

**PENGEMBANGAN APLIKASI SISTEM KERJA
PRAKTIK DAN TUGAS AKHIR PROGRAM
STUDI SISTEM INFORMASI (SITASI)**

**LAPORAN KERJA PRAKTIK
PRODI SISTEM INFORMASI ITENAS BANDUNG**

Oleh:

DIKY AKMAL FAUZI

16-2019-016



**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI NASIONAL
BANDUNG
2023**

PROGRAM STUDI SISTEM INFORMASI

**LEMBAR PENGESAHAN
LAPORAN KERJA PRAKTIK**

KATA PENGANTAR

Puji dan syukur dipanjatkan kehadiran Tuhan Yang Maha Esa atas segala limpahan rahmat dan karunia – Nya penulis dapat menyelesaikan laporan kerja praktik yang berjudul “Pembangunan Sistem Informasi Kerja Praktik Dan Tugas Akhir Institut Teknologi Nasional”.

Selama proses kerja praktik yang berlangsung, penulis mendapat bimbingan dan arahan dari berbagai entitas yang selalu memberikan dorongan, bantuan, dan juga masukan yang membangun kepada penulis. Untuk itu penulis menyampaikan terima kasih kepada berbagai pihak, yaitu.

1. Ibu Mira Musrini Barmawi S. Si., M.T. selaku ketua program studi Sistem Informasi Itenas.
2. Bapak Kurnia Ramadhan Putra, S.Kom, M.T. selaku dosen pembimbing kerja praktik.
3. Ibu Sofia Umaroh, S.Pd., M.T. selaku dosen pembimbing perusahaan.
4. Ibu Nur Fitrianti Fahrudin, S.Kom, M.T. selaku dosen penguji kerja praktik.
5. Kedua orangtua dan keluarga yang telah memberi dukungan dalam melaksanakan kerja praktik dan menyelesaikan laporan ini.
6. Semua pihak yang tidak dapat disebutkan satu – persatu.

Penulis juga menyadari sepenuhnya bahwa dalam laporan kerja praktik ini masih banyak kekurangan dan jauh dari kata sempurna, oleh sebab itu penulis mengharapkan kritik dan saran serta usulan yang bersifat membangun untuk perbaikan penulis di masa yang akan datang.

Bandung, Januari 2023

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	i
KATA PENGANTAR	ii
DAFTAR ISI.....	iii
DAFTAR GAMBAR	vi
DAFTAR TABEL.....	vii
BAB I.....	8
PENDAHULUAN	8
1.1. Latar Belakang	8
1.2. Rumusan Masalah	9
1.3. Tujuan	9
1.4. Batasan Masalah	10
1.5. Pembagian Tugas	10
1.6. Pelaksanaan Kerja Praktek.....	12
1.7. Sistematika Penulisan	13
BAB II.....	15
PROFIL INSTANSI.....	15
2.1. Sejarah dan Profil Singkat Instansi	15
2.2. Visi dan Misi.....	16
2.3. Struktur Organisasi	16
2.4. Prodi Sistem Informasi	17
2.5. Lokasi Instansi	18
BAB III	20
LANDASAN TEORI.....	20
3.1. Aplikasi Berbasis <i>Website</i>	20
3.2. Unsur-Unsur Dalam Penyediaan <i>Website</i>	20
3.3. Sistem Informasi	21
3.4. <i>Hypertext Markup Language</i> (HTML)	22
3.5. <i>Cascading Style Sheets</i> (CSS).....	23
3.6. Bootstrap	23
3.7. JavaScript (JS)	24

3.8. Laravel	25
3.9. JQuery	25
3.10. XAMPP	26
3.11. PhpMyAdmin (MySQL)	26
3.12. Postman	27
3.13. Kode Status Respon HTTP	27
3.14. UI	32
3.15. MVC	33
BAB IV	34
ANALISIS DAN PERANCANGAN	34
4.1. Deskripsi Umum Sistem	34
4.2. Analisis Kebutuhan Fungsional	34
4.3. Analisa Kebutuhan Nonfungsional	36
4.4. Perancangan Basis Data	36
4.4.1. Diagram Chen	36
4.4.2. Diagram Crowfoot	37
4.5. Perancangan Sistem	44
4.5.1. Diagram <i>Use Case</i>	44
4.5.2. <i>Use Case</i> Skenario	46
4.5.3. Activity Diagram	50
4.5.3.1. Activity Diagram Mengisi Form Pengajuan KP (Form-001)	50
4.5.3.2. Activity Diagram Melihat dan Mengunggah Surat Pengantar KP	51
4.5.3.3. Activity Diagram Mengelola Data Mahasiswa KP	52
4.5.3.4. Activity Diagram Menentukan Pembimbing dan Penguji KP	52
4.5.3.5. Activity Diagram Mengelola Data Mahasiswa TA	53
4.5.3.6. Activity Diagram Menentukan Pembimbing dan Penguji TA	54
4.5.3.7. Activity Diagram Melihat Data Form-001 KP	54
4.5.3.8. Activity Diagram Mengunggah Surat Pengantar KP	55
4.5.3.9. Activity Diagram Melihat Data Sidang KP	55
4.5.4. CRUD Matrik	56
4.5.5. Desain <i>Application Programming Interface</i>	57
4.5.6. Rancangan UI	59

4.5.7.1.	Tampilan Dashboard Kerja Praktik	59
4.5.7.2.	Tampilan <i>Dashboard</i> Sidang Kerja Praktik	60
4.5.7.3.	Tampilan <i>Form</i> Isi Daftar Kerja Praktik	60
4.5.7.4.	Tampilan <i>Form</i> Isi Daftar Sidang Kerja Praktik	61
4.5.7.	Arsitektur MVC	61
BAB V	65
IMPLEMENTASI DAN PENGUJIAN	65
5.1.	Spesifikasi Sistem	65
5.2.	Teknologi yang digunakan.....	65
5.2.1.	Spesifikasi Komputer.....	65
5.2.2.	Teknologi <i>Database</i>	66
5.2.3.	Teknologi Frontend.....	66
5.3.	Rancangan Strategi Percabangan Git.....	66
5.4.	Pengujian API	68
BAB VI	74
PENUTUP	74
6.1.	Kesimpulan	74
6.2.	Saran	74
DAFTAR PUSTAKA	76

DAFTAR GAMBAR

Gambar 2. 1. Struktur organisasi Institut Teknologi Nasional.....	17
Gambar 2. 2. Struktur organisasi prodi Sistem Informasi Itenas	17
Gambar 2. 3. Lokasi Institut Teknologi Nasional	19
Gambar 3. 1. Arsitektur MVC	33
Gambar 4. 1. Diagram <i>chen</i> perancangan <i>website</i>	37
Gambar 4. 2. Diagram <i>croowfoot</i> perancangan <i>website</i>	38
Gambar 4. 3. Diagram <i>use case</i> preancangan <i>website</i>	45
Gambar 4. 4. Diagram aktifitas mendaftar peserta tugas akhir.....	51
Gambar 4. 5. Diagram aktifitas mendaftar peserta kerja praktik	51
Gambar 4. 6. diagram aktifitas mengelola data kerja praktik	52
Gambar 4. 7. Diagram aktifitas mengelola data tugas akhir	53
Gambar 4. 8. Diagram aktifitas membimbing tugas akhir	53
Gambar 4. 9. Diagram aktifitas membimbing kerja praktik	54
Gambar 4. 10. Diagram aktifitas memberi nilai.....	54
Gambar 4. 11. Diagram aktifitas melihat data sidang proposal tugas akhir	55
Gambar 4. 12. Diagram aktifitas melihat data sidang kerja praktik.....	55
Gambar 4. 13. Component diagram tugas akhir	Error! Bookmark not defined.
Gambar 4. 14. Component diagram kerja praktik..	Error! Bookmark not defined.
Gambar 4. 15. <i>Component</i> diagram pendaftaran tugas akhir	Error! Bookmark not defined.
Gambar 4. 16. <i>Component</i> diagram pendaftaran sidang proposal tugas akhir	Error! Bookmark not defined.
Gambar 4. 17. <i>Component</i> diagram pendaftaran seminar tugas akhir	Error! Bookmark not defined.
Gambar 4. 18. <i>Component</i> diagram pendaftaran kerja praktik	Error! Bookmark not defined.
Gambar 4. 19. <i>Component</i> diagram pendaftaran sidang kerja praktik.....	Error! Bookmark not defined.
Gambar 4. 20. Desain <i>dashboard</i> tampilan proposal tugas akhir	Error! Bookmark not defined.
Gambar 4. 21. Desain <i>dashboard</i> tampilan seminar tugas akhir	Error! Bookmark not defined.
Gambar 4. 22. Desain <i>dashboard</i> tampilan <i>form</i> daftar tugas akhir	Error! Bookmark not defined.
Gambar 4. 23. Desain <i>dashboard</i> tampilan <i>form</i> sidang proposal tugas akhir	Error! Bookmark not defined.
Gambar 4. 24. Desain <i>dashboard</i> tampilan <i>form</i> sidang tugas akhir	Error! Bookmark not defined.
Gambar 4. 25. Desain <i>dashboard</i> tampilan kerja praktik	60
Gambar 4. 26. Desain <i>dashboard</i> tampilan sidang kerja praktik.....	60
Gambar 4. 27. Desain <i>dashboard</i> tampilan <i>form</i> kerja praktik.....	61

Gambar 4. 28. Desain <i>dashboard</i> tampilan form sidang kerja praktik	61
Gambar 4. 29. Arsitektur MVC pengembangan website tugas akhir dan kerja praktik	62
Gambar 5. 1. Percabangan pada GITLAB	67
Gambar 5. 2. Performa test website pengembangan website tugas akhir dan kerja praktik	73

DAFTAR TABEL

Tabel 1. 1. CRUD diagram pembagian tugas	11
Tabel 2. 1. Perbandingan sistem informasi dengan informatika.....	18
Tabel 3. 1. List kode status respon HTTP dan penjelasan	27
Tabel 4. 1. Kebutuhan fungsional <i>user</i> mahasiswa.....	35
Tabel 4. 2. Kebutuhan fungsional <i>user</i> koordinator.....	35
Tabel 4. 3. Kebutuhan fungsional <i>user</i> dosen	35
Tabel 4. 4. Kebutuhan fungsional user tata usaha.....	36
Tabel 4. 5. Kebutuhan nonfungsional sistem informasi.....	36
Tabel 4. 6. Kamus data tabel mahasiswa	38
Tabel 4. 7. Kamus data tabel dosen.....	39
Tabel 4. 8. Kamus data tabel TA.....	39
Tabel 4. 9. Kamus data tabel KP	40
Tabel 4. 10. Kamus data tabel sidang KP	40
Tabel 4. 11. Kamus data tabel sidang proposal.....	41
Tabel 4. 12. kamus data tabel seminar TA.....	41
Tabel 4. 13. Kamus data tabel sidang TA	42
Tabel 4. 14. Kamus data tabel yudisium	43
Tabel 4. 15. <i>List use case</i> yang dirancang dan penjelasan	46
Tabel 4. 16. Use Case skenario mendaftar peserta tugas akhir.....	46
Tabel 4. 17. List activity diagram yang dibuat.....	50
Tabel 4. 18. CRUD matrik yang dibuat oleh penulis	56
Tabel 4. 19. List API yang dibuat pada sistem <i>website</i>	57
Tabel 4. 20. Arsitektur MVC yang terdapat pada sistem <i>website</i>	62
Tabel 5. 1. Perangkat lunak yang digunakan untuk pengembangan <i>website</i>	65
Tabel 5. 2. Spesifikasi laptop yang digunakan.....	65
Tabel 5. 3. Teknologi basisdata yang digunakan dan versinya.....	66
Tabel 5. 4. Teknologi <i>front-end</i> yang digunakan	66
Tabel 5. 5. List modul pada pengembangan <i>website</i>	Error! Bookmark not defined.
Tabel 5. 6. Pengujian API pada <i>website</i>	68
Tabel 5. 7. List branch yang terdapat pada Gitlab	67
Tabel 5. 8. Pengujian kemajuan pengerjaan pengembangan <i>website</i>	Error! Bookmark not defined.
Tabel 5. 9. Skenario pengujian modul tugas akhir.....	Error! Bookmark not defined.
Tabel 5. 10. Skenario pengujian modul proposal tugas akhir.....	Error! Bookmark not defined.
Tabel 5. 11. Skenario pengujian modul seminar tugas akhir.....	Error! Bookmark not defined.
Tabel 5. 12. Skenario pengujian modul kerja praktik.....	Error! Bookmark not defined.

Tabel 5. 13. Skenario pengujian modul sidang praktik kerja**Error! Bookmark not defined.**

BAB I

PENDAHULUAN

1.1. Latar Belakang

Teknologi Informasi saat ini sudah berkembang dengan cepat, yang dapat membantu kepentingan masyarakat untuk mendapatkan Informasi dan pengolahan data secara tepat dan cepat. Bentuk implementasi teknologi Informasi salah satunya adalah *website* (Nurmi 2017). *Website* dapat digunakan untuk pengelolaan data, seperti memproses, mencari, menyusun, dan menyimpan data, dalam berbagai cara untuk menghasilkan Informasi yang bermanfaat dan berkualitas.

Institut Teknologi Nasional memiliki beberapa program studi salah satunya adalah program studi sistem Informasi. Program studi sistem Informasi adalah program studi yang keilmuannya menggabungkan ilmu komputer dengan bisnis dan manajemen. Pada program studi sistem Informasi mahasiswa belajar bagaimana cara mengidentifikasi kebutuhan sistem dan proses bisnis perusahaan yang berdasarkan dari data-data yang disimpan perusahaan, kemudian membuat perancangan sistem yang dibutuhkan oleh perusahaan. Selain itu di program sistem Informasi juga mempelajari bidang ilmu pemrograman, lulusan program studi diharapkan bisa mengembangkan sistem pengolahan data dari sumber sumber untuk dapat diolah dan disajikan menjadi Informasi yang bermanfaat untuk suatu instansi.

Program studi sistem Informasi saat ini membutuhkan sistem yang mampu untuk mengelola data praktik kerja dan tugas akhir. Sebelumnya sistem pendaftaran kerja praktik dan tugas akhir menggunakan proses konvensional seperti pengajuan *form* pendaftaran kerja praktik dan tugas akhir menggunakan lembar kertas yang nantinya harus di tanda tangan oleh dosen wali, kaprodi, dan pembimbing kerja praktik. Dengan demikian proses pengajuan menjadi lebih rumit dan memakan waktu.

Berdasarkan masalah tersebut maka diperlukan solusi sistem Informasi untuk pengelolaan pengajuan praktik kerja dan tugas akhir agar menjadi lebih mudah dan

menjadi lebih cepat. Solusi dari permasalahan tersebut dapat diselesaikan dengan pembuatan sistem Informasi berbasis *website*. Pembuatan sistem Informasi ini bertujuan untuk memudahkan dan mempercepat proses kegiatan kerja praktik dan tugas akhir.

Oleh karena itu dalam kegiatan kerja praktik ini yang dikerjakan oleh beberapa orang mahasiswa yang saat ini sedang mengambil mata kuliah kerja praktik yang mana fokus dari pekerjaan yang dilakukan adalah pengembangan dari sisi backend untuk modul pendaftaran tugas akhir, pendaftaran sidang proposal, seminar tugas akhir, pendaftaran kerja praktik dan sidang kerja praktik sehingga judul kerja yang diusulkan adalah “Pengembangan Sistem Kerja Praktik dan Tugas Akhir Program Studi Sistem Informasi (SITASI)”

Hal ini bisa menjadi masalah jika di kemudian hari tidak dibuat suatu sistem Informasi dikarenakan mahasiswa program studi sistem Informasi Institut Teknologi Nasional setiap tahunnya mengalami pertumbuhan yang cukup pesat.

1.2. Rumusan Masalah

Berdasarkan uraian dari latar belakang tersebut, didapatkan beberapa permasalahan yang perlu di tindak lanjuti dan di kaji lebih lanjut, yaitu:

1. Bagaimana mengidentifikasi kebutuhan sistem Informasi pengelolaan kerja praktik dan tugas akhir.
2. Bagaimana mengembangkan sistem Informasi kerja praktik dan tugas akhir yang memudahkan proses mulai dari pendaftaran sampai dengan sidang pada kegiatan kerja praktik dan tugas akhir.

1.3. Tujuan

Tujuan dari kegiatan kerja praktik ini adalah:

1. Mengidentifikasi kebutuhan sistem Informasi pengelolaan kerja praktik dan tugas akhir.
2. Mengembangkan sistem Informasi kerja praktik dan tugas akhir yang memudahkan proses mulai dari pendaftaran sampai dengan sidang pada

kegiatan kerja praktik dan tugas akhir.

1.4. Batasan Masalah

Pengembangan sistem Informasi kerja praktik dan tugas akhir memiliki beberapa batasan, yaitu:

1. Sistem Informasi yang di implementasikan memiliki beberapa *User* yaitu:
 - a. Mahasiswa, *User* ini dapat mengakses *dashboard* mahasiswa dan melakukan pendaftaran tugas akhir, sidang proposal, seminar tugas akhir, sidang tugas akhir, dan yudisium.
 - b. Dosen, *User* ini dapat mengakses dashboar dosen penguji dan dosen pembimbing. Pada *dashboard* dosen pembimbing, *User* dapat memberikan komentar pada setiap data mahasiswa yang diuji dan pada *dashboard* dosen penguji, *User* dapat menambahkan komentar dan mengubah status dari mahasiswa yang diujinya.
 - c. Koordinator, *User* ini dapat mengakses dashboar koordinator dan menentukan dosen pembimbing dan dosen penguji pada mahasiswa yang mendaftar tugas akhir, koordinator juga memiliki hak seperti *role* dosen dan mahasiswa. Pada *User* ini juga memiliki hak untuk menambahkan dan menentukan *role* setiap *User* yang berada pada *website*.
 - d. Tata Usaha, *User* ini dapat mengakses *dashboard* tata usaha dan memiliki hak untuk melihat data mahasiswa yang mendaftar tugas akhir dan menambahkan data mahasiswa yang mengikuti tugas akhir

1.5. Pembagian Tugas

Dikarenakan dalam pengerjaan kerja praktek ini dilakukan oleh 3 orang, berikut ini adalah pembagian tugas yang sudah dilakukan menggunakan tabel CRUD. CRUD adalah singkatan dari *Create*, *Read*, *Update*, dan *Delete*. Keempat hal tersebut merupakan perintah dengan peran yang esensial dalam sebuah aplikasi *database* yang sifatnya relational.

