



KERJA PRAKTIK - IF184801

Implementasi Aplikasi Monitoring Kinerja FTEIC ITS berbasis Web

**Fakultas Teknologi Elektro dan Informatika Cerdas - ITS
Surabaya**

Gedung Rektorat Lt.3 Kampus ITS Sukolilo Surabaya

Periode: 1 Agustus 2020 - 31 Oktober 2020

Oleh:

Arif Darma Althia

05111740000182

Pembimbing Jurusan

Waskitho Wibisono, S.Kom., M.Eng., Ph.D

Pembimbing Lapangan

Dr. I Ketut Eddy Purnama, S.T., M.T.

DEPARTEMEN TEKNIK INFORMATIKA

Fakultas Teknologi Elektro dan Informatika Cerdas

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]



KERJA PRAKTIK - IF184801

Implementasi Aplikasi Monitoring Kinerja FTEIC ITS berbasis Web

**Fakultas Teknologi Elektro dan Informatika
Cerdas - ITS Surabaya**

**Gedung Rektorat Lt.3 Kampus ITS Sukolilo
Surabaya**

Periode: 1 Agustus 2020 - 31 Oktober 2020

Oleh:

Arif Darma Althia

05111740000182

Pembimbing Jurusan

Waskitho Wibisono, S.Kom., M.Eng., Ph.D

Pembimbing Lapangan

Dr. I Ketut Eddy Purnama, S.T., M.T.

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2020

[Halaman ini sengaja dikosongkan]

**LEMBAR PENGESAHAN
KERJA PRAKTIK**

**Implementasi Aplikasi Monitoring Kinerja FTEIC ITS
berbasis Web**

Oleh:
Arif Darma Althia 05111740000182

Disetujui oleh Pembimbing Kerja Praktik:

1. Waskitho Wibisono, S.Kom., M.Eng., Ph.D.
NIP. 197512202001122002



(Pembimbing Departemen)

2. Dr. I Ketut Eddy Purnama, S.T., M.T.
NIP. 196907301995121001



(Pembimbing Lapangan)

[Halaman ini sengaja dikosongkan]

Implementasi Aplikasi Monitoring Kinerja FTEIC ITS berbasis Web

Nama Mahasiswa : Arif Darma Althia
NRP : 05111740000182
Departemen : Informatika FTEIC-ITS
Pembimbing Jurusan : Waskitho Wibisono,
S.Kom., M.Eng., Ph.D.

Abstrak

Dosen serta tenaga kependidikan di Fakultas Teknologi Elektro dan Informatika Cerdas memiliki *track record* yang sangat banyak tiap tahunnya. Mulai dari prestasi, kuliah tamu, training, sertifikasi, konferensi dan jurnal. Karena masalah tersebut, diperlukan adanya suatu sistem informasi untuk menampung semua catatan yang ada agar data yang tersimpan dapat digunakan dengan lebih baik.

Sehingga perlu dibangun aplikasi untuk melakukan monitoring terhadap kinerja yang telah dilakukan terhadap semua kegiatan yang terjadi di FTEIC. Aplikasi ini mampu untuk melakukan proses administratif sekaligus melakukan masukan setiap catatan pada kategorinya masing-masing.

Pembuatan aplikasi menggunakan beberapa *framework*. Untuk *frontend* menggunakan React.js. Lalu untuk *backend* menggunakan Django. basis data yang digunakan adalah MySQL. Aplikasi Web didesain menggunakan paradigma *Single Page Application* sehingga tampilan terlihat lebih dinamis dalam satu buah halaman.

Kata kunci:

FTEIC, Monitoring, Single Page Application

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa karena atas berkat limpahan rahmat dan lindungan-Nya penulis dapat melaksanakan salah satu kewajiban sebagai mahasiswa Teknik Informatika ITS yaitu Kerja Praktik (KP). Penulis menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini, namun kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan penulisan buku laporan ini. Melalui laporan ini penulis juga ingin menyampaikan rasa terima kasih kepada kepada orang-orang yang telah membantu dalam pelaksanaan kerja praktik hingga penyusunan laporan Kerja praktik baik secara langsung maupun tidak langsung. Orang-orang tersebut antara lain adalah:

1. Orang tua penulis,
2. Bapak Waskitho Wibisono, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing kerja praktik yang telah membimbing penulis selama kerja praktik berlangsung.
3. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., Ph.D selaku koordinator Kerja Praktik.
4. Bapak Dr. I Ketut Eddy Purnama, S.T., M.T. selaku pembimbing lapangan selama kerja praktik yang telah memberikan bimbingan serta ilmunya kepada penulis.
5. Sahabat penulis.

