALUES (nims, juduls , NOW());

SET praid = LAST\_INSERT\_ID();

OPEN MyCursor;

getData: LOOP

FETCH MyCursor INTO MyField;

IF \_continue = 1 THEN

LEAVE getData;

END IF;

INSERT INTO dospem\_sidangs(`id\_dospem`, `id\_sidang`, `accdp`, `created\_at`) VALUES (MyField, praid,"pending", NOW());

END LOOP getData;

CLOSE

1. Procedure buatSemhas

BEGIN

INSERT INTO semhas(`tanggal`, `jam`, `tempat`, `created\_at`)

VALUES (tgl, jams, place, NOW());

END

1. Procedure buatSidang

BEGIN

INSERT INTO sidangs(`tanggal`, `jam`, `tempat`, `created\_at`)

VALUES (tgl, jams, place, NOW());

END

1. Procedure UplBerkasSemhas

BEGIN

IF( (SELECT status FROM mahasiswas WHERE nim = (SELECT nim FROM prasemhas WHERE id = idprasem) ) != 7 )

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "Belum dipersilahkan";

ELSE

UPDATE prasemhas

SET `berkas\_semhas` = berkas, `updated\_at` = NOW()

WHERE `id` = idprasem;

UPDATE mahasiswas

SET status = 8

WHERE nim = (SELECT nim FROM prasemhas WHERE id = idprasem);

END IF;

END

1. Procedure UplBerkasSempros

BEGIN

IF( (SELECT status FROM mahasiswas WHERE nim = (SELECT nim FROM prasempros WHERE id = idprasem) ) != 4 )

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "Judul belum diterima Dospem atau Nilai uji belum keluar";

ELSE

UPDATE prasempros

SET `berkas\_sempro` = berkas, `updated\_at` = NOW()

WHERE `id` = idprasem;

UPDATE mahasiswas

SET status = 5

WHERE nim = (SELECT nim FROM prasempros WHERE id = idprasem);

END IF;

END

1. Procedure UplBerkasSidang

BEGIN

IF( (SELECT status FROM mahasiswas WHERE nim = (SELECT nim FROM prasidangs WHERE id = idprasem) ) != 11 )

THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = "Belum dipersilahkan";

ELSE

UPDATE prasidangs

SET `berkas\_sidang` = berkas, `updated\_at` = NOW()

WHERE `id` = idprasem;

UPDATE mahasiswas

SET status = 12

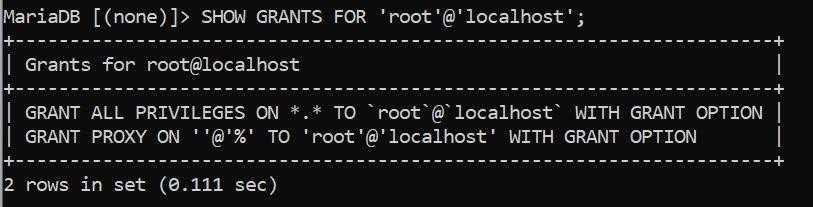
WHERE nim = (SELECT nim FROM prasidangs WHERE id = idprasem);

END IF;

END

## **2.10 Daftar User dan Privileges**

Use root :



# BAB III

# IMPLEMENTASI SISTEM

## **3.1 Kebutuhan Perangkat Keras dan Lunak**

* Hardware : cpu,memory,disk.
* Software : Visual Studio Code , XAMPP , Web Browser.
* Other : HTML , CSS , Java Script , Bootstrap , Laravel , PHP,composer.

## **3.2 Fitur , Subfitur dan Keterangan**

Adapun fitur, subfitur, dan keterangan dalam pengimplementasian website pengerjaan tugas akhir mahasiswa TI ini adalah sebagai berikut:

1. Mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Subfitur** | **Keterangan** |
| 1 | Pra Seminar  Proposal | Formulir Calon  Pembimbing | Menampilkan Nama dan NIM Mahasiswa beserta Dosen Pembimbingya |
| Pengajuan dan Perbaikan  Judul Skripsi | Menambahkan judul skripsi mahasiswa dan dapat mengedit atau memperbaiki judulnya. |
| Jadwal Seminar Poposal | Melihat jadwal seminar proposal mahasiswa |
| Hasil Uji Seminar Proposal | Melihat hasil uji seminar proposal mahasiswa |
| 2 | Pra Seminar Hasil | Pengajuan berkas seminar hasil | Mengupload berkas seminar hasil mahasiswa |
| Jadwal Seminar Hasil | Melihat jadwal seminar hasil mahasiswa |
| 3 | Pra Sidang Meja  Hijau | Status pra sidang | Melihat status pra sidang mahasiswa |
| Jadwal Sidang Meja Hijau | Melihat jadwal sidang meja hijau mahasiswa |
| 4 | Pasca Sidang  Meja HIjau | Penggandaan Skripsi | Mencetak form pengajuan penggandaan skripsi jika dibutuhkan |
| Format Jurnal | Mengunduh format jurnal |

1. Dosen

|  |  |  |
| --- | --- | --- |
| **Fitur** | **Subfitur** | **Keterangan** |
| Seminar proposal ,seminar hasil dan seminar sidang | Rincian Data mahasiswa | Melihat data mahasiswa bimbingan termasuk dosen pembimbing dan progress skripsi |
| Memberikan nilai | Dosen dapat memberikan nilai saat seminar dimana dosen tersebut berperan sebagai dosen pembimbing/pembanding dalam seminar tersebut |
| Catatan Bimbingan | Dosen dapat memberikan catatan bimbingan bagi mahasiswa yang dibimbingnya |

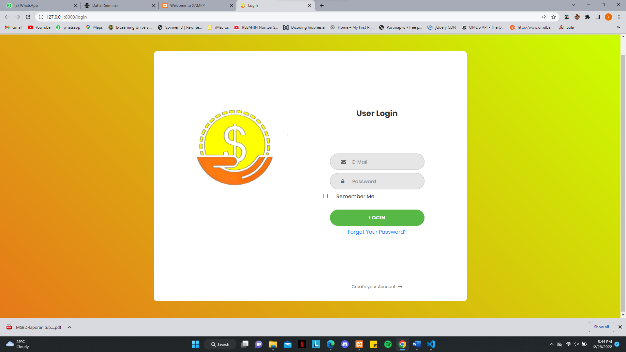
1. Staff

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Subfitur** | **Keterangan** |
| 1 | Penentuan Dosen Pembanding | Melihat dosen pembanding yang diajukan oleh mahasiswa | Melihat dan mengubah dosen pembimbing untuk setiap mahasiswa |
| 2 | Seminar Proposal | Jadwal dan tempat seminar proposal | Menampilkan tanggal, jam dan tempat seminar proposal |
| Edit seminar proposal | Mengedit tanggal, jam dan tempat seminar proposal. |
| 3 | Seminar Hasil | Jadwal dan tempat seminar hasil | Menampilkan tanggal, jam dan tempat seminar hasil |
| Membuat seminar hasil | Membuat tanggal, jam dan tempat seminar hasil |
| 4 | sidang | Jadwal dan tempat sidang | Menampilkan tanggal, jam dan tempat sidang |
| Buat sidang | Membuat tanggal, jam dan tempat sidang |

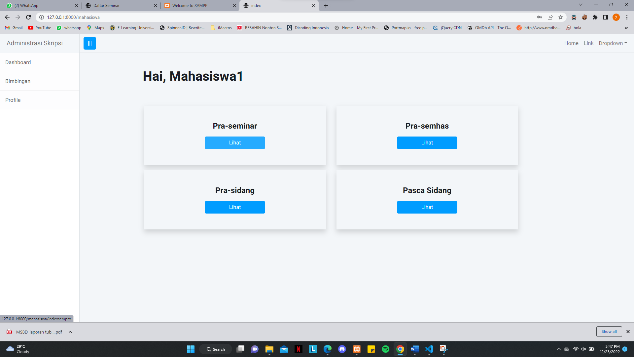
## **3.3 Implementasi Sistem**

Tampilan web database tugas akhir teknologi informasi :