Keterangan;

1. C: *User* dapat menambahkan data

2. R: *User* dapat melihat data
3. U: *User* dapat mengubah data
4. D: *User* dapat menghapus data

Tabel 1. 1. CRUD diagram pembagian tugas

No	Modul Pengerjaan	Role				PIC
		Mahasiswa	Dosen	Koordinator	TU	
Modul Tambahan						
1	Login/Logout Tugas Akhir	CRUD	CRUD	CRUD	CRUD	Daffa
2	Website CRUD Kerja Praktek	CRUD	RU	RUD	RU	Daffa
3	Fungsional Website (Diluar Modul)	-	-	-	-	Daffa
4	Mengelola Semua Komponen Kebutuhan Website	-	-	-	-	Daffa
Modul 2: Coding Pendaftaran TA (Role: Mahasiswa)						
1	Input Data: Daftar Sidang Proposal Tugas Akhir	CR	R	R	R	Daffa
2	Input Data: Seminar Tugas Akhir	CR	R	R	R	Daffa
3	Input Data: Sidang Tugas Akhir	CR	R	R	R	Ridwan
4	Input Data: Yudisium	CR	R	R	R	Ridwan
Modul 3: Coding Data Management Pendaftaran Sidang (Role: Koor TA)						
1	Lihat Data: Daftar Mahasiswa	R	R	RUD	R	Boyke
2	Read Update Data: Proposal Tugas Akhir	RU	RU	RUD	R	Daffa
3	Read Update Data: Seminar Tugas Akhir	RU	RU	RUD	R	Daffa
4	Read Update Data: Sidang Tugas Akhir	RU	RU	RUD	R	Ridwan

No	Modul Pengerjaan	Role				PIC
		Mahasiswa	Dosen	Koordinator	TU	
5	Read Update Data: Yudisium	RU	RU	RUD	R	Ridwan
Modul 4: Coding Daftar Mahasiswa Bimbingan (Role : Dosen)						
1	Read Update Data: Proposal Tugas Akhir	RU	RUD	RU	R	Daffa
2	Read Update Data: Seminar Tugas Akhir	RU	RUD	RU	R	Boyke
3	Read Update Data: Sidang Tugas Akhir	RU	RUD	RU	R	Ridwan
Modul 5: Coding Daftar Mhs Yang Diuji (Role: Dosen)						
1	Read Update Data: Proposal Tugas Akhir	RU	RUD	R	R	Daffa
2	Read Update Data: Seminar Tugas Akhir	RU	RUD RUD	R	R	Boyke
3	Read Update Data: Sidang Tugas Akhir	RU	RUD	R	R	Ridwan
Modul 6: Coding Data Management Pendaftaran Sidang (Role : TU)						
1	Lihat Data: Daftar Proposal Mahasiswa	RU	R	RU	RUD	Daffa
2	Read Update Data: Seminar Tugas Akhir	RU	R	RU	RUD	Boyke
3	Read Update Data: Sidang Tugas Akhir	RU	R	RU	RUD	Ridwan
Modul 7:						

Keterangan warna:

1. Kuning: pekerjaan yang dikerjakan oleh penulis.

1.6. Pelaksanaan Kerja Praktek

Pelaksanaan kerja praktek ini dilakukan selama 3 bulan pada hari (Selasa - Kamis) dimulai pada tanggal 12 Juli 2022 sampai dengan 5 Oktober 2022. Bertempat di Institut Teknologi Nasional Bandung yang terletak di Jalan

PHH.Mustofa No. 23, Neglasari Kec. Cibeunying Kaler, Kota Bandung, Jawa Barat 40124.

1.7. Sistematika Penulisan

Sistematika penulisan penulis uraikan pada beberapa bab yang akan dibahas sebagai berikut:

BAB I: PENDAHULUAN

Bab ini berisikan latar belakang masalah, rumusan masalah, tujuan, tujuan penulisan laporan kerja praktek, batasan masalah dan sistematika penulisan laporan yang dilakukan penulis di program studi Sistem Informasi Institut Teknologi Nasional Bandung.

BAB II: PROFIL INSTANSI

Bab ini berisikan penjelasan tentang sejarah, profil singkat, layanan, motto, dan struktur organisasi dari program studi Sistem Informasi Institut Teknologi Nasional Bandung.

BAB III: LANDASAN TEORI

Bab ini menjelaskan teori-teori yang mendukung pembahasan mengenai pembuatan sistem Informasi beasiswa yang penulis lakukan untuk program studi Sistem Informasi Institut Teknologi Nasional Bandung.

BAB IV: ANALISIS DAN PERANCANGAN

Pada bab ini menguraikan mengenai proses – proses bagaimana perancangan sistem Informasi mahasiswa praktek kerja dan mahasiswa tugas akhir yang di laksanakan penulis di program studi Sistem Informasi Institut Teknologi Nasional Bandung.

BAB V: IMPLEMENTASI DAN PENGUJIAN

Pada bab ini menjelaskan bagaimana perancangan yang telah dibuat diimplementasikan pada sistem Informasi yang dibuat dan dilakukan pengujian pada setiap fiturnya.

BAB VI: PENUTUP

Pada bab ini berisi kesimpulan, kendala yang ada dan hal yang perlu di beri catatan serta saran – saran yang bisa diberikan pada laporan kerja praktek yang penulisa lakukan di program studi Sistem Informasi Institut Teknologi Nasional Bandung

BAB II

PROFIL INSTANSI

2.1. Sejarah dan Profil Singkat Instansi

Institut Teknologi Nasional Bandung didirikan oleh Yayasan Pendidikan Dayang Sumbi. Itenas pada awalnya merupakan Akademi, yang didirikan pada tanggal 14 Desember 1972 (SK Pengurus Yayasan Pendidikan Dayang Sumbi No. 01/Kep/DS/1972) dengan nama Akademi Teknologi Nasional (Atenas) yang terdiri dari jurusan Arsitektur, Sipil, Elektro, dan Teknik. Pada saat itu Prof R. Soetodjo, Ir., diangkat sebagai Direktur Itenas.

Pada tahun 1984 Atenas berubah menjadi Itenas (Institut Teknologi Nasional) dan pada saat itu, R. Mansoer Wiratmadja, Ir., diangkat sebagai Rektor Itenas (SK Pengurus Yayasan Pendidikan Dayang Sumbi No. 01/Kep/DS/1984 tanggal 3 Januari 1984). Hingga saat ini Itenas terus berkembang menjadi perguruan tinggi yang unggul di tataran nasional dan internasional.

Pada Tahun 2019 Itenas masuk ke dalam 100 perguruan tinggi terbaik nonvokasi terbaik nasional dengan menduduki peringkat ke-70 berdasarkan klasterisasi kinerja perguruan tinggi oleh Kemenristekdikti. Institut Teknologi Nasional memiliki Biro, salah satunya adalah Biro Kemahasiswaan dan Alumni. Biro Kemahasiswaan dan Alumni (BKA) adalah organ Itenas yang berfungsi untuk merencanakan, melaksanakan, memantau, mengevaluasi, dan meningkatkan sistem administrasi dan sistem layanan administrasi kemahasiswaan dan alumni, meliputi kegiatan kemahasiswaan, beasiswa, bimbingan dan konseling, tracer study, dan pengembangan karir. BKA dipimpin oleh Kepala Biro Kemahasiswaan dan Alumni.

Biro Kemahasiswaan dan Alumni mempunyai tugas merencanakan, mengelola, dan mengembangkan kegiatan administrasi kemahasiswaan, membina organisasi kemahasiswaan, memfasilitasi mahasiswa untuk konseling, CDC (Career Development Center) dan membina hubungan alumni dengan institut. Pada 8 pelaksanaannya Biro Kemahasiswaan dan Alumni berkoordinasi dengan seluruh

Wakil Rektor. Selain itu Biro Kemahasiswaan dan Alumni wajib melaporkan seluruh pelaksanaan kegiatan Biro kepada Rektor.

2.2. Visi dan Misi

Pada dasarnya visi dijadikan sebagai acuan, gambaran akan situasi dan karakteristik mengenai tujuan dari suatu individu, organisasi atau perusahaan. Sedangkan, misi adalah suatu pernyataan tentang apa yang harus dikerjakan dalam usaha mewujudkan suatu visi yang telah dibuat.

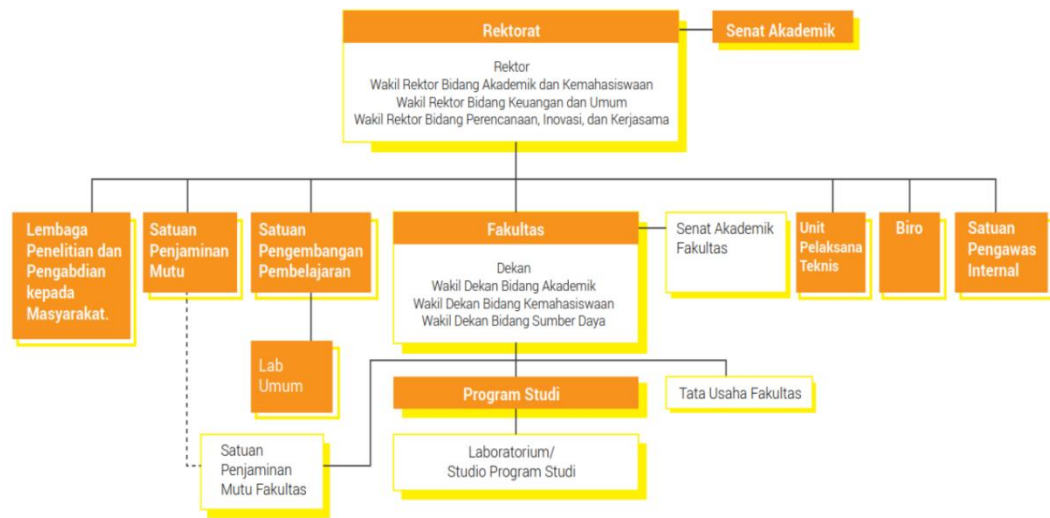
Visi dari Institut Teknologi Nasional Bandung adalah *“Itenas menjadi perguruan tinggi terkemuka di bidang teknologi, sains, dan seni yang berperan aktif dalam pembangunan berkelanjutan di lingkup nasional dan global, berlandaskan nilai – nilai integritas, kualitas, dan inovasi yang tinggi.”*

Berikut merupakan misi dari Institut Teknologi Nasional Bandung, yaitu:

1. Membangun karakter bangsa melalui penyelenggaraan kegiatan tridarma perguruan tinggi yang berkualitas.
2. Menghasilkan lulusan yang unggul dalam kecerdasan intelektual serta memiliki integritas dan moral yang tinggi.
3. Menghasilkan karya ilmiah dan karya inovatif yang bermanfaat bagi kesejahteraan masyarakat.
4. Mengembangkan infrastruktur dan sistem manajemen yang berbasis sains dan teknologi untuk menumbuhkan suasana akademik yang kondusif.

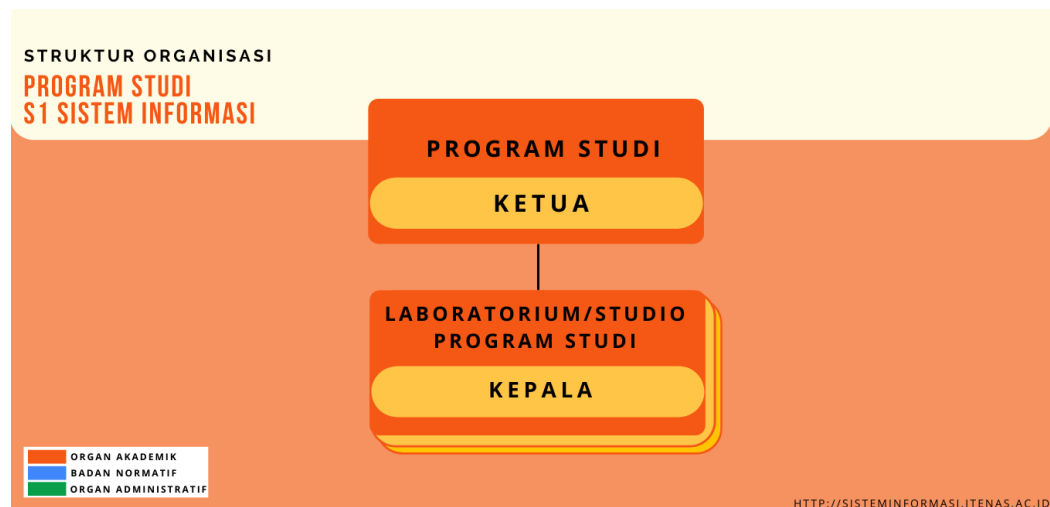
2.3. Struktur Organisasi

Pada bagian ini menampilkan bagaimana struktur organisasi dari Institut Teknologi Nasional Bandung:



Gambar 2. 1. Struktur organisasi Institut Teknologi Nasional

Kemudian berikut ini adalah struktur organisasi dari Prodi Sistem Informasi yang Kampus Institut Teknologi Nasional Bandung:



Gambar 2. 2. Struktur organisasi prodi Sistem Informasi Itenas

2.4. Prodi Sistem Informasi

Program Studi Sistem Informasi Itenas memiliki kurikulum yang mewajibkan mahasiswanya untuk mempelajari berbagai konsep dan praktik menganalisis, mendesain dan menerapkan sistem Informasi dalam organisasi. Di samping itu juga mahasiswa harus mempelajari proses penciptaan, pengolahan, penyimpanan, pendistribusian, dan pengawasan data/Informasi/*knowledge* di lingkungan

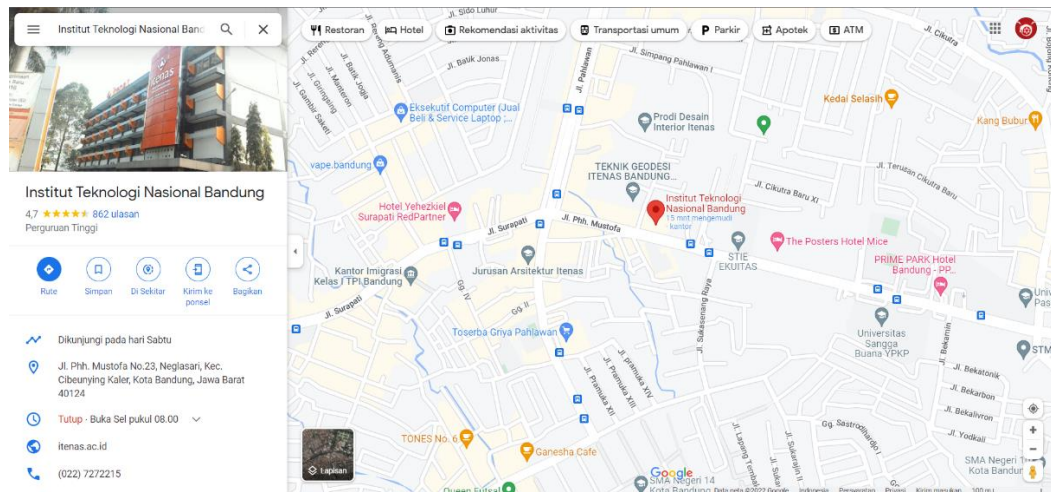
organisasi. Lulusan Program Studi Sistem Informasi dipastikan mampu berpikir secara analitis kritis dalam menentukan solusi sistem Informasi berbasis kompetensi rekayasa basis data, sistem analis, tata kelola teknologi Informasi dan perencanaan teknologi Informasi. Selama perkuliahan mahasiswa juga akan mengembangkan keterampilan dalam pemrograman (pemrograman *web*, pemrograman berorientasi obyek, pemrograman basis data rekayasa perangkat lunak), manajemen proyek, analisa dan desain sistem aplikasi, *Enterprise Resource Planning*, E bisinis, tata kelola teknologi Informasi. Keterampilan dan pengetahuan tersebut akan mendukung kemampuan mahasiswa dalam menentukan solusi IT yang optimal sesuai kebutuhan organisasi. Perbedaan informatika dan sistem Informasi. Lulusan informatika mampu membuat aplikasi komputer yang mampu meningkatkan kinerja komputer, dan aplikasi. Lulusan informatika dituntut untuk memiliki tingkat pemahaman matematika dan pemrograman yang sangat tinggi. Lulusan sistem Informasi mampu mengusulkan, mengevaluasi sistem Informasi yang selaras dengan target bisnis dan strategi organisasi. Profesi lulusan Sistem Informasi ini ditunjang oleh kemampuan berpikir secara analitis yang tajam dan kemampuan pemrograman, matematika yang memadai yang tidak terlalu kompleks. Perbandingan kandungan kurikulum sistem Informasi dan informatika:

Tabel 2. 1. Perbandingan sistem informasi dengan informatika

Kemampuan, Pengetahuan dan Keterampilan	Teknik Pemrograman	Analisa dan Desain Sistem	Perencanaan Teknologi Informasi untuk Organisasi	Matematika
Program Studi Informatika	Tinggi	Rendah	Rendah	Tinggi
Program Studi Sistem Informasi	Medium	Tinggi	Tinggi	Rendah

2.5. Lokasi Instansi

Institut Teknologi Nasional (Itenas) Bandung beralamat di Jl. PH. H. Mustopa No. 23 Bandung, Jawa Barat, Indonesia Kode Pos 40135 Bandung serta mempunyai *website* <https://www.itenas.ac.id/>. Berikut ini adalah lokasi yang didapati dari GoogleMaps.



Gambar 2. 3. Lokasi Institut Teknologi Nasional

BAB III

LANDASAN TEORI

Pada bab ini dibahas mengenai teori – teori pendukung yang digunakan dalam melakukan pembangunan *website* Beasiswa Itenas. Teori – teori ini terdiri dari teori tentang pembangunan *website*.

3.1. Aplikasi Berbasis Website

Aplikasi berbasis *website* merupakan salah satu aplikasi yang bisa dilakukan akses melalui internet sehingga Informasi dapat diakses dengan waktu dan tempat yang tidak ditentukan (Anita, Susanto, and Wahyudi 2020). Pada zaman ini, banyak perusahaan – perusahaan yang mengaplikasikan aplikasi berbasis *website* dalam merencanakan program untuk mengelola perusahaan, hingga pengolahan data.

Aplikasi berbasis *website* bisa digunakan untuk berbagai macam tujuan yang berbeda. Sebagai contoh, aplikasi berbasis *website* dapat digunakan untuk membantu mendata serta mencatat Informasi stok barang, transaksi penjualan, transaksi pembelian, membuat *invoice* dan memberikan cara yang mudah dalam penyimpanan data di *database*.

Selain fungsi – fungsi tersebut, salah satu keunggulan dari aplikasi berbasis *website* adalah bahwa aplikasi yang dibuat berbasis *website* dapat dikatakan ‘ringan’ dan dapat diakses dengan cepat melalui browser dan koneksi internet ke *server*. Dengan kata lain, pengguna dapat mengakses data atau Informasi perusahaan mereka melalui laptop, *smartphone*, atau bahkan komputer PC di rumah mereka sendiri dengan mudah dan efisien. Hal tersebut berbanding terbalik dengan aplikasi desktop dimana pengguna harus mengunduh perangkat lunak atau aplikasi yang diperlukan hanya untuk mengakses data / Informasi.

3.2. Unsur-Unsur Dalam Penyediaan Website

Untuk menyediakan sebuah *website*, maka kita harus menyediakan unsur – unsur penunjang, seperti:

1. Nama Domain (*Domain Name* / URL)

Nama domain atau biasa disebut dengan *Domain Name* atau URL adalah alamat unit di internet yang digunakan untuk mengidentifikasi sebuah *website*. Sebagai contoh : <http://itenas.ac.id>.

2. Web Hosting

Web Hosting dapat diartikan sebagai ruangan yang terdapat didalam harkdisk tempat meyimpan berbagai data, gambar, file – file dan lain sebagainya yang akan ditampilkan didalam *website*. Besarnya data yang bisa dimasukan tergantung dari besarnya web hosting yang disewa / dipunyai (Krishna Patel, 2018).