Surabaya, Januari 2021

Penulis

[Halaman ini sengaja dikosongkan]

Daftar Isi

LEMBAR PENGESAHAN	5
KERJA PRAKTIK	5
ABSTRAK.....	7
KATA PENGANTAR	9
DAFTAR GAMBAR	14
DAFTAR TABEL	15
BAB I PENDAHULUAN	1
1.1. LATAR BELAKANG.....	1
1.2. TUJUAN.....	1
1.3. MANFAAT	1
1.4. RUMUSAN PERMASALAHAN	1
1.5. LOKASI DAN WAKTU KERJA PRAKTIK	2
1.6. METODOLOGI KERJA PRAKTIK.....	2
1.6.1. <i>Perumusan Masalah</i>	2
1.6.2. <i>Studi Literatur</i>	2
1.6.3. <i>Analisis dan Perancangan Sistem</i>	3
1.6.4. <i>Implementasi</i>	3
1.6.5. <i>Pengujian dan Evaluasi</i>	3
1.7. SISTEMATIKA LAPORAN	3
1.7.1. <i>Bab I Pendahuluan</i>	4
1.7.2. <i>Bab II Profil Perusahaan</i>	4
1.7.3. <i>Bab III Tinjauan Pustaka</i>	4
1.7.4. <i>Bab IV Analisis dan Perancangan Sistem</i>	4
1.7.5. <i>Bab V Implementasi Sistem</i>	4
1.7.6. <i>Bab VI Pengujian dan Evaluasi</i>	4
1.7.7. <i>Bab VII Kesimpulan dan Saran</i>	4
BAB II PROFIL INSTANSI	7
2.1 PROFIL INSTANSI	7
2.2 STRUKTUR ORGANISASI	7
2.3 LOKASI INSTANSI	8

BAB III TINJAUAN PUSTAKA	9
5.3. APLIKASI WEB	9
5.4. PYTHON	9
5.5. DJANGO	9
5.6. JAVASCRIPT	10
5.7. HTML	10
5.8. CSS.....	10
5.9. REACT.....	10
5.10. WEB SERVER.....	11
5.11. VISUAL STUDIO CODE	11
BAB IV.....	13
4.1. ANALISIS SISTEM.....	13
4.1.1 <i>Definisi Umum Aplikasi</i>	13
4.1.2 <i>Analisis Kebutuhan</i>	13
4.2. DIAGRAM KASUS PENGGUNAAN	15
4.3. SPESIFIKASI KASUS PENGGUNAAN APLIKASI E-LEARNING	15
4.3.1 <i>Membuka Halaman Awal</i>	15
4.3.2 <i>Melakukan Login</i>	16
4.3.3 <i>Memasukan data baru</i>	17
4.3.4 <i>Membuat User Baru</i>	18
4.3.5 <i>Memvalidasi Data Baru</i>	19
4.4. DIAGRAM AKTIVITAS.....	21
4.4.1 <i>Membuka Halaman Awal</i>	21
4.4.2 <i>Melakukan Login</i>	21
4.4.3 <i>Memasukan Data Baru</i>	22
4.4.4 <i>Membuat User Baru</i>	23
<i>Gambar 4.5 berikut merupakan Diagram Aktivitas dari Aplikasi</i> <i>Monitoring Kinerja FTEIC untuk membuat user baru.</i>	23
4.4.5 <i>Memvalidasi Data Baru</i>	24
4.5. CONCEPTUAL DATA MODEL	25
4.6. PHYSICAL DATA MODEL	26
4.7. STRUKTUR TABEL	27
BAB V.....	33
IMPLEMENTASI SISTEM	33
5.1. IMPLEMENTASI ARSITEKTUR SISTEM.....	33