1. Mahasiswa



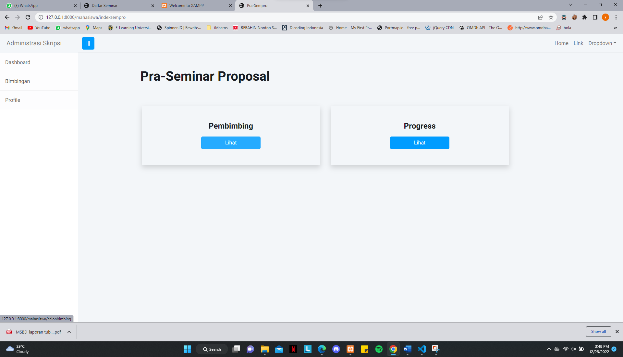
Login mahasiswa



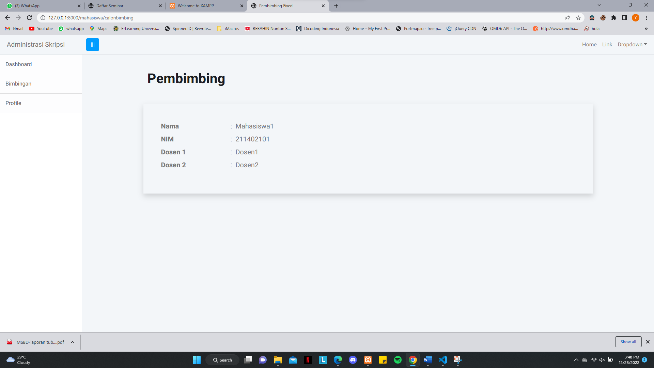
Dashboard mahasiswa



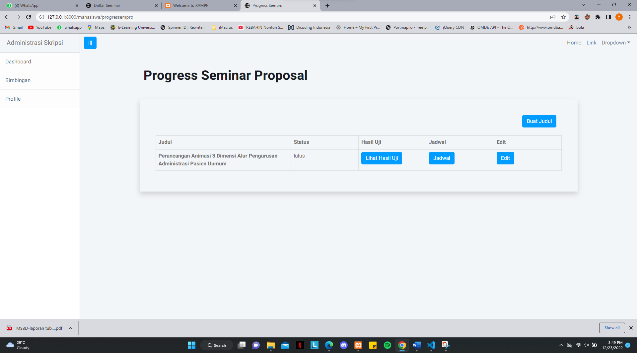
Bimbingan seminar proposal

**

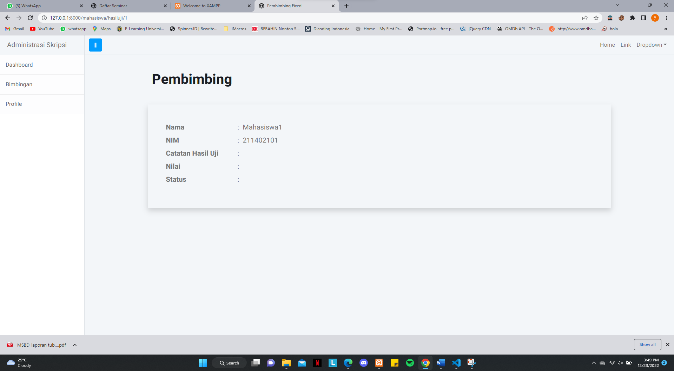
Pra seminar proposal



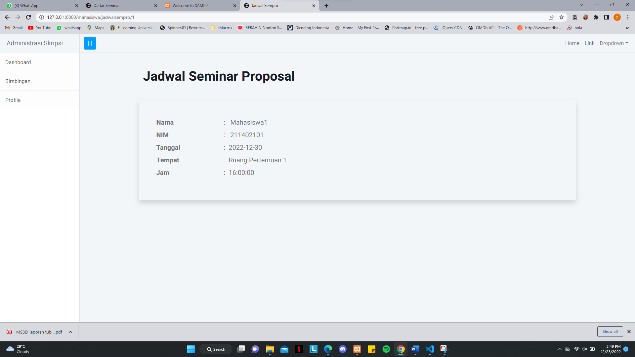
Dosen pembimbing seminar proposal



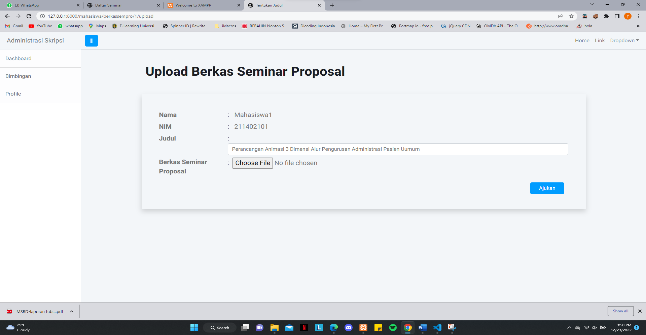
Progress seminar proposal



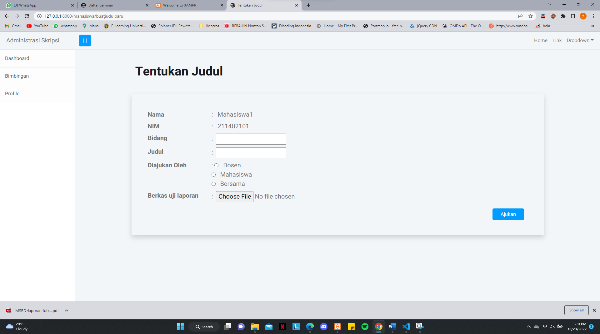
Tampilan hasil uji

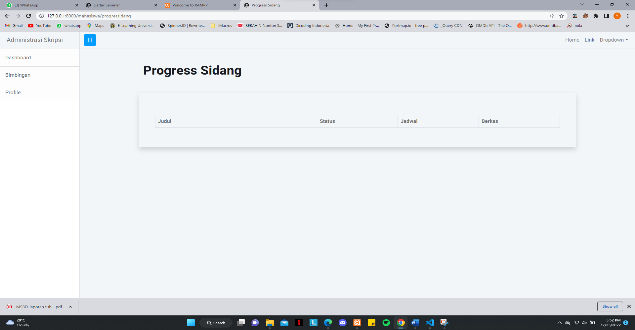
**

Tampilan jadwal



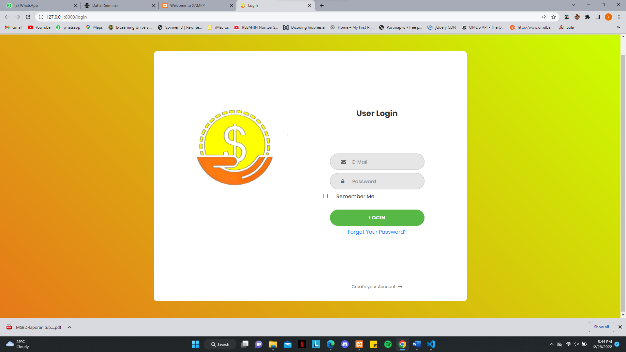
Fitur upload berkas seminar proposal

Fitur tentukan judul

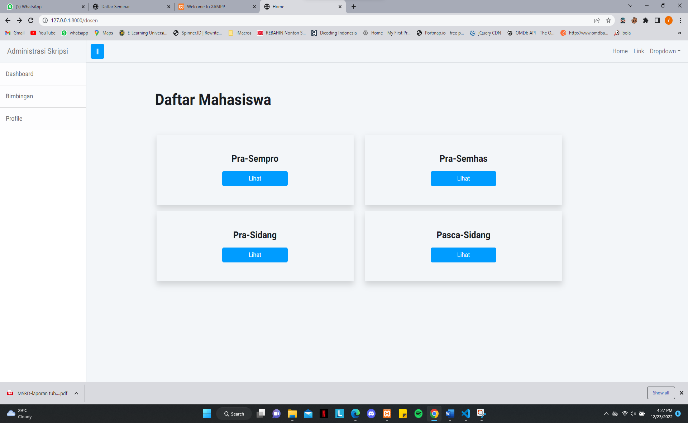


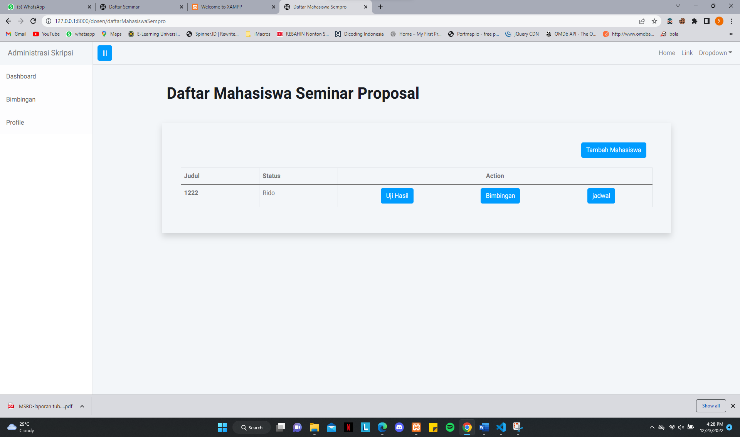
Progress sidang

1. Dosen