3. Bahasa Program (*Scripts Program*)

Bahasa program merupakan bahasa yang digunakan untuk menerjemahkan setiap perintah dalam *website* pada saat diakses. Jenis bahasa program sangat menentukan interaktif, statis atau dinamisnya sebuah *website*.

4. Bahasa Program (*Scripts Program*)

Setelah melakukan penyewaan domain *name* dan web hosting serta penguasaan Bahasa Program, unsur *website* yang penting selanjutnya adalah desain. Desain *website* menentukan kualitas dan keindahan sebuah *website*.

5. Publikasi *Website*

Keberadaan situs tidak ada gunanya tanpa dikunjungi atau dikenal oleh orang – orang atau pengunjung internet. Karena efektif tidaknya situs sangat tergantung dari besarnya pengunjung dan komentar yang masuk. Untuk mengenalkan situs kepada masyarakat memerlukan apa yang disebut publikasi atau promosi *website*

3.3. Sistem Informasi

Sistem Informasi adalah sistem terpadu antara manusia – mesin untuk menyediakan Informasi untuk mendukung sistem operasi, manajemen, analisa dan

pengambilan keputusan dalam organisasi (Nurmi 2017). Dari definisi di atas terlihat bahwa pada hakikatnya sistem Informasi adalah suatu sistem yang berkaitan dengan 12 pengumpulan, penyimpanan dan pemrosesan data, baik yang dilakukan secara manual dan atau dengan bantuan komputer dalam proses pengambilan keputusan. Dengan demikian sistem Informasi tidak selalu harus berbasis komputer.

Pengolahan data dalam sistem Informasi dilakukan secara manual atau kombinasi antara sistem manual dengan sistem berbasis komputer. Keberadaan komputer pada sistem Informasi pada dasarnya tidak mutlak. Akan tetapi komputer dengan segenap kemampuannya dalam memproses data, akan meningkatkan efektivitas, produktivitas, serta efisiensi suatu sistem Informasi (Setiawan, 2016).

3.4. *Hypertext Markup Language (HTML)*

Hypertext Markup Language (HTML) adalah bahasa komputer yang di gunakan untuk membuat suatu halaman situs (Febio 2011). Di dalam skema tambahan HTML terdapat kekuatan untuk membuat aplikasi – aplikasi *client – server*, multimedia, *form* interaktif. Sebenarnya, HTML merupakan dokumen ASCII atau teks biasa, yang dirancang untuk tidak tergantung pada satu sistem operasi tertentu. Secara teknis, HTML didefinisikan sebagai *Standard Generalized Markup Language (SGML)*. Sebuah dokumen HTML dapat dikatakan contoh sebuah dokumen SGL.

HTML adalah kumpulan elemen yang digunakan untuk mendefinisikan berbagai bagian dokumen, seperti *header*, *footer*, *heading*, dan badan teks (Krishna Patel, 2018). Pada dasarnya, ketika sebuah dokumen diurai, browser membuat representasi hierarkis dari elemen – elemen di dalamnya, yang dikenal sebagai *Model Document Object Model (DOM)*. *Model* inilah yang memungkinkan dokumen untuk di-*render* dengan baik ke dalam halaman *website*. *Model* ini juga memungkinkan penggunaan bahasa *script*, seperti JavaScript, untuk memanipulasi elemen halaman untuk membuat dinamisme.

Dokumen HTML terdiri dari dua bagian utama, yaitu bagian *Header* dan *body section*, dan masing – masing didefinisikan oleh dan Seluruh dokumen HTML

ditentukan oleh elemen . Selain itu, dokumen HTML 13 biasanya dimulai dengan deklarasi tipe dokumen, yang menentukan bahwa dokumen tersebut adalah dokumen HTML.

3.5. *Cascading Style Sheets* (CSS)

CSS merupakan bahasa *style sheet* yang dipergunakan untuk memodifikasi / mengatur tampilan dokumen (Rahmatika, Pradana, and Bachtiar 2020). CSS menggunakan *Model* kotak, di mana dokumen yang sudah ditandai dipandang sebagai hierarki kotak persegi panjang, dengan masing – masing kotak mewakili elemen dengan dokumen. Ukuran dari kotak tersebut bergantung dengan ukuran konten yang diberikan oleh elemen. Dengan mengontrol properti dan isi kotak dokumen, tampilan dari halaman *website* yang telah dibuat dapat kita kontrol dengan baik.

Terdapat 3 macam CSS di dalam kode HTML, yaitu *external*, *internal*, atau *inline*. Terdapat perbedaan mencolok diantara ketiga macam CSS tersebut yaitu *inline* CSS ditempatkan langsung di dalam konten HTML. Penempatan CSS secara *inline* hanya akan mempengaruhi satu bagian baris kode saja. Berbeda halnya dengan *External* CSS, *External* CSS ditempatkan pada baris kode CSS menggunakan file .css. Lalu yang terakhir Internal CSS sekilas mirip dengan *inline* CSS, namun di dalam penggunaan CSS secara internal, baris kode HTML dituliskan di bagian atas (*header*) file HTM

3.6. Bootstrap

Bootstrap adalah *framework* yang mengatur dan mengelola *layout website*. HTML mengatur konten dan struktur halaman *website* (Putra 2020). Salah satu masalah dengan desain HTML dasar adalah bahwa halaman *website* mungkin terlihat agak berbeda di browser ataupun perangkat – perangkat lain yang berbeda. Oleh karena itu, kita mungkin perlu memodifikasi kode sesuai dengan browser atau perangkat tertentu. Namun, masalah tersebut dapat diselesaikan dengan mudah dengan menggunakan Bootstrap.

Bootstrap menggunakan HTML, CSS dan JavaScript untuk desain *website*. Bootstrap menyertakan beberapa kelas yang telah didefinisikan sebelumnya untuk mengatur tata letak yang mudah contohnya seperti *Dropdown*, *Button*, dan masih 14 banyak lagi. Selain itu Bootstrap memiliki sifat yang *responsive* dan dapat berubah secara otomatis dengan menyesuaikan dengan perangkat seperti contohnya *smartphone*, laptop, dan masih banyak lagi (Wehrens et al., 2000).

Bootstrap merupakan *framework front – end* gratis yang belakangan ini semakin populer dikalangan *front – end developer*. Selain karena mudah digunakan, *framework* ini juga dapat membantu kita mempersingkat pekerjaan *developer* sehingga tidak perlu menulis *script* secara manual dan berulang kali. Bootstrap juga fleksibel dan mendukung hampir semua keperluan pengembangan *front – end website*. Fungsi bootstrap yang tidak terbatas pada tampilan *website* semakin mempercantik tampilan dilengkapi dengan fitur *responsive* agar halaman *website* bisa bekerja secara optimal di semua ukuran layar perangkat.

3.7. JavaScript (JS)

JavaScript adalah bahasa pemrograman situs web yang merupakan bahasa pemrograman sisi klien (Renaldi and Anggoro 2020). Bahasa pemrograman sisi klien adalah jenis bahasa pemrograman yang diproses oleh klien. Aplikasi *client* yang dimaksud adalah web browser seperti Google Chrome, Mozilla Firefox, Opera Mini. JavaScript digunakan dalam pengembangan situs web agar lebih dinamis dan interaktif.

JavaScript pertama kali dikembangkan pada tahun 1994. Meskipun memiliki nama yang hampir serupa dengan Java, JavaScript berbanding terbaik dengan Bahasa Pemrograman Java. Untuk cara penulisannya, JavaScript dapat disisipkan di dalam dokumen HTML ataupun dijadikan dokumen tersendiri yang kemudian diasosiasikan dengan dokumen lain yang dituju. JavaScript sendiri mengimplementasikan fitur yang dirancang untuk mengendalikan bagaimana sebuah halaman *website* berinteraksi dengan penggunaannya.

Bahasa ini pertama kali muncul di browser Netscape 2 dengan nama LiveScript pada pertengahan 1990-an. Secara teknis, JavaScript adalah merek dagang dari Sun's Microsystems, yang mengimplementasikan standar yang disebut ECMAScript. Terlepas dari namanya, JavaScript tidak terkait langsung dengan bahasa pemrograman Java.

3.8. Laravel

Laravel adalah *framework* berbasis bahasa pemrograman PHP yang bisa digunakan untuk membantu proses pengembangan sebuah *website* agar lebih maksimal (Herdiansah, Borman, and Maylinda 2021). Dengan menggunakan Laravel, *website* yang dihasilkan akan lebih dinamis.

Framework Laravel menggunakan struktur MVC (*Model View Controller*). MVC merupakan *Model* aplikasi yang memisahkan antara data dan tampilan berdasarkan komponen aplikasi. Dengan adanya *Model* MVC, pengguna Laravel menjadi lebih mudah dalam mempelajari Laravel. Serta menjadikan proses pembuatan aplikasi berbasis *website* menjadi lebih cepat.

Laravel juga menyediakan fitur bawaan yang lengkap, salah satunya adalah fitur otentikasi. *Framework* yang satu ini cenderung berfokus pada level *end-User*. Di mana *framework* ini memiliki keunggulan pada kesederhanaannya, baik dalam sisi penulisan kode maupun tampilannya. Meskipun demikian, *framework* Laravel tetap dapat digunakan untuk membuat aplikasi berbasis *website* dengan fitur yang lengkap.

3.9. JQuery

JQuery adalah *library* JavaScript yang berisi kumpulan kode JavaScript atau fungsi program yang siap untuk membuat aplikasi web (Putra 2020). Untuk membangun aplikasi web yang menggunakan JQuery, cukup panggil fungsi JavaScript yang disertakan dalam *library* JQuery. Proses pemanggilan fungsi-fungsi tersebut memudahkan programmer aplikasi web untuk membuat antarmuka aplikasi web yang lebih interaktif. Kode program yang dihasilkan lebih sederhana

dan lebih sederhana, sehingga lebih mudah dipahami. JQuery pertama kali diterbitkan oleh John Resig pada tahun 2006. Banyak perusahaan besar sekarang menggunakan JQuery dalam aplikasi web mereka. Perusahaan-perusahaan ini termasuk Google, Microsoft, Oracle, dan Dell.

3.10. XAMPP

XAMPP adalah *software* web server Apache yang dapat menggunakan *database server* MySQL dan mendukung pemrograman PHP (Anggraini, Pasha, and Setiawan 2020). XAMPP adalah 16 perangkat lunak gratis yang mudah digunakan yang mendukung instalasi di Linux dan Windows. Program aplikasi XAMPP bertindak sebagai *server* lokal yang memproses berbagai jenis data situs web yang sedang dikembangkan.

XAMPP adalah *software* web server Apache yang dapat menggunakan *database server* MySQL dan mendukung pemrograman PHP. XAMPP adalah 16 perangkat lunak gratis yang mudah digunakan yang mendukung instalasi di Linux dan Windows. Program aplikasi XAMPP bertindak sebagai *server* lokal yang memproses berbagai jenis data situs web yang sedang dikembangkan

3.11. PhpMyAdmin (MySQL)

PhpMyadmin adalah perangkat lunak gratis yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk mengelola MySQL melalui World Wide Web (Ramadhan and Mukhaiyar 2020). phpMyAdmin mendukung berbagai operasi MySQL termasuk (*database*, tabel, *fields*), hubungan, indeks, pengguna, izin dan banyak lagi. Pada dasarnya, mengelola *database* menggunakan MySQL harus memasukkan dan menjalankan baris perintah yang sesuai untuk setiap tujuan tertentu.

Jika seseorang ingin membuat *database*, tentu harus memasukkan baris perintah yang sesuai untuk membuat *database*. Tentu saja ini sangat sulit karena seseorang harus mengingat dan memasukkan perintah satu demi satu. Saat ini sudah banyak *software* yang dapat digunakan untuk mengelola *database* MySQL, salah satunya adalah phpMyAdmin. PhpMyAdmin dapat dengan mudah membuat *database*,

membuat tabel, memasukkan data, dan lainnya tanpa harus mengingat baris perintah. PHP MyAdmin adalah bagian untuk mengelola *database* MySQL di dalam komputer.

3.12. Postman

Postman adalah *toolchain* penting bagi pengembang API untuk berbagi, uji coba, dokumentasi dan memantau API (Herman and Charles 2022). Lebih dari 3 juta insinyur dan pengembang di seluruh dunia menggunakan Postman untuk membangun perangkat lunak yang terhubung melalui API-cepat, mudah dan akurat.

Ide untuk Postman muncul ketika para pendiri bekerja bersama-sama, dan frustrasi dengan alat yang ada untuk pengujian API. Mereka merasa harus ada bahasa yang lebih baik bagi para pengembang untuk berkomunikasi tentang API. Hal ini menyebabkan penciptaan Postman. Postman membantu pengembang membangun, menguji, dan mendokumentasikan lebih cepat.

3.13. Kode Status Respon HTTP

Status *request* pada HTTP berupa angka 3 digit yang dikirimkan. Ada 5 jenis kelompok kode respons : 100- 199 untuk respon Informasi, 200-299 untuk respon sukses, 300-399 untuk respons *request*, 400-499 untuk respon error pada *client*, 500-599 untuk error pada *server* (Implementasi 2022). Berikut adalah list kode setiap kelasnya.

Tabel 3. 1. List kode status respon HTTP dan penjelasan

Kode	Deskripsi
<i>100 Continue</i>	Respons sementara ini menunjukkan bahwa klien harus melanjutkan permintaan atau mengabaikan respons jika permintaan sudah selesai.
<i>101 Switching Protocols</i>	Kode ini dikirim sebagai respons terhadap <i>header</i> permintaan <i>Upgrade</i> dari klien dan mengindikasikan protokol yang digunakan <i>server</i> untuk beralih.
<i>102 Processing</i>	Kode ini menunjukkan bahwa <i>server</i> telah menerima dan sedang memproses permintaan, tetapi belum ada respons yang tersedia.

Kode	Deskripsi
<i>103 Early Hints</i>	Kode status ini terutama dimaksudkan untuk digunakan dengan <i>header</i> Link, membiarkan agen pengguna memulai pramuat sumber daya sementara <i>server</i> menyiapkan respons.
<i>200 OK</i>	permintaan berhasil. Makna hasil "sukses" tergantung pada metode HTTP: <ul style="list-style-type: none"> a. GET: Sumber daya telah diambil dan dikirimkan dalam badan pesan. b. HEAD: <i>Header</i> representasi disertakan dalam respons tanpa badan pesan. c. PUT atau POST: Sumber daya yang menggambarkan hasil tindakan ditransmisikan dalam badan pesan. d. TRACE: Badan pesan berisi pesan permintaan seperti yang diterima oleh <i>server</i>.
<i>201 Created</i>	Permintaan berhasil, dan sumber daya baru dibuat sebagai hasilnya. Ini biasanya merupakan respons yang dikirim setelah permintaan POST, atau beberapa permintaan PUT.
<i>202 Accepted</i>	Permintaan telah diterima tetapi belum ditindaklanjuti. Ini noncommittal, karena tidak ada cara dalam HTTP untuk kemudian mengirim respons asinkron yang menunjukkan hasil permintaan. Ini dimaksudkan untuk kasus di mana proses atau <i>server</i> lain menangani permintaan, atau untuk pemrosesan batch.
<i>203 Non-Authoritative Information</i>	Kode respons ini berarti metadata yang dikembalikan tidak persis sama dengan yang tersedia dari <i>server</i> asal, tetapi dikumpulkan dari salinan lokal atau pihak ketiga. Ini sebagian besar digunakan untuk mirror atau backup dari sumber daya lain. Kecuali untuk kasus spesifik itu, respons 200 OK lebih disukai daripada status ini.
<i>204 No Content</i>	Tidak ada konten yang harus dikirim untuk permintaan ini, tetapi tajuknya mungkin berguna. Agen pengguna dapat memperbarui <i>header</i> yang di-cache untuk sumber daya ini dengan yang baru.
<i>205 Reset Content</i>	Memberitahu agen pengguna untuk mengatur ulang dokumen yang mengirim permintaan ini.
<i>206 Partial Content</i>	Kode respons ini digunakan ketika <i>header</i> Range dikirim dari klien untuk meminta hanya sebagian dari sumber daya.
<i>207 Multi-Status</i>	Menyampaikan Informasi tentang beberapa sumber daya, untuk situasi di mana beberapa kode status mungkin sesuai.
<i>208 Already Reported</i>	Digunakan di dalam elemen respons <dav:propstat> untuk menghindari penghitungan berulang kali anggota internal dari beberapa binding ke koleksi yang sama.
<i>226 IM Used</i>	<i>Server</i> telah memenuhi permintaan GET untuk sumber daya, dan responsnya adalah representasi dari hasil satu atau lebih manipulasi instance yang diterapkan pada instance saat ini.
<i>300 Multiple Choices</i>	Permintaan memiliki lebih dari satu kemungkinan respons. Agen pengguna atau pengguna harus memilih salah satunya. (Tidak ada cara standar untuk memilih salah satu respons, tetapi tautan HTML ke kemungkinan-kemungkinan yang disarankan sehingga pengguna dapat memilih.)

Kode	Deskripsi
<i>301 Moved Permanently</i>	URL sumber daya yang diminta telah diubah secara permanen. URL baru diberikan dalam respons.
<i>302 Found</i>	Kode respons ini berarti bahwa URI sumber daya yang diminta telah diubah untuk sementara. Perubahan lebih lanjut dalam URI mungkin dilakukan di masa mendatang. Oleh karena itu, URI yang sama ini harus digunakan oleh klien dalam permintaan di masa mendatang.
<i>303 See Other</i>	<i>Server</i> mengirim respons ini untuk mengarahkan klien untuk mendapatkan sumber daya yang diminta di URI lain dengan permintaan GET.
<i>304 Not Modified</i>	Ini digunakan untuk tujuan caching. Ini memberi tahu klien bahwa respons belum dimodifikasi, sehingga klien dapat terus menggunakan versi cache yang sama dari respons.
<i>305 Use Proxy</i>	Didefinisikan dalam versi sebelumnya dari spesifikasi HTTP untuk menunjukkan bahwa respons yang diminta harus diakses oleh proxy. Ini telah tidak digunakan lagi karena masalah keamanan terkait konfigurasi in-band proxy.
<i>306 unused</i>	Kode respons ini tidak lagi digunakan; hanya dicadangkan. Kode ini digunakan dalam versi sebelumnya dari spesifikasi HTTP/1.1.
<i>307 Temporary Request</i>	<i>Server</i> mengirimkan respons ini untuk mengarahkan klien untuk mendapatkan sumber daya yang diminta di URI lain dengan metode yang sama yang digunakan dalam permintaan sebelumnya. Ini memiliki semantik yang sama dengan kode respons HTTP 302 Found, dengan pengecualian bahwa agen pengguna tidak boleh mengubah metode HTTP yang digunakan: jika POST digunakan dalam permintaan pertama, POST harus digunakan dalam permintaan kedua.
<i>308 Permanent Request</i>	Ini berarti bahwa sumber daya sekarang berada secara permanen di URI lain, yang ditentukan oleh <i>header</i> Location: Tajuk Respons HTTP. Ini memiliki semantik yang sama dengan kode respons HTTP 301 Moved Permanently, dengan pengecualian bahwa agen pengguna tidak boleh mengubah metode HTTP yang digunakan: jika POST digunakan dalam permintaan pertama, POST harus digunakan dalam permintaan kedua.
<i>400 Bad Request</i>	<i>Server</i> tidak dapat atau tidak akan memproses permintaan karena sesuatu yang dianggap sebagai kesalahan klien (misalnya, sintaks permintaan yang cacat, framing pesan permintaan yang tidak valid, atau routing permintaan yang menipu).
<i>401 Unauthorized</i>	Meskipun standar HTTP menetapkan "tidak sah", secara semantik respons ini berarti "tidak terautentikasi". Artinya, klien harus mengotentikasi dirinya sendiri untuk mendapatkan respons yang diminta.
<i>402 Payment Required Experimental</i>	Kode respons ini dicadangkan untuk penggunaan di masa mendatang. Tujuan awal untuk membuat kode ini adalah menggunakannya untuk sistem pembayaran digital, namun kode status ini sangat jarang digunakan dan tidak ada konvensi standar.
<i>403 Forbidden</i>	Klien tidak memiliki hak akses ke konten; yaitu, tidak sah, sehingga <i>server</i> menolak untuk memberikan sumber daya yang

Kode	Deskripsi
	diminta. Tidak seperti 401 <i>Unauthorized</i> , identitas klien diketahui oleh <i>server</i> .
404 <i>Not Found</i>	<i>Server</i> tidak dapat menemukan sumber daya yang diminta. Di browser, ini berarti URL tidak dikenali. Dalam API, ini juga dapat berarti bahwa titik akhir valid tetapi sumber daya itu sendiri tidak ada. <i>Server</i> juga dapat mengirim respons ini alih-alih 403 <i>Forbidden</i> untuk menyembunyikan keberadaan sumber daya dari klien yang tidak sah. Kode respons ini mungkin yang paling terkenal karena sering terjadi di web.
405 <i>Method Not Allowed</i>	Metode permintaan diketahui oleh <i>server</i> tetapi tidak didukung oleh sumber daya target. Misalnya, API mungkin tidak mengizinkan pemanggilan DELETE untuk menghapus sumber daya.
406 <i>Not Acceptable</i>	Respons ini dikirim ketika <i>server</i> web, setelah melakukan negosiasi konten yang digerakkan oleh <i>server</i> , tidak menemukan konten apa pun yang sesuai dengan kriteria yang diberikan oleh agen pengguna.
407 <i>Proxy Authentication Required</i>	Ini mirip dengan 401 <i>Unauthorized</i> tetapi otentikasi diperlukan untuk dilakukan oleh proxy.
408 <i>Request Timeout</i>	Respons ini dikirim pada koneksi idle oleh beberapa <i>server</i> , bahkan tanpa permintaan sebelumnya oleh klien. Ini berarti bahwa <i>server</i> ingin menutup koneksi yang tidak digunakan ini. Respons ini lebih banyak digunakan karena beberapa browser, seperti Chrome, Firefox 27+, atau IE9, menggunakan mekanisme pra-koneksi HTTP untuk mempercepat penjelajahan. Juga perhatikan bahwa beberapa <i>server</i> hanya mematikan koneksi tanpa mengirimkan pesan ini.
409 <i>Conflict</i>	Respons ini dikirim ketika permintaan bertentangan dengan keadaan <i>server</i> saat ini.
410 <i>Gone</i>	Respons ini dikirim ketika konten yang diminta telah dihapus secara permanen dari <i>server</i> , tanpa alamat penerusan. Klien diharapkan untuk menghapus <i>cache</i> dan tautan mereka ke sumber daya. Spesifikasi HTTP bermaksud kode status ini digunakan untuk "layanan promosi dengan waktu terbatas". API seharusnya tidak merasa terdorong untuk menunjukkan sumber daya yang telah dihapus dengan kode status ini.
411 <i>Length Required</i>	<i>Server</i> menolak permintaan karena bidang <i>header Content-Length</i> tidak didefinisikan dan <i>server</i> memerlukannya.
412 <i>Precondition Failed</i>	Klien telah menunjukkan prasyarat dalam <i>headernya</i> yang tidak dipenuhi <i>server</i> .
413 <i>Payload Too Large</i>	Entitas permintaan lebih besar dari batas yang ditentukan oleh <i>server</i> . <i>Server</i> mungkin menutup koneksi atau mengembalikan bidang <i>header Retry-After</i> .
414 <i>URI Too Long</i>	URI yang diminta oleh klien lebih panjang dari yang dapat diinterpretasikan oleh <i>server</i> .
415 <i>Unsupported Media Type</i>	<i>Format</i> media dari data yang diminta tidak didukung oleh <i>server</i> , sehingga <i>server</i> menolak permintaan.

Kode	Deskripsi
416 Range Not Satisfiable	Rentang yang ditentukan oleh bidang <i>header</i> Rentang dalam permintaan tidak dapat dipenuhi. Mungkin saja rentangnya berada di luar ukuran data URI target.
417 Expectation Failed	Kode respons ini berarti ekspektasi yang ditunjukkan oleh bidang tajuk permintaan <i>Expect</i> tidak dapat dipenuhi oleh <i>server</i> .
418 I'm a teapot	<i>Server</i> menolak upaya untuk menyeduh kopi dengan teko.
421 Misdirected Request	Permintaan diarahkan ke <i>server</i> yang tidak dapat menghasilkan respons. Ini dapat dikirim oleh <i>server</i> yang tidak dikonfigurasi untuk menghasilkan respons untuk kombinasi skema dan otoritas yang disertakan dalam URI permintaan.
422 Unprocessable Entity	Permintaan itu terbentuk dengan baik tetapi tidak dapat diikuti karena kesalahan semantik.
423 Locked	Sumber daya yang sedang diakses terkunci.
424 Failed Dependency	Permintaan gagal karena kegagalan permintaan sebelumnya.
425 Too Early Experimental	Menunjukkan bahwa <i>server</i> tidak mau mengambil risiko memproses permintaan yang mungkin diputar ulang.
426 Upgrade Required	<i>Server</i> menolak untuk melakukan permintaan menggunakan protokol saat ini tetapi mungkin bersedia melakukannya setelah klien meningkatkan ke protokol yang berbeda. <i>Server</i> mengirimkan <i>header Upgrade</i> dalam respons 426 untuk menunjukkan protokol yang diperlukan.
428 Precondition Required	<i>Server</i> asal mengharuskan permintaan untuk bersyarat. Respons ini dimaksudkan untuk mencegah masalah ' <i>lost update</i> ', di mana klien MENDAPATKAN status sumber daya, memodifikasinya, dan PUT kembali ke <i>server</i> , ketika sementara itu pihak ketiga telah memodifikasi status di <i>server</i> .
429 Too Many Requests	Pengguna telah mengirim terlalu banyak permintaan dalam jumlah waktu tertentu (" <i>rate limiting</i> ").
431 Request Header Fields Too Large	<i>Server</i> tidak mau memproses permintaan karena bidang <i>headernya</i> terlalu besar. Permintaan dapat dikirim ulang setelah mengurangi ukuran bidang <i>header</i> permintaan.
451 Unavailable For Legal Reasons	Agen pengguna meminta sumber daya yang tidak dapat disediakan secara legal, seperti halaman web yang disensor oleh pemerintah.
500 Internal Server Error	<i>Server</i> mengalami situasi yang tidak diketahui bagaimana cara menanganinya.
501 Not Implemented	Metode permintaan tidak didukung oleh <i>server</i> dan tidak dapat ditangani. Satu-satunya metode yang harus didukung oleh <i>server</i> (dan oleh karena itu tidak boleh mengembalikan kode ini) adalah GET dan HEAD.
502 Bad Gateway	Respons kesalahan ini berarti bahwa <i>server</i> , saat bekerja sebagai <i>gateway</i> untuk mendapatkan respons yang diperlukan untuk menangani permintaan, mendapat respons yang tidak valid.
503 Service Unavailable	<i>Server</i> tidak siap untuk menangani permintaan. Penyebab umumnya adalah <i>server</i> yang sedang <i>down</i> untuk pemeliharaan atau yang kelebihan beban. Perhatikan bahwa bersama dengan

Kode	Deskripsi
	respons ini, halaman yang ramah pengguna yang menjelaskan masalah harus dikirim. Respons ini harus digunakan untuk kondisi sementara dan <i>header</i> Retry-After HTTP harus, jika memungkinkan, berisi perkiraan waktu sebelum pemulihan layanan. Webmaster juga harus berhati-hati tentang <i>header</i> terkait <i>caching</i> yang dikirim bersama dengan respons ini, karena respons kondisi sementara ini biasanya tidak boleh di-cache.
504 Gateway Timeout	Respons kesalahan ini diberikan ketika <i>server</i> bertindak sebagai gateway dan tidak bisa mendapatkan respons tepat waktu.
505 HTTP Version Not Supported	Versi HTTP yang digunakan dalam permintaan tidak didukung oleh <i>server</i> .
506 Variant Also Negotiates	<i>Server</i> memiliki kesalahan konfigurasi internal: sumber daya varian yang dipilih dikonfigurasi untuk terlibat dalam negosiasi konten transparan itu sendiri, dan oleh karena itu bukan titik akhir yang tepat dalam proses negosiasi.
507 Insufficient Storage	Metode ini tidak dapat dilakukan pada sumber daya karena <i>server</i> tidak dapat menyimpan representasi yang diperlukan untuk menyelesaikan permintaan dengan sukses.
508 Loop Detected	<i>Server</i> mendeteksi <i>loop</i> tak terbatas saat memproses permintaan.
510 Not Extended	Perluasan lebih lanjut ke permintaan diperlukan agar <i>server</i> dapat memenuhinya.
511 Network Authentication Required	Menunjukkan bahwa klien perlu mengotentikasi untuk mendapatkan akses jaringan.

3.14. UI

UI adalah bagian dari UX yang berupa tampilan visual design sebuah sistem. Tampilan tersebut memungkinkan pengguna terhubung dan berinteraksi dengan suatu produk (Muhyidin, Sulhan, and Sevtiana 2020).

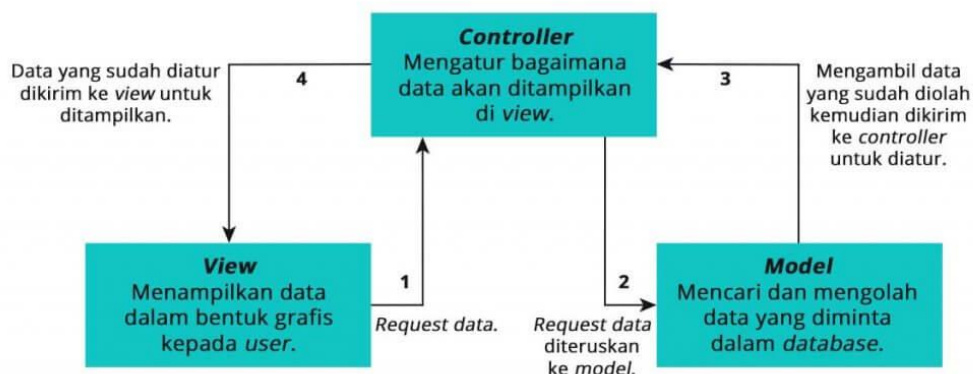
Selain berfungsi sebagai penghubung, UI juga berfungsi untuk memperindah tampilan sehingga dapat meningkatkan kepuasan pengguna. Namun, tak hanya harus indah, UI juga harus mudah digunakan.

Beberapa komponen UI diantaranya adalah komponen tombol, ikon tipografi, tema, *layout*, animasi yang tampil pada produk, dan visual interaktif lainnya. Semua komponen UI tersebut didesain dengan berfokus pada keindahan dan kemudahan pengguna.

3.15. MVC

MVC atau *Model View Controller* adalah sebuah pola desain arsitektur dalam sistem pengembangan *website* yang terdiri dari tiga bagian(Saputra and Syazili n.d.). Bagian tersebut adalah:

1. *Model*, bagian yang mengelola dan berhubungan langsung dengan *database*;
2. *View*, bagian yang akan menyajikan tampilan Informasi kepada pengguna;
3. *Controller* , bagian yang menghubungkan *Model* dan *View* dalam setiap proses *request* dari *User*.



Gambar 3. 1. Arsitektur MVC

Dengan konsep MVC ini, *website* seakan memiliki bagian yang terpisah dan bisa dikembangkan masing-masing. Maka, proses pembuatan *website* bisa dilakukan lebih cepat karena *developer* akan lebih fokus pada pengerjaan salah satu bagian saja.

Karena dianggap efektif, konsep MVC banyak diterapkan di berbagai *framework*. Sebagai contoh, di *framework* PHP terbaik seperti Laravel, CodeIgniter, Symfony, Yii, dan Zend sudah menggunakan konsep ini.

BAB IV

ANALISIS DAN PERANCANGAN

Pada bab ini dilakukan analisis dan perancangan *website* sistem Informasi Tugas Akhir dan Kerja Praktik.

4.1. Deskripsi Umum Sistem

Sistem Informasi adalah seperangkat komponen yang saling berhubungan yang berfungsi untuk mengumpulkan, memproses, menyimpan, dan mendistribusikan Informasi untuk mendukung pembuatan keputusan dan pengawasan dalam organisasi.

Sistem Kerja Praktik dan Tugas Akhir Program Studi Sistem Informasi (SITASI) memiliki empat *User* yaitu mahasiswa, dosen, koordinator, dan tata usaha. Mahasiswa dapat melakukan pendaftaran tugas akhir dan mengajukan pendaftaran sidang proposal tugas akhir, seminar tugas akhir, sidang tugas akhir dan yudisium. Koordinator dapat menambahkan *User* dan menentukan *role User* yang ditambahkan dan dapat mengelola data tugas akhir dan menentukan pembimbing dan penguji mahasiswa tugas akhir, sidang proposal tugas akhir, seminar proposal tugas akhir, sidang tugas akhir, dan yudisium.

4.2. Analisis Kebutuhan Fungsional

System yang dibangun, diharapkan *User* dapat menggunakan sistem Informasi ini dalam melakukan kegiatan praktik kerja dan tugas akhir. Pada aturan penomoran ini adalah M-FR-xx, KTA-FR-xx, KKP-FR-xx, D-FR-xx, dan T-FR-xx dimana M adalah inisial mahasiswa, KTA adalah inisial koordinator tugas akhir, KKP adalah inisial koordinator kerja praktik, D adalah inisial dosen, dan T adalah inisial tata usaha sedangkan FR singkatan dari *functional requirement* serta xx adalah nomor urut. Berikut ini kebutuhan fungsional pada *system* ini:

1. Kebutuhan fungsional mahasiswa

Tabel 4. 1. Kebutuhan fungsional *user* mahasiswa

Kode	Deskripsi
M-FR-01	Mahasiswa dapat mengajukan kerja praktik (Form-001)
M-FR-02	Mahasiswa dapat mengajukan surat pengantar untuk perusahaan
M-FR-03	Mahasiswa dapat melakukan pendaftaran kerja praktik
M-FR-04	Mahasiswa dapat mengajukan sidang kerja praktik

2. Kebutuhan fungsional koordinator tugas akhir

Tabel 4. 2. Kebutuhan fungsional *user* koordinator tugas akhir

Kode	Deskripsi
KTA-FR-01	Koordinator TA dapat mengelola data <i>User</i> mahasiswa, dosen, dan koordinator TA
KTA -FR-02	Koordinator dapat mengelola data tugas akhir
KTA -FR-03	Koordinator dapat mengelola data sidang proposal tugas akhir
KTA -FR-04	Koordinator dapat mengelola data seminar tugas akhir
KTA -FR-05	Koordinator dapat mengelola data sidang tugas akhir
KTA -FR-06	Koordinator dapat mengelola data yudisium

Tabel 4. 3. Kebutuhan fungsional *user* koordinator kerja praktik

Kode	Deskripsi
KKP-FR-01	Koordinator KP dapat mengelola data <i>User</i> mahasiswa, dosen, dan koordinator KP
KKP-FR-02	Koordinator dapat mengelola data kerja praktik
KKP-FR-03	Koordinator dapat mengelola data sidang kerja praktik

3. Kebutuhan fungsional dosen

Tabel 4. 3. Kebutuhan fungsional *user* dosen

Kode	Deskripsi
D-FR-01	Dosen dapat mengelola data sidang proposal mahasiswa yang dibimbing
D-FR-02	Dosen dapat mengelola data seminar proposal mahasiswa yang dibimbing
D-FR-03	Dosen dapat mengelola data sidang tugas akhir mahasiswa yang dibimbing
D-FR-04	Dosen dapat mengelola data sidang proposal mahasiswa yang diuji
D-FR-05	Dosen dapat mengelola data seminar proposal mahasiswa yang diuji
D-FR-06	Dosen dapat mengelola data sidang tugas akhir mahasiswa yang diuji
D-FR-07	Dosen dapat mengelola data sidang kerja praktik yang dibimbing
D-FR-08	Dosen dapat mengelola data sidang kerja praktik yang diuji

4. Kebutuhan fungsional tata usaha

Tabel 4. 4. Kebutuhan fungsional user tata usaha

Kode	Deskripsi
T-FR-01	Tata usaha dapat mengelola data <i>User</i> mahasiswa
T-FR-02	Tata usaha dapat melihat data proposal tugas akhir
T-FR-03	Tata usaha dapat melihat data seminar tugas akhir
T-FR-04	Tata usaha dapat melihat data sidang tugas akhir
T-FR-05	Tata usaha dapat melihat data yudisium
T-FR-06	Tata usaha dapat melihat data Form-001
T-FR-07	Tata usaha dapat mengelola surat pengantar praktik kerja
T-FR-08	Tata usaha dapat melihat data sidang praktik kerja

4.3. Analisa Kebutuhan Nonfungsional

Kebutuhan Non-fungsional mengacu pada batasan layanan atau fungsionalitas yang disediakan oleh sistem. Pada aturan penomoran ini adalah NFR-xx, dimana NFR singkatan dari *non functional requirement* serta xx adalah nomor urut. Berikut ini kebutuhan non fungsional pada *system* ini:

Tabel 4. 5. Kebutuhan nonfungsional sistem informasi

Kode	Deskripsi
NFR-01	Sistem memiliki tampilan yang sederhana yang dapat mudah dipahami.
NFR-02	Membuat web yang <i>responsive</i> untuk meningkatkan kenyamanan pengguna.
NFR-03	Sistem diharapkan bisa mempercepat pengaksesan data dan informasi

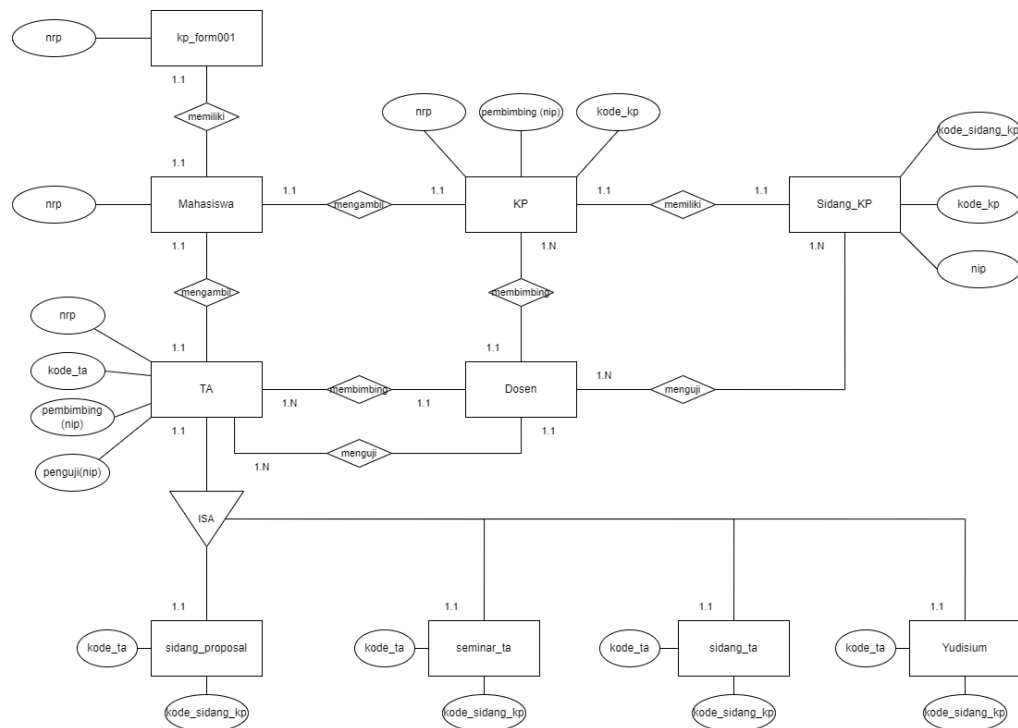
4.4. Perancangan Basis Data

Proses perancangan *database* merupakan bagian dari *micro lifecycle*. Sedangkan kegiatan-kegiatan yang terdapat di dalam proses tersebut diantaranya : pengumpulan data dan analisis, perancangan *database* secara konsensual, pemilihan DBMS, perancangan *database* secara logika (*data Model mapping*), perancangan *database* secara fisik, dan implementasi sistem *database*.