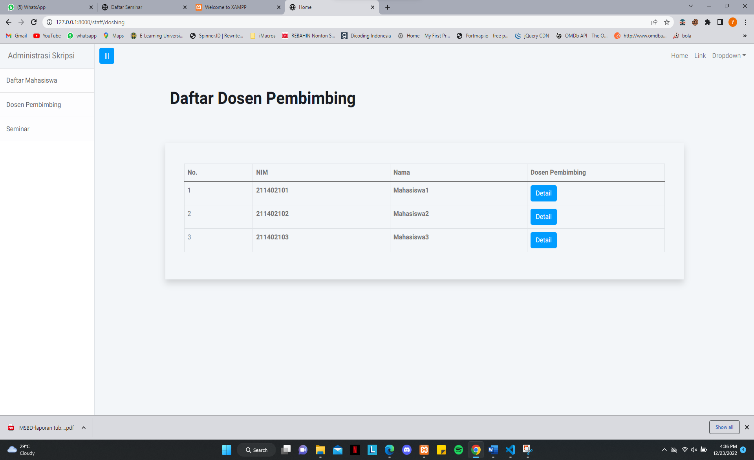
Login dosen



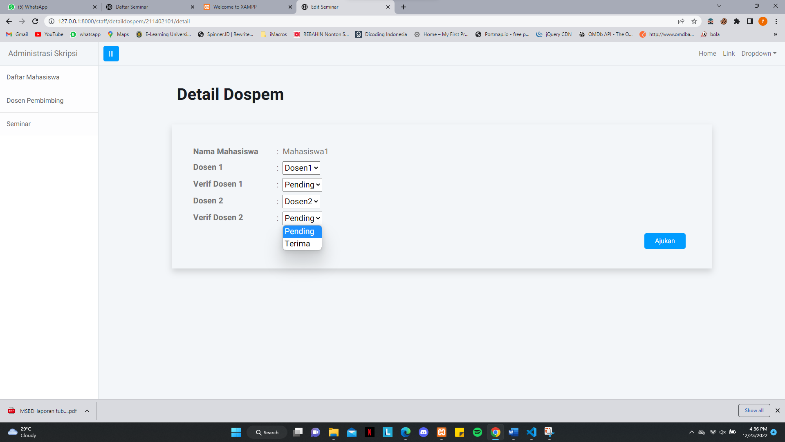
Dashboard dosen

Daftar mahasiswa seminar proposal

1. Staff



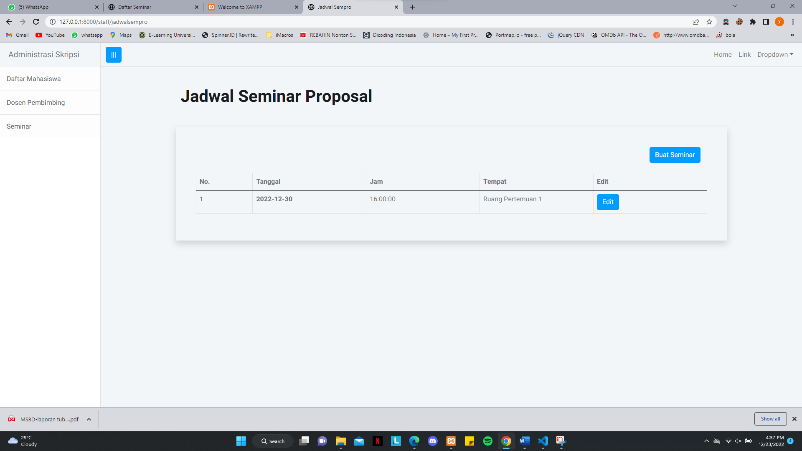
Daftar dosen pembimbing



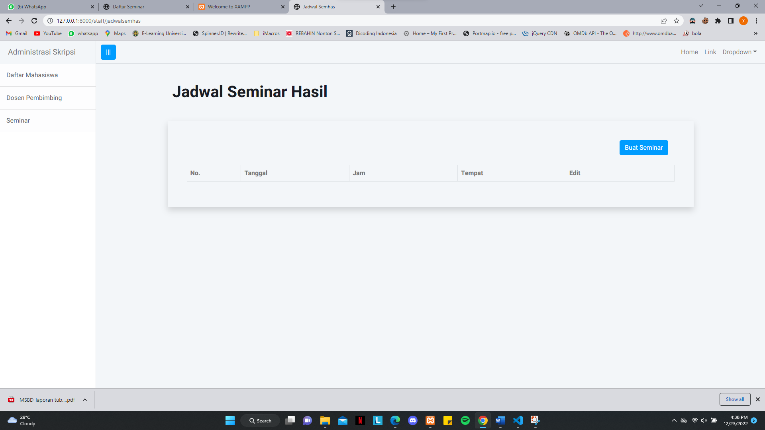
Detail dosen pembimbing



Daftar Seminar

**

Jadwal seminar proposal



Jadwal seminar hasil

# BAB IV

# PENUTUP

## **4.1 Kesimpulan**

Berdasarkan hasil tugas besar ini, telah berhasil dibuat website aplikasi administrasi skripsi Universitas Sumatera Utara. Website administrasi skripsi ini menggunakan laravel, dengan bahasa pemrograman PHP dan database MySql. Dengan adanya website ini diharapkan dapat memudahkan mahasiswa selama masa skripsi, serta mempermudah berbagai pihak prodi, meliputi