4.4.1. Diagram Chen

Rancangan *database* berupa diagram Chen yang diperlukan untuk melihat kebutuhan sistem dan membantu pembuatan *database* dalam penentuan relasi antar entitas yang akan dibuat. Diagram ini dibuat berdasarkan kebutuhan yang ada pada

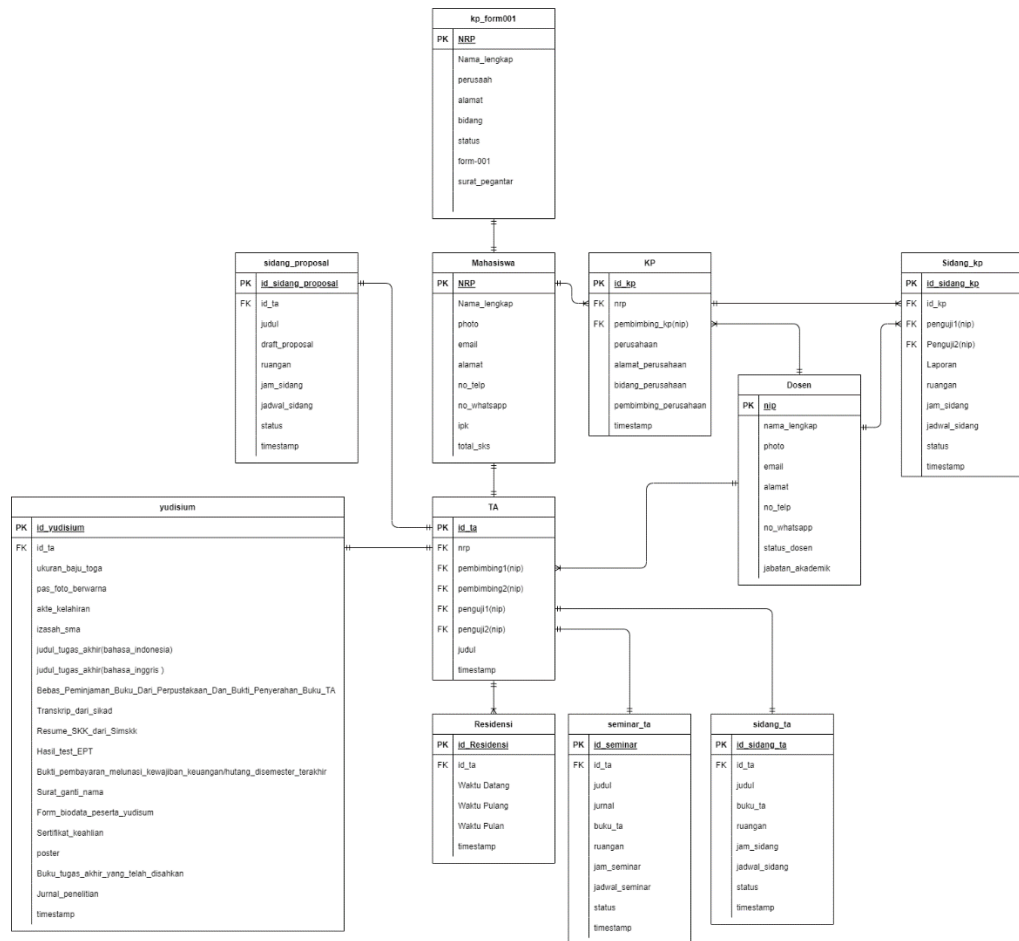
program studi Sistem Informasi Institut Teknologi Nasional Bandung. Adapun gambaran diagram Chen yang dibuat penulis dari sistem Informasi mahasiswa praktek kerja dan mahasiswa tugas akhir program studi Sistem Informasi Institut Teknologi Nasional Bandung dapat dilihat pada Gambar 4. 1.



Gambar 4. 1. Diagram chen perancangan *website*

4.4.2. Diagram Crowfoot

Rancangan *database* sistem Informasi praktek kerja dan tugas akhir berupa diagram crowfoot yang diperlukan untuk membantu pembuatan *database* dalam menentukan relasi setiap entitas yang diperlukan. Diagram crowfoot ini berbeda dengan diagram chen yang paling terlihat adalah penambahan atribut yang dibutuhkan setiap entitasnya. Diagram *crowfoot* bisa dilihat pada Gambar 4. 2.

Gambar 4. 2. Diagram crowfoot perancangan *website*

Kamus data

Tabel 4. 6. Kamus data tabel mahasiswa

Nama Tabel: Mahasiswa					
Deskripsi: tabel yang berisikan data setiap mahasiswa					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
NRP	Varchar	150	Not_null	Primary key	NRP mahasiswa
Nama_lengkap	Varchar	150	Null		Nama lengkap mahasiswa
Foto	Varchar	150	Null		Foto mahasiswa
Email	Varchar	150	Null		Alamat email mahasiswa
Alamat	Varchar	150	Null		Alamat tempat tinggal mahasiswa
No_telp	Varchar	150	Null		No telp mahasiwa

Nama Tabel: Mahasiswa					
Deskripsi: tabel yang berisikan data setiap mahasiswa					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
No_whatsapp	Varchar	150	Null		No whatsapp mahasiswa
IPK	Double	50	Null		Jumlah IPK mahasiswa
Total_sks	Integer	50	Null		Jumlah SKS mahasiswa

Tabel 4. 7. Kamus data tabel dosen

Nama Tabel: Dosen					
Deskripsi: tabel yang berisikan data setiap dosen					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
NIP	Varchar	150	Not_null	Primary key	NIP dosen
Nama_lengkap	Varchar	150	Null		Nama lengkap dosen
Foto	Varchar	150	Null		Foto dosen
Email	Varchar	150	Null		Alamat email dosen
Alamat	Varchar	150	Null		Alamat tempat tinggal dosen
No_telp	Varchar	150	Null		No telp Dosen
No_whatsapp	Varchar	150	Null		No whatsapp dosen
Status_dosen	Varchar	150	Null		Jumlah IPK dosen
Jabatan_akademik	Varchar	150	Null		Jumlah SKS dosen

Tabel 4. 8. Kamus data tabel TA

Nama Tabel: TA					
Deskripsi: Tabel yang berisikan data tugas akhir					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_TA	Varchar	150	Not_null	Primary key	Kode tugas akhir
NIP_Pembimbing1	Varchar	150	Null	Foreign key	NIP dosen Pembimbing 1
NIP_Pembimbing2	Varchar	150	Null	Foreign key	NIP dosen pembimbing 2
NIP_Penguji1	Varchar	150	Null	Foreign key	NIP penguji 1

Nama Tabel: TA					
Deskripsi: Tabel yang berisikan data tugas akhir					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
NIP_Penguji2	Varchar	150	Null	Foreign key	NIP penguji 2
Judul	Varchar	150	Null		Judul mahasiswa
Draft_proposal	Varchar	150	Null		Draft proposal tugas akhir
Timestamp	Time		Null		Waktu daftar tugas akhir

Tabel 4. 9. Kamus data tabel KP

Nama Tabel: KP					
Deskripsi: tabel yang berisikan data kerja praktik					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_KP	Varchar	150	Not_null	Primary key	Kode kerja praktik
NRP	Varchar	150	Not_null	Foreign key	NRP mahasiswa
NIP_Pembimbing_KP	Varchar	150	Null	Foreign key	NIP dosen Pembimbing 1
Perusahaan	Varchar	150	Null		Nama perusahaan
Alamat_perusahaan	Varchar	150	Null		Alamat perusahaan
Bidang_perusahaan	Varchar	150	Null		Bidang perusahaan
Pembimbing_perusahaan	Varchar	150	Null		Nama pembimbing perusahaan
Timestamp	Varchar	150	Null		Waktu daftar KP

Tabel 4. 10. Kamus data tabel sidang KP

Nama Tabel: sidang_KP					
Deskripsi: tabel yang berisikan data sidang kerja praktik					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_Sidang_KP	Varchar	150	Not_null	Primary key	Kode sidang kerja praktik
Id_KP	Varchar	150	Not_null	Foreign key	Kode kerja praktik
Penguji1	Varchar	150	Null	Foreign key	NIP dosen penguji 1

Nama Tabel: sidang_KP					
Deskripsi: tabel yang berisikan data sidang kerja praktik					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
Penguji2	Varchar	150	Null	Foreign key	NIP dosen penguji 2
Laporan	Varchar	150	Null		Laporan kerja praktik
Ruangan	Varchar	150	Null		No ruangan
Jam_sidang	Varchar	150	Null		Waktu sidang
Jadwal_sidang	Varchar	150	Null		Tanggal sidang
Status	Varchar	150	Null		Status sidang KP
Timestamp	Time		Null		Waktu upload sidang KP

Tabel 4. 11. Kamus data tabel sidang proposal

Nama Tabel: sidang_proposal					
Deskripsi: tabel yang berisikan data sidang proposal					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_Sidang_Proposal	Varchar	150	Not_null	Primary key	Kode sidang proposal
Id_TA	Varchar	150	Not_null	Foreign key	Kode tugas akhir
Judul	Varchar	150	Null		Judul tugas akhir
Proposal	Varchar	150	Null		Proposal tugas akhir
Ruangan	Varchar	150	Null		No ruangan
Jam_sidang	Varchar	150	Null		Waktu sidang
Jadwal_sidang	Varchar	150	Null		Tanggal sidang
Status	Varchar	150	Null		Status sidang KP
Timestamp	Time		Null		Waktu upload sidang KP

Tabel 4. 12. kamus data tabel seminar TA

Nama Tabel: seminar_ta					
Deskripsi: tabel yang berisikan data seminar tugas akhir					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_seminar	Varchar	150	Not_null	Primary key	Kode seminar
Id_TA	Varchar	150	Not_null	Foreign key	Kode tugas akhir

Nama Tabel: seminar_ta					
Deskripsi: tabel yang berisikan data seminar tugas akhir					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
Judul	Varchar	150	Null		Judul tugas akhir
Jurnal	Varchar	150	Null		Jurnal tugas akhir
Draft_buku_TA	Varchar	150	Null		Draft buku tugas akhir
Ruangan	Varchar	150	Null		No ruangan
Jam_seminar	Varchar	150	Null		Waktu seminar
Jadwal_seminar	Varchar	150	Null		Tanggal seminar
Status	Varchar	150	Null		Status seminar KP
Timestamp	Time		Null		Waktu upload seminar KP

Tabel 4. 13. Kamus data tabel sidang TA

Nama Tabel: sidang_ta					
Deskripsi: tabel yang berisikan data sidang tugas akhir					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_Sidang_ta	Varchar	150	Not_null	Primary key	Kode sidang tugas akhir
Id_TA	Varchar	150	Not_null	Foreign key	Kode tugas akhir
Judul	Varchar	150	Null		Judul tugas akhir
Buku_ta	Varchar	150	Null		Buku tugas akhir
Ruangan	Varchar	150	Null		Nomor ruangan
Jam_sidang	Varchar	150	Null		Waktu sidang
Jadwal_sidang	Varchar	150	Null		Tanggal sidang
Status	Varchar	150	Null		Status sidang KP
Timestamp	Time		Null		Waktu upload sidang KP

Tabel 4. 14. Kamus data tabel yudisium

Nama Tabel: yudisium					
Deskripsi: tabel yang berisikan data yudisium					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
ID_ yudisium	Varchar	150	Not_null	Primary key	Kode yudisium
Id_TA	Varchar	150	Not_null	Foreign key	Kode tugas akhir
Ukuran_baju_toga	Varchar	150	Null		Nomor ukuran baju
Pas_foto_berwarna	Varchar	150	Null		Pas foto berwarna
Akte_kelahiran	Varchar	150	Null		Akte kelahiran
Izasah_sma	Varchar	150	Null		Izasah SMA
Judul_ta_indonesia	Varchar	150	Null		Judul tugas akhir
Judul_ta_inggris	Varchar	150	Null		Judul tugas akhir
Bebas_Peminjaman_Buku_Dari_Perpustakaan_Dan_Bukti_Penyerahan_Buku_TA	Varchar	150	Null		Kartu bebas peminjaman
Transkrip_dari_sikad	Varchar	150	Null		Transkrip nilai sari sikad
Resume_SKK_dari_Simskk	Varchar	150	Null		Resume SKK dari SIMSKK
Hasil_test_EPT	Varchar	150	Null		Hasil test EPT
Bukti_pembayaran_melunasi_kewajiban_keuangan/hutang_disemester_terakhir	Varchar	150	Null		Bukti pembayaran kuliah
Surat_ganti_nama	Varchar	150	Null		Surat ganti nama
Form_biodata_peserta_yudisum	Varchar	150	Null		Form biodata peserta yudisium
Sertifikat_keahlian	Varchar	150	Null		Sertifikat keahlian
Poster	Varchar	150	Null		Poster
Buku_tugas_akhir_yang_telah_disahkan	Varchar	150	Null		Buku tugas akhir

Nama Tabel: yudisium					
Deskripsi: tabel yang berisikan data yudisium					
Nama Atribut	Tipe Data	Panjang Data	Null	Key	Deskripsi
Jurnal_penelitian	Varchar	150	Null		Jurnal penelitian
timestamp	Time				Waktu upload yudisium

4.5. Perancangan Sistem

Perancangan sistem adalah proses perancangan untuk merancang sistem atau memperbaiki sistem yang telah ada sehingga sistem menjadi lebih baik serta dapat mengerjakan pekerjaan secara efektif dan efisien

4.5.1. Diagram *Use Case*

Diagram *Use Case* menggambarkan interaksi antara satu atau lebih aktor yang terhubung dengan sistem yang dibuat. Dalam hal ini, *Use Case* menggambarkan aktivitas setiap aktor dalam sistem yang dibuat dan penggunaan fitur dari setiap sistem. Aktivitas aktor diwakili oleh diagram lingkaran yang berisi deskripsi aktivitas yang dilakukan oleh sistem.

diberikan beberapa fungsi saja untuk menunjang pembelajaran. Pada *Use Case* yang berwarna biru merupakan *Use Case* yang akan dibuat oleh penulis. Berikut ini merupakan *child* dari setiap *Use Case* :

Tabel 4. 15. *List use case* yang dirancang dan penjelasan

ID	Child	Keterangan	Aktor
UC-02	UC-02.01	Mengisi Form Pengajuan KP (Form-001)	Mahasiswa
UC-02	UC-02.02	Melihat dan Menggugah Surat Pengantar Kerja Praktik	Mahasiswa
UC-03		Mengelola Data Mahasiswa Kerja Praktik	Koordinator KP
UC-03	UC-03.01	Menentukan Pembimbing dan Penguji Kerja Praktik	Koordinator KP
UC-04		Mengelola Data Mahasiswa Tugas Akhir	Koordinator TA
UC-04	UC-04.01	Menentukan Pembimbing dan Penguji Tugas Akhir	Koordinator TA
UC-12		Melihat Data Form-001 Kerja Praktik	Tata Usaha
UC-12	UC-12.01	Mengunggah Surat Pengantar Kerja Praktik	Tata Usaha
UC-13		Melihat Data Sidang Kerja Praktik	Tata Usaha

4.5.2. *Use Case* Skenario

Use Case skenario merupakan sebuah dokumentasi terhadap kebutuhan fungsional dari sebuah sistem. Pada tabel *Use Case* skenario berisikan nomor *Use Case*, nama *Use Case*, tujuan, deskripsi, kondisi awal, aktor, skenario normal, dan kondisi akhir skenario. Berikut ini adalah skenario *Use Case* yang ada.

1. *Use Case* Skenario Mendaftar Peserta Kerja Praktik

Pada bagian ini menjelaskan skenario bagaimana mahasiswa mendaftar sebagai peserta kerja praktik

Tabel 4. 16. *Use Case* skenario mendaftar peserta tugas akhir

Identifikasi	UC-02
Nama <i>Use Case</i>	Mendaftar Peserta Kerja Praktik
Tujuan	Mahasiswa mengajukan kerja praktik, mendaftar menjadi peserta kerja praktik, dan sidang kerja praktik

Aktor	Mahasiswa
Deskripsi	Mahasiswa mengajukan kerja praktik, mendaftar menjadi peserta kerja praktik, dan sidang kerja praktik
Kondisi awal	Mahasiswa login lalu menuju halaman daftar kerja praktik
Skenario normal: sistem menampilkan <i>form</i> praktik kerja	
Aktor	Sistem/Software
	1. Menampilkan halaman landing page
2. Melakukan login	3. Menampilkan <i>dashboard</i> mahasiswa
4. Menuju halaman pengajuan kerja praktik	5. Menampilkan halaman pengajuan kerja praktik
6. Melakukan pengajuan kerja praktik (Form-001)	7. Menyimpan data hasil pengajuan
	8. Menampilkan <i>data</i> hasil pendaftaran
9. Menuju halaman pendaftaran kerja praktik	10. Menampilkan halaman kerja praktik
11. Menekan tombol pendaftaran kerja praktik	12. Menampilkan <i>form</i> pendaftaran kerja praktik
13. Melakukan pendaftaran	14. Menyimpan data hasil pendaftaran
15.	16. Menampilkan <i>data</i> hasil pendaftaran
17. Menuju halaman pendaftaran sidang kerja praktik	18. Menampilkan halaman sidang kerja praktik
19. Menekan tombol pendaftaran sidang kerja praktik	20. Menampilkan <i>form</i> pendaftaran sidang kerja praktik
21. Melakukan pendaftaran sidang kerja praktik	22. Menyimpan data hasil pendaftaran
23.	24. Menampilkan <i>data</i> hasil pendaftaran
Kondisi akhir: sistem dapat menyimpan data hasil pengajuan, pendaftaran, dan sidang kerja praktik.	

2. Use Case Skenario Mengelola Data Kerja Praktik

Pada bagian ini menjelaskan skenario bagaimana mahasiswa mendaftar sebagai peserta kerja praktik.

Identifikasi	UC-03
Nama Use Case	Mengelola Data Mahasiswa Kerja Praktik
Tujuan	Menentukan pembimbing dan penguji mahasiswa kerja praktik
Aktor	Koordinator

Deskripsi	Koordinator mengelola data peserta kerja praktik
Kondisi awal	Koordinator login lalu menuju halaman daftar kerja praktik
Skenario normal: sistem menampilkan <i>form</i> daftar praktik kerja	
Aktor	Sistem/Software
1. Menuju ke halaman kerja praktik	2. Menampilkan data mahasiswa yang mendaftar kerja praktik
3. Menekan tombol edit	4. Menampilkan <i>form</i> mahasiswa kerja praktik
5. Menentukan penguji dan pembimbing kerja praktik	6. Menampilkan daftar dosen sistem informasi
7. Menekan tombol save	8. Menyimpan data hasil edit koordinator
	9. Menampilkan data hasil edit koordinator
Kondisi akhir: sistem dapat menyimpan data hasil update dari koordinator	

3. Use Case Skenario Mengelola Data Tugas Akhir

Pada bagian ini menjelaskan skenario bagaimana koordinator mengelola data peserta kerja praktik.

Identifikasi	UC-04
Nama Use Case	Mengelola Data Mahasiswa Tugas Akhir
Tujuan	Menentukan pembimbing dan penguji mahasiswa tugas akhir
Aktor	Koordinator
Deskripsi	Koordinator mengelola data peserta tugas akhir
Kondisi awal	Koordinator login lalu menuju halaman daftar tugas akhir
Skenario normal: sistem menampilkan <i>form</i> daftar tugas akhir	
Aktor	Sistem/Software
1. Menuju ke halaman tugas akhir	2. Menampilkan data mahasiswa yang mendaftar tugas akhir
3. Menekan tombol edit	4. Menampilkan <i>form</i> mahasiswa tugas akhir
5. Menentukan penguji dan pembimbing tugas akhir	6. Menampilkan daftar dosen sistem informasi
7. Menekan tombol save	8. Menyimpan data hasil edit koordinator
	9. Menampilkan data hasil edit koordinator
Kondisi akhir: sistem dapat menyimpan data hasil update dari koordinator	

4. Use Case Skenario Mengelola Data Form-001 Kerja Praktik

Pada bagian ini menjelaskan skenario bagaimana tata usaha mengelola data form-001 kerja praktik.

Identifikasi	UC-13
Nama Use Case	Mengelola Data Form-001 Kerja Praktik
Tujuan	Tata usaha dapat mengelola data mahasiswa form-001 kerja praktik
Aktor	Tata usaha
Deskripsi	Tata usaha dapat mengelola data mahasiswa form-001 kerja praktik
Kondisi awal	Tata usaha login lalu menuju halaman sidang kerja praktik
Skenario normal: sistem menampilkan data mahasiswa sidang kerja praktik	
Aktor	Sistem/Software
1. Menuju halaman sidang kerja praktik	2. Menampilkan data mahasiswa yang mendaftar sidang kerja praktik
3. Menekan tombol lihat	4. Menampilkan laporan sidang kerja praktik
Kondisi akhir: tata usaha dapat melihat data mahasiswa sidang kerja praktik	

5. Use Case Skenario Melihat Data Sidang Kerja Praktik

Pada bagian ini menjelaskan skenario bagaimana dosen membimbing peserta tugas akhir.

Identifikasi	UC-13
Nama Use Case	Melihat Data Sidang Kerja Praktik
Tujuan	Tata usaha dapat melihat data mahasiswa sidang kerja praktik
Aktor	Tata usaha
Deskripsi	Tata usaha melihat data sidang kerja praktik
Kondisi awal	Tata usaha login lalu menuju halaman sidang kerja praktik
Skenario normal: sistem menampilkan data mahasiswa sidang kerja praktik	
Aktor	Sistem/Software

5. Menuju halaman sidang kerja praktik	6. Menampilkan data mahasiswa yang mendaftar sidang kerja praktik
7. Menekan tombol lihat	8. Menampilkan laporan sidang kerja praktik
Kondisi akhir: tata usaha dapat melihat data mahasiswa sidang kerja praktik	

4.5.3. Activity Diagram

Dari hasil analisis sistem yang dibutuhkan setelah membuat *Use Case* diagram selanjutnya adalah membuat activity diagram untuk menggambarkan urutan aktifitas proses pada sistem. Selain itu activity diagram membantu untuk memahami proses secara keseluruhan. Berikut ini adalah activity diagram dari *Use Case* scenario diagram yang telah dibuat.

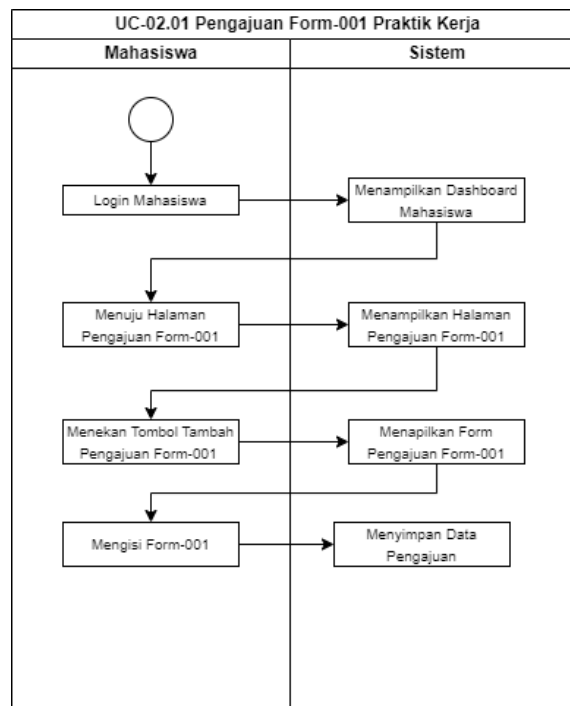
Activity Diagram menjelaskan urutan aktivitas dalam suatu proses. Di dalam dunia bisnis biasanya digunakan untuk *Modeling* (memperlihatkan urutan proses bisnis). Selain itu Activity Diagram mudah dalam memahami proses yang ada dalam sistem secara keseluruhan. Selain itu Activity Diagram merupakan metode perancangan yang terstruktur. Berikut adalah list Activity Diagram yang dibuat:

Tabel 4. 17. List activity diagram yang dibuat

No	Nama Activity Diagram	Use Case
1	Activity Diagram Mendaftarkan Peserta Kerja Praktik	UC-02
2	Activity Diagram Mengelola Data Mahasiswa Kerja Praktik	UC-03
3	Activity Diagram Mengelola Data Mahasiswa Tugas Akhir	UC-04
4	Activity Diagram Melihat Data Mahasiswa Kerja Praktik	UC-12
5	Activity Diagram Mengelola Data Form-001 Kerja Praktik	UC-13
8	Activity Diagram Mengunggah Surat Pengantar KP	UC-12.01
9	Activity Diagram Melihat Data Sidang KP	UC-13

4.5.3.1. Activity Diagram Mengisi Form Pengajuan KP (Form-001)

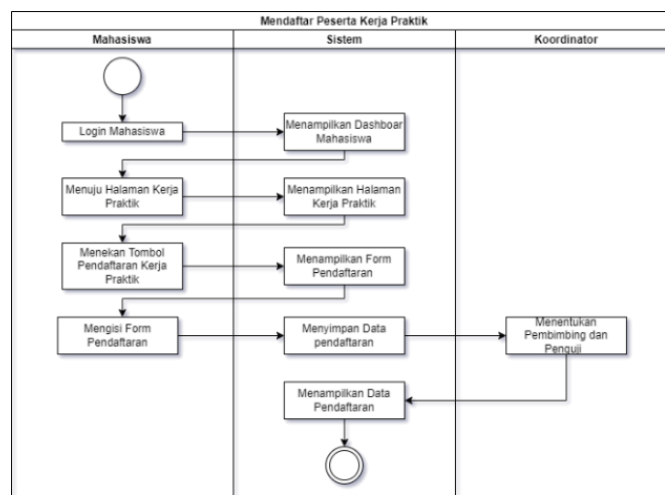
Pada activity diagram ini terjadi proses pendaftaran peserta tugas akhir yang dimana *pool* yang berperan pada proses ini adalah mahasiswa, koordinator sistem *website*.



Gambar 4. 4. Diagram aktifitas mendaftar peserta tugas akhir

4.5.3.2. Activity Diagram Melihat dan Mengunggah Surat Pengantar KP

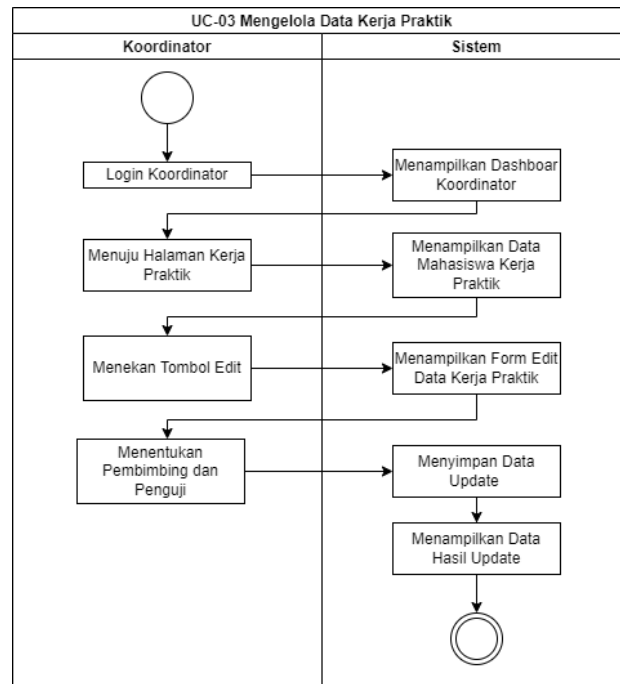
Pada activity diagram ini terjadi proses pendaftaran peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah mahasiswa, koordinator dan sistem *website*.



Gambar 4. 5. Diagram aktifitas mendaftar peserta kerja praktik

4.5.3.3. Activity Diagram Mengelola Data Mahasiswa KP

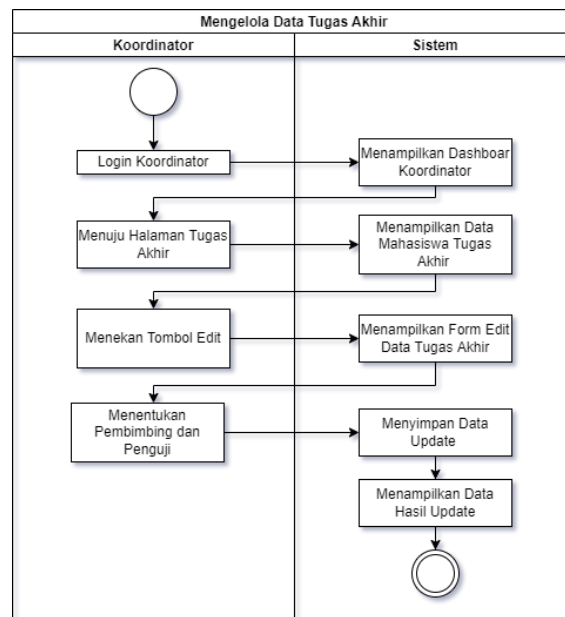
Pada activity diagram ini terjadi proses pengelolaan data peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah koordinator dan sistem *website*.



Gambar 4. 6. diagram aktifitas mengelola data kerja praktik

4.5.3.4. Activity Diagram Menentukan Pembimbing dan Penguji KP

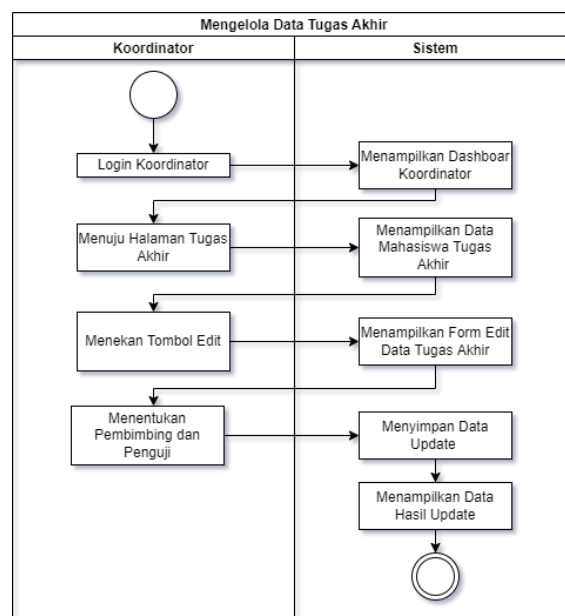
Pada activity diagram ini terjadi proses pengelolaan data peserta tugas akhir yang dimana *pool* yang berperan pada proses ini adalah koordinator dan sistem *website*.



Gambar 4. 7. Diagram aktifitas mengelola data tugas akhir

4.5.3.5. Activity Diagram Mengelola Data Mahasiswa TA

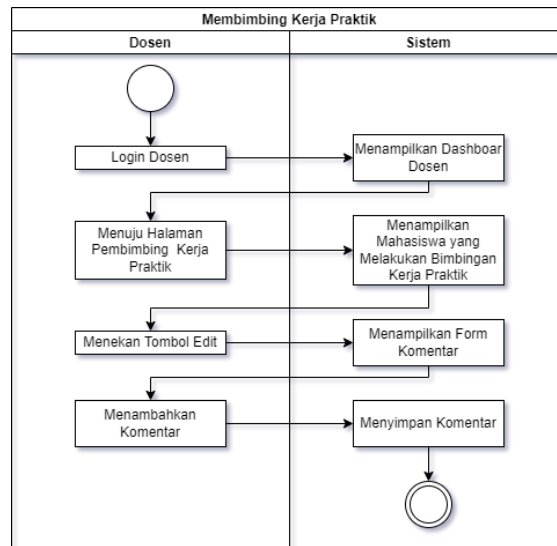
Pada activity diagram ini terjadi proses membimbing peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah dosen dan sistem *website*.



Gambar 4. 8. Diagram aktifitas membimbing tugas akhir

4.5.3.6. Activity Diagram Menentukan Pembimbing dan Penguji TA

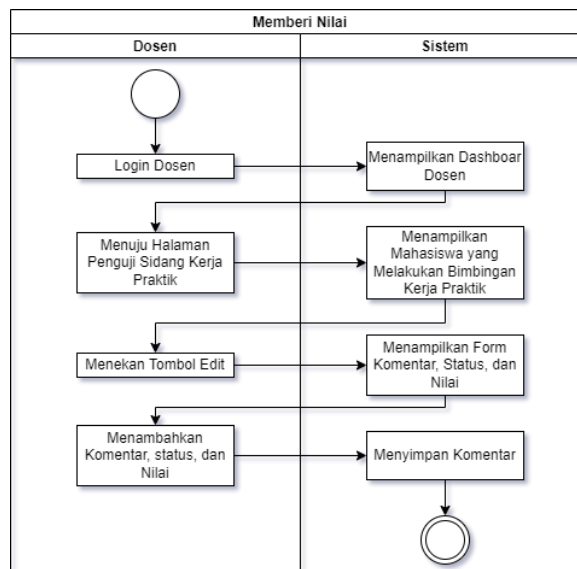
Pada activity diagram ini terjadi proses membimbing peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah koordinator dan sistem *website*.



Gambar 4. 9. Diagram aktifitas membimbing kerja praktik

4.5.3.7. Activity Diagram Melihat Data Form-001 KP

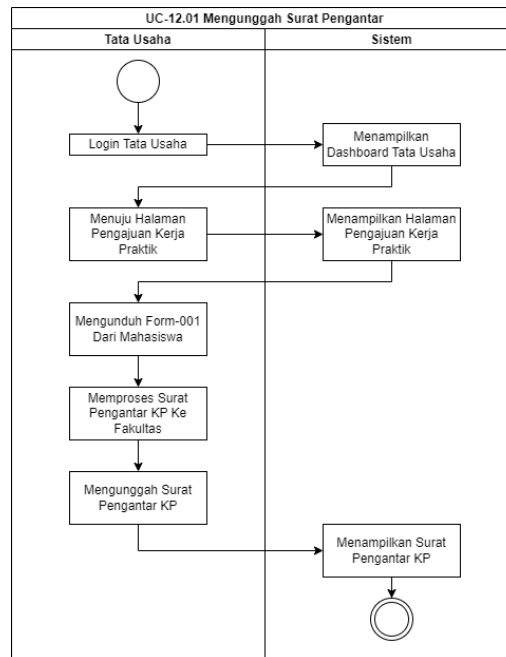
Pada activity diagram ini terjadi proses pemberian nilai kepada peserta sidang kerja praktik yang dimana *pool* yang berperan pada proses ini adalah dosen dan sistem *website*.



Gambar 4. 10. Diagram aktifitas memberi nilai

4.5.3.8. Activity Diagram Mengunggah Surat Pengantar KP

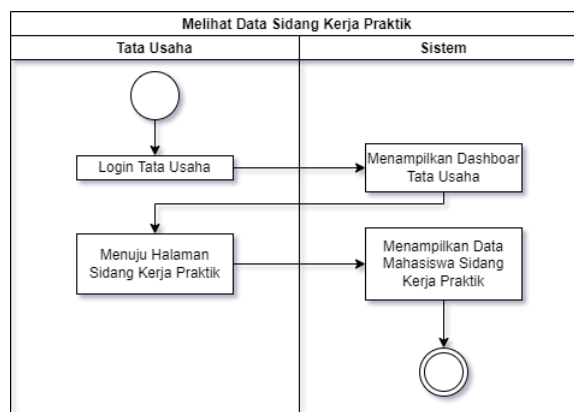
Pada activity diagram ini terjadi proses melihat data sidang proposal peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah tata usaha dan sistem *website*.



Gambar 4. 11. Diagram aktifitas melihat data sidang proposal tugas akhir

4.5.3.9. Activity Diagram Melihat Data Sidang KP

Pada activity diagram ini terjadi proses melihat data sidang kerja praktik peserta kerja praktik yang dimana *pool* yang berperan pada proses ini adalah tata usaha dan sistem *website*



Gambar 4. 12. Diagram aktifitas melihat data sidang kerja praktik

4.5.4. CRUD Matrik

Berikut merupakan CRUD matrik dibuat pada modul sistem Informasi tugas akhir dan kerja praktik Institut Teknologi Nasional.