dosen pembimbing, petugas prodi, dan kepala prodi dalam memonitoring atau memantau jalannya skripsi mahasiswa prodinya. Selain itu, dengan pengelolaan data skripsi yang lebih rapi, data yang otomatis tersimpan dalam dabase, dan fitur-fitur yang disediakan disistem ini menjadikan kegiatan administrasi setiap semesternya lebih praktis.

## **4.2 Saran**

Dengan adanya aplikasi berbasis website pengerjaan tugas akhir mahasiswa ini, kami menyadari sistem ini akan membutuhkan beberapa pengembangan sistem nantinya. Pengembangan ini dapat berupa design yang dibuat lebih menarik lagi, pengembangan sistem database, pengembangan fitur, dll.

# DAFTAR PUSTAKA

Aditya, Yusmardhany, Yulius, Purwo. 2020. [Pembuatan Sistem Administrasi Tugas Akhir](https://media.neliti.com/media/publications/323870-pembuatan-sistem-administrasi-tugas-akhi-9e55b4c4.pdf) [Berbasis Teknologi Informasi Pada Program Studi S1 Teknik Kimia UPN "Veteran"](https://media.neliti.com/media/publications/323870-pembuatan-sistem-administrasi-tugas-akhi-9e55b4c4.pdf) [Yogyakarta.](https://media.neliti.com/media/publications/323870-pembuatan-sistem-administrasi-tugas-akhi-9e55b4c4.pdf)

<https://www.geeksforgeeks.org/structured-query-language/> (diakses pada tanggal 12 desember 2022)

<https://id.scribd.com/embeds/263492890/content?start_page=1&view_mode=scroll&access_> [key=key-fFexxf7r1bzEfWu3HKwf](https://id.scribd.com/embeds/263492890/content?start_page=1&view_mode=scroll&access_key=key-fFexxf7r1bzEfWu3HKwf) ( diakses pada tanggal 8 Desember 2022)