Tabel 4. 18. CRUD matrik yang dibuat oleh penulis

No	Modul Pengerjaan	Role				PIC
		Mahasiswa	Dosen	Koordinator	TU	
Modul Tambahan						
1	Login/Logout Tugas Akhir	CRUD	CRUD	CRUD	CRUD	Daffa
2	Website CRUD Kerja Praktek	CRUD	RU	RUD	RU	Daffa
3	Fungsional Website (Di luar Modul)	-	-	-	-	Daffa
4	Mengelola Semua Komponen Kebutuhan Website	-	-	-	-	Daffa
Modul 2: Coding Pendaftaran TA (Role: Mahasiswa)						
1	Input Data: Daftar Sidang Proposal Tugas Akhir	CRU	RU	CRUD	R	Daffa
2	Input Data: Seminar Tugas Akhir	CRU	RU	CRUD	R	Daffa
3	Input Data: Sidang Kerja Praktik	CRU	RU	CRUD	R	Daffa
Modul 3: Coding Data Management Pendaftaran Sidang (Role : Koor TA)						
1	Read Update Data: Proposal Tugas Akhir	RU	RU	RUD	R	Daffa
2	Read Update Data: Sidang Kerja Praktik	RU	RU	RUD	R	Daffa
Modul 4 Coding Daftar Mahasiswa Bimbingan (Role: Dosen)						
1	Read Update Data: Proposal Tugas Akhir	RU	RUD	RU	R	Daffa
2	Read Update Data: Sidang Kerja Praktik	RU	RUD	RU	R	Daffa
Modul 5: Coding Daftar Mhs Yang Diuji (Role: Dosen)						

No	Modul Pengerjaan	Role				PIC
		Mahasiswa	Dosen	Koordinator	TU	
1	Read Update Data: Proposal Tugas Akhir	RU	RUD	R	R	Daffa
2	Read Update Data: Sidang Kerja Praktik	RU	RUD	R	R	Daffa
Modul 6: Coding Data Management Pendaftaran Sidang (Role: TU)						
1	Lihat Data: Daftar Proposal Mahasiswa	RU	R	RU	RUD	Daffa
2	Lihat Data : Daftar Sidang Kerja praktik	RU	R	RU	RUD	Daffa

Keterangan;

1. C: *User* dapat menambahkan data
2. R: *User* dapat melihat data
3. U: *User* dapat mengubah data
4. D: *User* dapat menghapus data

4.5.5. Desain Application Programming Interface

Serangkaian protokol, rutinitas, dan alat untuk membangun aplikasi dikenal sebagai Antarmuka Pemrograman Aplikasi (API). Berdasarkan metode permintaan permintaan HTTP, sumber daya dalam penelitian ini dikategorikan dalam suatu pola. Berikut ini merupakan daftar API yang digunakan pada sistem ini.

Tabel 4. 19. List API yang dibuat pada sistem *website*

Nama	Method	URL	Deskripsi
Dashboard Mahasiswa	GET	http://127.0.0.1:8000/dashboard-mahasiswa-form-001	
Tambah Data Form-001 (Mahasiswa)	GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-form-001	
Simpan Data Form-001 (Mahasiswa)	POST	http://127.0.0.1:8000/dashboard-mahasiswa-form-001	
Edit Data Form-001 (Mahasiswa)	GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	
Update Data Form-001 (Mahasiswa)	PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	

Nama	Method	URL	Deskripsi
Hapus Data Form-001 (Mahasiswa)	DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	
Generate Form-001 (Mahasiswa)	GET	http://127.0.0.1:8000/mahasiswa-generate-form-001/{id}	
Daftar Data Mahasiswa Kerja Praktik (Mahasiswa)	GET	http://127.0.0.1:8000/dashboard-mahasiswa-kp	
	POST	http://127.0.0.1:8000/dashboard-mahasiswa-kp	
	PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-kp	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-kp	
	POST	http://127.0.0.1:8000/dashboard-mahasiswa-kp	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	
	PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	
	DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-kp/{id}	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-sidang-kp	
	POST	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp	
	GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-sidang-kp/{id}	
	PUT	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp/{id}	
	DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp/{id}	
	GET	http://127.0.0.1:8000/dashboard-tata-usaha	
	GET	http://127.0.0.1:8000/dashboard-tata-usaha-tambah-data-mahasiswa	
	POST	http://127.0.0.1:8000/dashboard-tata-usaha-tambah-data-mahasiswa	
	GET	http://127.0.0.1:8000/dashboard-tata-usaha-edit-data-mahasiswa/{id}	
	PUT	http://127.0.0.1:8000/dashboard-tata-usaha/{id}	

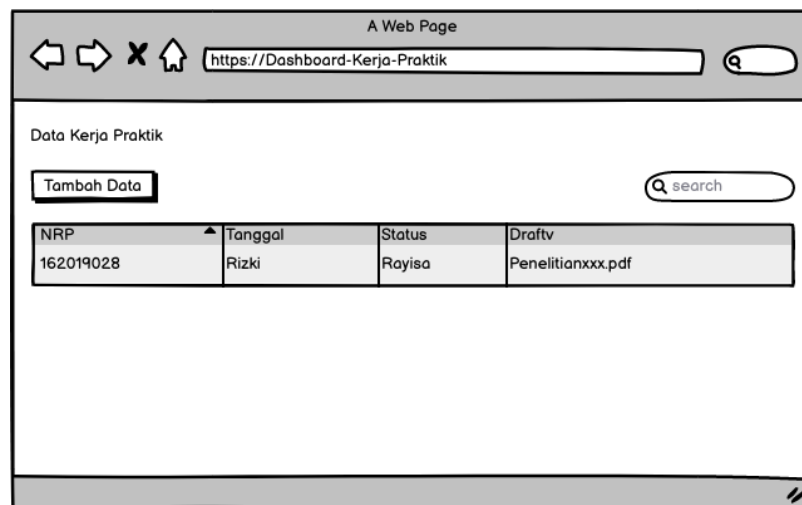
Nama	Method	URL	Deskripsi
	GET	http://127.0.0.1:8000/dashboard-tata-usaha-form-001	
	GET	http://127.0.0.1:8000/dashboard-tata-usaha-edit-form-001/{id}	
	PUT	http://127.0.0.1:8000/dashboard-tata-usaha-form-001/{id}	
	GET	http://127.0.0.1:8000/dashboard-tata-usaha-sidang-kp	
	GET	http://127.0.0.1:8000/dashboard-koordinator-kp	
	GET	http://127.0.0.1:8000/dashboard-koordinator-tambah-kp	
	POST	http://127.0.0.1:8000/dashboard-koordinator-tambah-kp	
	GET	http://127.0.0.1:8000/dashboard-koordinator-edit-kp/{id}	
	PUT	http://127.0.0.1:8000/dashboard-koordinator-sidang-kp/{id}	
	DELETE	http://127.0.0.1:8000/dashboard-koordinator-sidang-kp/{id}	

4.5.6. Rancangan UI

Rancangan UI yang terdapat pada web Tugas Akhir dan Kerja Praktik Sistem Informasi Itenas yang dibuat menggunakan Balsamiq.

4.5.7.1. Tampilan Dashboard Kerja Praktik

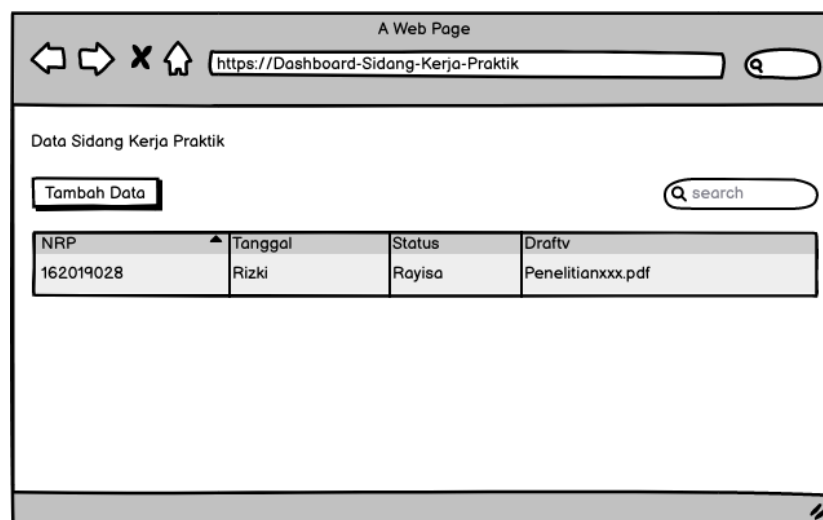
Pada rancangan UI dashboard kerja praktik terdapat tombol yang berfungsi untuk menambahkan data, tabel untuk menampilkan data, dan kolom *search* untuk mencari data.



Gambar 4. 13. Desain *dashboard* tampilan kerja praktik

4.5.7.2. Tampilan *Dashboard* Sidang Kerja Praktik

Pada rancangan UI dashbooard sidang kerja praktik terdapat tombol yang berfungsi untuk menambahkan data, tabel untuk menampilkan data, dan kolom *search* untuk mencari data.



Gambar 4. 14. Desain *dashboard* tampilan sidang kerja praktik

4.5.7.3. Tampilan *Form* Isi Daftar Kerja Praktik

Pada rancangan UI *form* pendaftaran kerja praktik terdapat delapan kolom isian yang berfungsi untuk mengisi data yang diperlukan dalam melakukan pendaftaran kerja praktik.

A Web Page
https://Dashboard-Daftar-Kerja-Praktik

Kode Kerja Praktik	NRP
<input type="text"/>	<input type="text"/>
Nama Perusahaan	Alamat Perusahaan
<input type="text"/>	<input type="text"/>
Bidang Perusahaan	Pembimbing Perusahaan
<input type="text"/>	<input type="text"/>
Tanggal Mulai KP	Tanggal Selesai KP
<input type="text"/>	<input type="text"/>

Submit

Gambar 4. 15. Desain *dashboard* tampilan *form* kerja praktik

4.5.7.4. Tampilan *Form* Isi Daftar Sidang Kerja Praktik

Pada rancangan ui *form* isi daftar sidang kerja praktik terdapat enam *form* isian berisi *form* kode sidang kerja praktik, kode kerja praktik, laporan kerja praktik, ruangan, jam, dan tanggal sidang kerja praktik.

A Web Page
https://Dashboard-Daftar-Kerja-Praktik

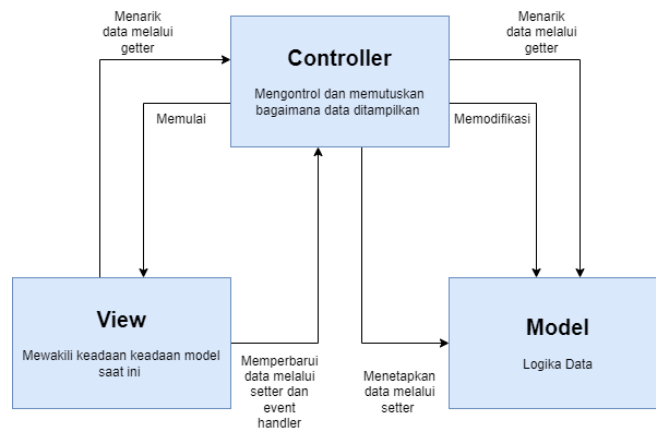
Kode Sidang Kerja Praktik	Kode KP
<input type="text"/>	<input type="text"/>
Laporan Kerja Praktik	Ruangan
<input type="text"/>	<input type="text"/>
Jam	Tanggal
<input type="text"/>	<input type="text"/>

Submit

Gambar 4. 16. Desain *dashboard* tampilan form sidang kerja praktik

4.5.7. Arsitektur MVC

Berikut merupakan *View*, *Model*, dan *Controller* yang digunakan untuk merancang *website* tugas akhir dan kerja praktik sistem Informasi itenas. Arsitektur MVC dari proyek ini dapat dilihat pada gambar 31 dan tabel 16.



Gambar 4. 17. Arsitektur MVC pengembangan website tugas akhir dan kerja praktik

Tabel 4. 20. Asritektur MVC yang terdapat pada sistem *website*

No	View	Model	Controller
1.	koordinator_kp.dashboard-koordinator-edit-kp.blade.php		ProposalController.php
2.	koordinator_kp.dashboard-koordinator-edit-sidang-kp.blade.php		TAController .php
3.	koordinator_kp.dashboard-koordinator-kp.blade,php		SidangController .php
4.	koordinator_kp.dashboard-koordinator-sidang-kp.blade.php		
5.	koordinator_kp.dashboard-koordinator-sidang-ta.blade.php		
6.	koordinator_kp.dashboard-koordinator-tambah-kp.blade,php		
7.	koordinator_kp.dashboard-koordinator-tambah-sidang-kp.blade.php		
8.	koordinator_kp.dashboard-mahasiswa-edit-sidang-kp.blade.php		
9.	koordinator_kp.dashboard-mahasiswa-kp.blade,php		
10.	koordinator_kp.dashboard-mahasiswa-sidang-kp.blade.php		
11.	koordinator_kp.dashboard-mahasiswa-tambah-kp.blade.php		

No	View	Model	Controller
12.	koordinator_kp.dashboard-mahasiswa-tambah-sidang-kp.blade.php		
13.	koordinator.dashboard-koordinator-edit-data-dosen.blade.php		KPController .php
14.	koordinator.dashboard-koordinator-edit-data-mahasiswa.blade.php		MahasiswaController .php
15.	koordinator.dashboard-koordinator-edit-proposal-ta.blade.php		SeminarController .php
16.	koordinator.dashboard-koordinator-edit-seminar-ta.blade.php		SidangKPController .php
17.	koordinator.dashboard-koordinator-edit-ta.blade.php		TUProposalSeminar
18.	koordinator.dashboard-koordinator-proposal-ta.blade.php		
19.	koordinator.dashboard-koordinator-seminar-ta.blade.php		
20.	koordinator.dashboard-koordinator-ta-dosen.blade.php		
21.	koordinator.dashboard-koordinator-ta.blade.php		
22.	koordinator.dashboard-koordinator-tambah-data-dosen.blade.php		
23.	koordinator.dashboard-koordinator-tambah-data-mahasiswa.blade.php		
24.	koordinator.dashboard-koordinator-tambah-proposal-ta.blade.php		
25.	koordinator.dashboard-koordinator-tambah-seminar-ta.blade.php		
26.	koordinator.dashboard-koordinator-tambah-ta.blade.php		
27.	dashboard-mahasiswa-edit-proposal-ta.blade.php		
28.	dashboard-mahasiswa-edit-seminar-ta.blade.php		

No	View	Model	Controller
29.	dashboard-mahasiswa-edit-ta.blade.php		
30.	dashboard-mahasiswa-proposal-ta.blade.php		
31.	dashboard-mahasiswa-seminar-ta.blade.php		
32.	dashboard-mahasiswa-sidang-ta.blade.php		
33.	dashboard-mahasiswa-tambah-proposal-ta.blade.php		
34.	dashboard-mahasiswa-tambah-seminar-ta.blade.php		
35.	dashboard-mahasiswa-tambah-ta.blade.php		
36.	dashboard-mahasiswa.blade.php		
37.	dashboard-tata-usaha-proposal-ta.blade.php		
38.	dashboard-tata-usaha-seminar-ta.blade.php		
39.	dashboard-tata-usaha-sidang-ta.blade.php		
40.	dashboard-tata-usaha-tambah-data-mahasiswa.blade.php		
41.	footer.blade.php		
42.	index.blade.php		
43.	layout-dospem-dospenguji-ta.blade.php		
44.	layout-koordinator-ta.blade.php		
45.	layout-mahasiswa.blade.php		
46.	ayout-tata-usaha.blade.php		
47.	login-dosen.blade.php		
48.	login-mahasiswa.blade.php		
49.	login-tu.blade.php		
50.	navbar.blade.php		
51.	sidebar-dospem-dospenguji-ta.blade.php		
52.	sidebar-koordinator-ta.blade.php		
53.	sidebar-tata-usaha.blade.php		
54.	sidebar.blade.php		
55.	welcome.blade.php		

BAB V

IMPLEMENTASI DAN PENGUJIAN

5.1. Spesifikasi Sistem

Perangkat lunak yang dipergunakan untuk merancang dan mengimplementasikan *website* Beasiswa itenas ini adalah sebagai berikut:

Tabel 5. 1. Perangkat lunak yang digunakan untuk pengembangan *website*

Perangkat Lunak	Deskripsi
Visual Studio Code	Visual Studio Code adalah editor kode dengan dukungan untuk operasi pengembangan seperti debugging, menjalankan tugas, pengontrolan versi, dan proses coding <i>website</i>
Gitlab	GitLab adalah repositori kode dan platform pengembangan perangkat lunak kolaboratif sebagai tools pengolahan file program sehingga tidak terjadi konflik antara pengerjaan dengan tim
XAMPP	<i>Software</i> yang digunakan untuk menjalankan testing di localhost
Heidi/PhpMyAdmin	Digunakan untuk pengelolaan, pengembangan, dan administrasi basis data <i>website</i>
Google Chrome	Sebagai <i>website</i> browser untuk menampilkan tampilan <i>website</i>

5.2. Teknologi yang digunakan

Adapun teknologi yang digunakan penulis dalam melakukan pengembangan *website* ini yaitu:

5.2.1. Spesifikasi Komputer

Spesifikasi komputer yang digunakan untuk merancang dan mengimplementasikan *website* Beasiswa itenas ini adalah sebagai berikut:

Tabel 5. 2. Spesifikasi laptop yang digunakan

No	Hardware
1	Processor: Intel Core i7-7500U CPU 2.70GHz
2	Memory: 160GB RAM DDR4
3	Sistem operasi: Microsoft Windows 10 Pro
4	Penyimpanan: 1TB SSD

5.2.2. Teknologi Database

Berikut merupakan teknologi yang digunakan untuk menyimpan data dalam *website* Sistem Kerja Praktik Dan Tugas Akhir Program Studi Sistem Informasi (SITASI)

Tabel 5. 3. Teknologi basisdata yang digunakan dan versinya

Nama Tools	Versi	Deskripsi
Laravel	9.x	Laravel adalah sebuah <i>framework</i> yang digunakan dalam pengembangan <i>website</i> . Bahasa pemrograman yang digunakan dalam <i>framework</i> Laravel ini adalah PHP. Laravel merupakan <i>framework</i> backend yang berarti laravel dapat digunakan untuk membuat aplikasi fullstack menggunakan Laravel
XAMPP	3.3.0	XAMPP adalah <i>software</i> yang digunakan untuk menjalankan web <i>server</i> dan <i>database</i> pada <i>website</i> .

5.2.3. Teknologi Frontend

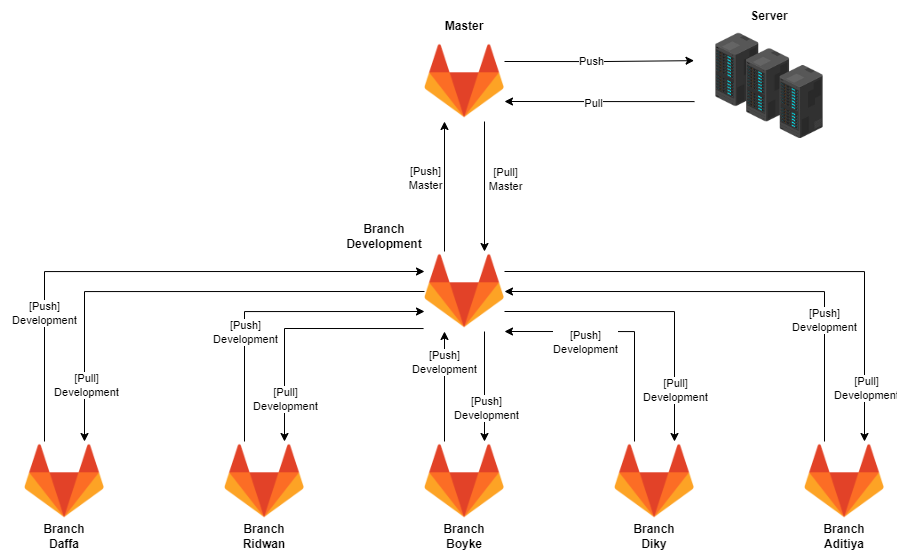
Berikut merupakan teknologi frontend yang digunakan untuk mengembangkan *website* Sistem Kerja Praktik Dan Tugas Akhir Program Studi Sistem Informasi (SITASI)

Tabel 5. 4. Teknologi *front-end* yang digunakan

Nama Tools	Deskripsi
HTML	HTML adalah bahasa digunakan untuk mendesain <i>website</i>
CSS	Fungsi CSS dalam pengembangan <i>website</i> ini adalah untuk memperindah <i>layout website</i> yang telah dibuat dari HTML. Mulai dari <i>font style</i> , warna, ukuran, dan aspek lainnya.
JavaScript	JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan <i>website</i> untuk membuat <i>dropdown menu</i> , <i>forms</i> , dan lainnya dapat interaktif dengan <i>User</i> .
Bootstrap	Bootstrap merupakan <i>framework</i> CSS yang digunakan untuk membantu mendesain <i>layout website</i> .

5.3. Rancangan Strategi Percabangan Git

Gambar di atas merupakan perancangan strategi menggunakan git, yang dimana pada pengerjaan aplikasi *website* tugas akhir dan kerja praktik ini pada branch master berhubungan langsung dengan *server* untuk proses development aplikasi yang telah selesai dibuat.



Gambar 5. 1. Percabangan pada GITLAB

Pada branch development akan menyimpan semua modul yang telah selesai dibuat oleh *developer*. Pada branch Daffa Alif, Ridwan, Boyke, Diky, dan Aditiya menyimpan feature yang telah dikerjakan oleh *developer* dan setelah semua modul selesai dikerjakan akan di push ke branch development. Pada aturan penamaan pada branch feature yaitu menggunakan nama *developer* yaitu *branch_(nama developer)*.

Berikut ini merupakan table dari setiap *branch* yang terdapat pada aplikasi *website* tugas akhir dan kerja praktik sistem Informasi:

Tabel 5. 5. List branch yang terdapat pada Gitlab

ID Branch	Nama Branch	Deskripsi	Developer
Branch_01	Branch Daffa Alif	Pada branch ini berisi modul yang telah dikerjakan yaitu modul tugas akhir, modul proposal tugas akhir, modul seminar, modul kerja praktik, dan modul sidang kerja praktik.	Muhammad Daffa Nur Alif
Brach_02	Branch Ridwan	Pada branch ini berisi modul yang telah dikerjakan yaitu modul sidang tugas akhir, dan yudisium.	Muhammad Ridwan prasetyo

Brach_03	Brach Boyke	Pada branch ini berisi modul yang telah dikerjakan yaitu modul <i>User</i> , dan modul seminar tugas akhir.	Boyke Agung Nugraha
Branch_04	Branch Diky	Pada branch ini berisi modul yang telah dikerjakan yaitu <i>Bug fixing</i> , Modul Memisahkan Koordinator KP-TA, dan Modul Pegajuan Kerja Praktik (Form-001)	Diky Akmal Fauzi

Keterangan:

Warna kuning: modul yang dibuat oleh penulis

5.4. Pengujian API

Pengujian API dilakukan menggunakan aplikasi Postman yang dimana dapat menampilkan status, time dan size akses dari URL yang digunakan. Berikut ini hasil dari pengujian API.

5.5. Tabel 5. 6. Pengujian API pada *website*

Method	URL	Status	Time	Size
GET	http://127.0.0.1:8000/dashboard-mahasiswa-form-001	200	403 ms	32.13 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-form-001	200	387 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-mahasiswa-form-001	419	546 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	200	295 ms	42.13 KB
PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	405	496 ms	883.82 KB
DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-edit-form-001/{id}	405	678 ms	883.82 KB
GET	http://127.0.0.1:8000/mahasiswa-generate-form-001/{id}	200	309 ms	42.13 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-kp	200	303 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-mahasiswa-kp	419	299 ms	7.16 KB
PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	405	578 ms	883.82 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-kp	200	345 ms	42.13 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-kp	200	387 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-mahasiswa-kp	419	288 ms	7.16 KB

Method	URL	Status	Time	Size
GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	200	306 ms	42.13 KB
PUT	http://127.0.0.1:8000/dashboard-mahasiswa-edit-kp/{id}	405	560 ms	883.7 KB
DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-kp/{id}	405	485	883.8 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp	200	303 ms	42.13 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-tambah-sidang-kp	200	415 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp	419	429 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-mahasiswa-edit-sidang-kp/{id}	200	293 ms	42.13 KB
PUT	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp/{id}	419	436 ms	7.16 KB
DELETE	http://127.0.0.1:8000/dashboard-mahasiswa-sidang-kp/{id}	419	398 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha	200	287 ms	42.17 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha-tambah-data-mahasiswa	200	303 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-tata-usaha-tambah-data-mahasiswa	419	357 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha-edit-data-mahasiswa/{id}	200	332 ms	42.13 KB
PUT	http://127.0.0.1:8000/dashboard-tata-usaha/{id}	419	387 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha-form-001	200	286 ms	42.13 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha-edit-form-001/{id}	200	295 ms	42.13 KB
PUT	http://127.0.0.1:8000/dashboard-tata-usaha-form-001/{id}	419	291 ms	7.16 KB
GET	http://127.0.0.1:8000/dashboard-tata-usaha-sidang-kp	200	299 ms	42.13 KB
GET	http://127.0.0.1:8000/dashboard-koordinator-kp	200	288 ms	42.17 KB
GET	http://127.0.0.1:8000/dashboard-koordinator-tambah-kp	200	303 ms	42.13 KB
POST	http://127.0.0.1:8000/dashboard-koordinator-tambah-kp	500	682 ms	1.19 MB
GET	http://127.0.0.1:8000/dashboard-koordinator-edit-kp/{id}	405	606 ms	883.8 KB

Method	URL	Status	Time	Size
PUT	http://127.0.0.1:8000/dashboard-koordinator-sidang-kp/{id}	419	260 ms	7.16 KB
DELETE	http://127.0.0.1:8000/dashboard-koordinator-sidang-kp/{id}	419	299 ms	7.16 KB

Dari hasil pengujian dapat disimpulkan bahwa URL dengan GET method menampilkan status 200 OK dan selain dari itu URL dengan POST, PUT, dan DELETE method menampilkan 419, dan 405 dikarenakan pada kodingan pembuatan *website* ini belum menggunakan API backend dan pengetesan API berjalan lancar.

5.6.List Bug Pada Website SITASI

Terdapat 6 bug pada tampilan aplikasi ANBK yang telah diidentifikasi, dengan keterangan 2 pada bagian user Mahasiswa, 1 pada Koordinator TA, 2 pada Koordinator KP, dan CSS *render bug* pada semua halaman user.

No	Tanggal	List Bug	Menu	User
1	25 Oktober 2023	CRUD Daftar Kerja Praktik Mahasiswa	Daftar Kerja Praktik Mahasiswa	Mahasiswa
2	25 Oktober 2023	CRUD Sidang Kerja Praktik	Sidang Kerja Praktik Mahasiswa	Mahasiswa
3	26 Oktober 2023	Login Koordinator Tugas Akhir tidak berfungsi	Login Koordinator Tugas Akhir	Koordinator TA
4	26 Oktober 2023	CSS <i>render bug</i> pada semua layout. Halaman website SITASI akan memuat halaman HTML terlebih dahulu, sebelum CSS-nya membuat semua halaman <i>website</i> tanpa <i>style</i> .	View Mahasiswa, Dosen, Koordinator KP, Koordinator TA, Tata Usaha, Yudisium	Mahasiswa, Dosen, Koordinator KP, Koordinator TA, Tata Usaha
5	27 Oktober 2023	Routing Sidebar	View Koordinator KP	Koordinator KP

6	28 Oktober 2023	Routing Dashboard	View Koordinator KP	Koordinator KP
---	-----------------	-------------------	---------------------	----------------

Tabel diatas merupakan list bug pada website SITASI yang diperuntukkan untuk penulis. Semua bug ini disampaikan oleh developer modul sebelumnya kepada penulis.

5.7. Implementasi Perbaikan

Semua perbaikan dilaksanakan oleh penulis menggunakan tools Visual Studio Code dengan framework Laravel. Table 18 merupakan list dari implementasi perbaikan beserta bukti pada website SITASI yang telah diperbaiki oleh penulis.

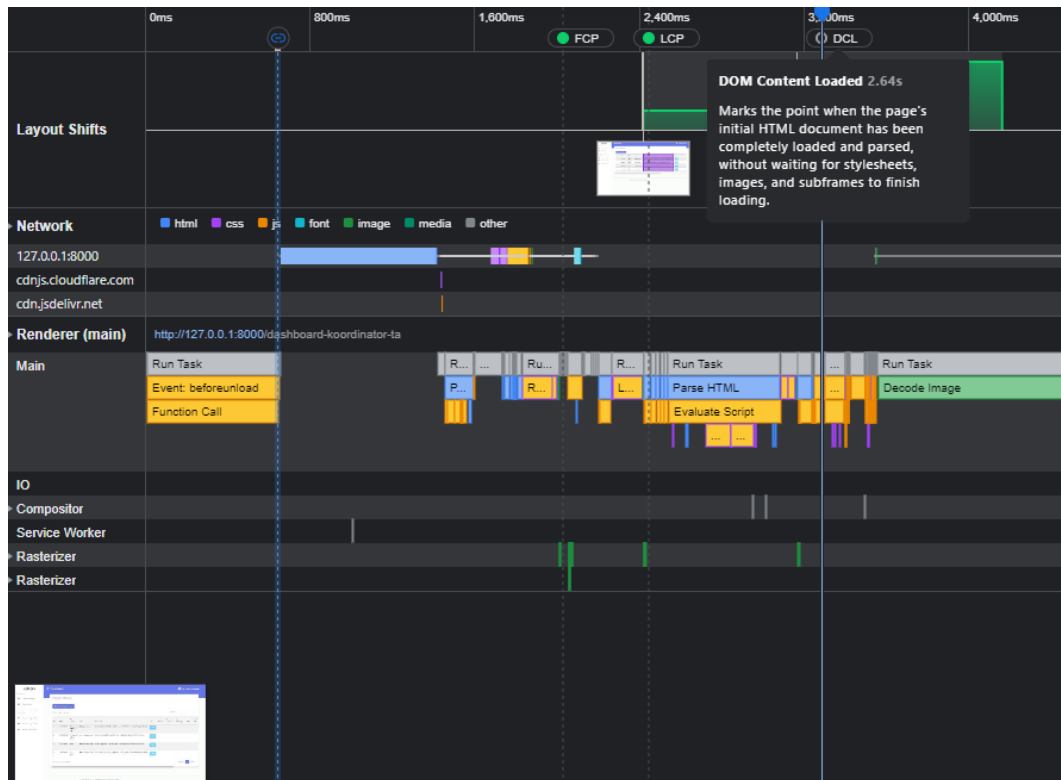
5.8. Pengujian Skenario Kasus

No	Use Case	Hasil Uji [Berhasil Gagal]
1	<p>Use Case Uji: Mendaftarkan Peserta Kerja Praktik</p> <p>Aktor: Mahasiswa</p> <p>Kasus Pengujian:</p> <ul style="list-style-type: none"> a Menambahkan data pada tabel pendaftaran kerja praktik b Melihat data pada tabel pendaftaran kerja praktik c Update data pada tabel pendaftaran kerja praktik <p>Hasil yang diharapkan</p> <ul style="list-style-type: none"> a Berhasil menambahkan data pada tabel pendaftaran kerja praktik b Berhasil melihat data pada tabel pendaftaran kerja praktik c Berhasil update data pada tabel pendaftaran kerja praktik 	Berhasil
2	<p>Use Case Uji: Mengelola Data Kerja Praktik Mahasiswa</p> <p>Aktor: Koordinator Tugas Akhir</p>	

	Kasus Pengujian: a Menentukan Pembimbing b Menentukan Penguji Hasil yang diharapkan a Berhasil Menentukan Pembimbing b Berhasil Menentukan Penguji	Berhasil
	Use Case Uji: Mengelola Data Tugas Akhir Mahasiswa Aktor: Koordinator Kerja praktik Kasus Pengujian: c Menentukan Pembimbing d Menentukan Penguji Hasil yang diharapkan c Berhasil Menentukan Pembimbing d Berhasil Menentukan Penguji	Berhasil

5.9. Pengujian Performance

Sebelumnya website SITASI memerlukan waktu sebanyak selama 3,2 detik untuk memuat halaman secara sempurna. Masalah tersebut disebabkan karena kesalahan yang terdapat pada kode bagian *Views* yang menjadikan *website* memuat halaman HTML terlebih dahulu, lalu memuat CSS-nya. Dari hasil pengujian performa yang dilakukan setelah *render bug* diperbaiki, waktu yang digunakan untuk memuat satu halaman *website* didapatkan selama 26400ms yang artinya diperlukan waktu selama 2.64 detik untuk memuat halaman secara sempurna. Yaitu 0.560 detik lebih cepat dan halaman *website* langsung ditampilkan dengan CSS.



Gambar 5. 8. Performa test website pengembangan website tugas akhir dan kerja praktik

BAB VI

PENUTUP

Pada bab ini, berisikan kesimpulan yang menjelaskan secara terperinci apa saja yang telah dilaksanakan dan apa saja *output* yang dihasilkan pada pembangunan *website*.

6.1. Kesimpulan

Setelah melakukan pengembangan aplikasi Tugas Akhir dan Kerja Praktik untuk program studi Sistem Informasi Itenas, maka dapat disimpulkan, diantaranya sebagai berikut:

1. Dari pengembangan aplikasi *website* tugas akhir dan kerja praktik terdapat beberapa fungsionalitas yang terselesaikan yaitu fungsionalitas pendaftaran tugas akhir, pendaftaran sidang proposal tugas akhir, pendaftaran seminar tugas akhir, pendaftaran kerja praktik, pendaftaran sidang kerja praktik.
2. Pengembangan aplikasi *website* tugas akhir dan kerja praktik ini menggunakan *framework* Laravel dan Bootstrap yang dapat membantu dalam memaksimalkan penggunaan bahasa pemrograman PHP dan dapat mempermudah dan membantu mengorganisir sumber daya yang ada pada *website*.
3. Kendala yang didapat pada saat pengembangan aplikasi *website* tugas akhir dan kerja praktik adalah lamanya *loading* halaman pada saat masuk ke halaman *dashboard website* dikarenakan kesalahan kodingan pada saat pengembangan bagian *View* pada *website*.

6.2. Saran

Saran untuk pengembang selanjutnya dari aplikasi *website* Tugas Akhir dan Kerja Praktik ini yaitu adalah sebagai berikut.

1. Agar menyelesaikan masalah yang terjadi pada saat memuat memuat halaman dan aplikasi *website* segera dapat diimplementasikan oleh program studi Sistem Informasi Itenas Bandung.

Memisahkan Tugas akhir dan Kerja Praktik dalam *dashboard* yang berbeda agar aktor yang terdapat pada *website* lebih mudah untuk menggunakan dan dibedakan dalam tugas akhir dan praktik kerja

DAFTAR PUSTAKA

- Anggraini, Yeni, Donaya Pasha, and Aan Setiawan. 2020. "Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework Codeigniter (Studi Kasus : Orbit Station)." *Jurnal Teknologi dan Sistem Informasi (JTSI)* 1(2): 64–70. <http://jim.teknokrat.ac.id/index.php/JTSI>.
- Anita, Komang, Erliyan Redy Susanto, and Agung Deni Wahyudi. 2020. "Aplikasi Lowongan Pekerjaan Berbasis Web Pada Smk Cahaya Kartika." *Jurnal Teknologi dan Sistem Informasi (JTSI)* 1(1): 75–80. <http://jim.teknokrat.ac.id/index.php/sisteminformasi>.
- Febio, Rini Sovia dan Jimmy. 2011. "MEMBANGUN APLIKASI E-LIBRARY MENGGUNAKAN HTML, PHP SCRIPT, DAN MYSQL DATABASE Rini Sovia Dan Jimmy Febio." *Processor* 6(2): 38–54.
- Herdiansah, Arief, Rohmat Indra Borman, and Sonia Maylinda. 2021. "Sistem Informasi Monitoring Dan Reporting Quality Control Proses Laminating Berbasis Web Framework Laravel." *Jurnal Tekno Kompak* 15(2): 13.
- Herman, H, and E Charles. 2022. "Perancangan Dan Implementasi Sistem Manajemen Sekolah Berbasis Website Di Sma Tabgha Menggunakan Kerangka Kerja Scrum." *National Conference for Community Service ...* 4: 1147–54.
<https://journal.uib.ac.id/index.php/nacospro/article/view/7056%0Ahttps://journal.uib.ac.id/index.php/nacospro/article/download/7056/2686>.
- Implementasi, Perancangan D A N. 2022. "RESTFUL API PADA SISTEM INFORMASI." 9(3): 15–23.
- Muhyidin, Muhammad Agus, Muhammad Afif Sulhan, and Agus Sevtiana. 2020. "Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma." *Jurnal Digit* 10(2): 208.
- Nurmi, Nurmi. 2017. "Membangun Website Sistem Informasi Dinas Pariwisata." *Edik Informatika* 1(2): 1–6.

- Putra, Mardi Yudhi. 2020. "Responsive Web Design Menggunakan Bootstrap Dalam Merancang Layout Website." *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS: Journal of Information System* 5(1): 61–70.
- Rahmatika, Anjumi Kholifatu, Fajar Pradana, and Fitra Abdurrachman Bachtiar. 2020. "Pengembangan Sistem Pembelajaran HTML Dan CSS Dengan Konsep Gamification Berbasis Web." *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer* 4(8 Agustus): 2655–63.
- Ramadhan, Rizky Fajar, and Riki Mukhaiyar. 2020. "Penggunaan Database Mysql Dengan Interface PhpMyAdmin Sebagai Pengontrolan Smarthome Berbasis Raspberry Pi." *JTEIN: Jurnal Teknik Elektro Indonesia* 1(2): 129–34.
- Renaldi, Ridwan, and Dimas Aryo Anggoro. 2020. "Sistem Informasi Geografis Pemetaan Sekolah Menengah Atas/Sederajat Di Kota Surakarta Menggunakan Leaflet Javascript Library Berbasis Website." *Emitor: Jurnal Teknik Elektro* 20(2): 109–16.
- Saputra, Dhimas Rosanto, and Ahmad Syazili. "Aplikasi Human Resource Management Berbasis Mvc (Studi Kasus Universitas Bina Darma)." *Bina Darma Conference on Computer Science*: 340–48.