5.2.	IMPLEMENTASI LAPISAN KONTROL DAN LAPISAN MODEL.....	34
5.2.1	<i>Lapisan Kontrol Aplikasi Monitoring Kinerja FTEIC.....</i>	34
5.2.2	<i>Lapisan Model Aplikasi Monitoring Kinerja FTEIC</i>	47
5.3	IMPLEMENTASI ANTARMUKA PENGGUNA.....	50
5.3.1	<i>Halaman Konferensi</i>	50
5.3.2	<i>Halaman Prestasi</i>	52
5.3.3	<i>Halaman Prestasi Dosen.....</i>	53
5.3.4	<i>Halaman Training.....</i>	55
BAB VI.....	58	
PENGUJIAN DAN EVALUASI.....	58	
6.1	TUJUAN PENGUJIAN	58
6.2	SKENARIO PENGUJIAN.....	58
6.2.1	<i>Membuka Halaman Konferensi.....</i>	58
6.2.2	<i>Membuka Halaman Detail Konferensi</i>	58
6.2.3	<i>Menghapus Konferensi.....</i>	59
6.2.4	<i>Memvalidasi Konferensi</i>	59
6.2.5	<i>Menambah Daftar Konferensi.....</i>	60
6.2.6	<i>Membuka Halaman Prestasi Dosen</i>	60
6.2.7	<i>Membuka Halaman Detail Prestasi Dosen</i>	60
6.2.8	<i>Menghapus Prestasi Dosen</i>	61
6.2.9	<i>Memvalidasi Prestasi Dosen.....</i>	61
6.2.10	<i>Menambah Daftar Prestasi Dosen.....</i>	61
6.2.11	<i>Membuka Halaman Training.....</i>	62
6.2.12	<i>Membuka Halaman Detail Training</i>	62
6.2.13	<i>Menghapus Training.....</i>	62
6.2.14	<i>Memvalidasi Training</i>	63
6.2.15	<i>Menambah Daftar Training.....</i>	63
6.3	EVALUASI PENGUJIAN	64
BAB VII.....	73	
KESIMPULAN	73	
7.1	KESIMPULAN	73
7.2	SARAN	73
DAFTAR PUSTAKA	75	
BIODATA PENULIS.....	77	

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1: Struktur organisasi FTEIC.....	7
Gambar 2.2: Foto Departemen Informatika FTEIC ITS.....	8
Gambar 4.1 Diagram Use Case Modul Scenario.....	15
Gambar 4.2 Diagram Aktivitas Membuka Halaman Awal.....	21
Gambar 4.3 Diagram Aktivitas Melakukan Login.....	22
Gambar 4.4 Diagram Aktivitas Memasukan Data Baru.....	23
Gambar 4.5 Diagram Aktivitas Membuat User Baru.....	24
Gambar 4.6 Diagram Aktivitas Memvalidasi Data Baru.....	25
Gambar 4.7 Conceptual Data Model Aplikasi Monitoring Kinerja FTEIC.....	25
Gambar 4.8 Physical Data Model Aplikasi Monitoring Kinerja FTEIC.....	26
Gambar 5.1 Diagram Arsitektur Sistem.....	33
Gambar 5.2 Halaman Awal Konferensi.....	51
Gambar 5.3 Form Menambahkan Konferensi.....	51
Gambar 5.4 Halaman Hapus Konferensi.....	52
Gambar 5.5 Halaman Awal Prestasi.....	52
Gambar 5.6 Form Menambahkan Prestasi.....	53
Gambar 5.7 Halaman Hapus Prestasi.....	53
Gambar 5.8 Halaman Awal Prestasi Dosen.....	54
Gambar 5.9 Form Menambahkan Prestasi Dosen.....	54
Gambar 5.10 Halaman Hapus Prestasi Dosen.....	55
Gambar 5.11 Halaman Awal Training.....	55
Gambar 5.12 Form Menambahkan Training.....	56
Gambar 5.13 Halaman Hapus Training.....	56

DAFTAR TABEL

Tabel 4.1 Karakteristik Pengguna.....	13
Tabel 4.2 Kebutuhan Fungsional.....	14
Tabel 4.3 Kebutuhan Non-Fungsional.....	14
Tabel 4.4 Tabel Use Case Membuka Halaman Awal.....	16
Tabel 4.5 Tabel Use Case Melakukan Login.....	17
Tabel 4.6 Tabel Use Case Memasukan Data Baru.....	18
Tabel 4.7 Tabel Use Case Membuat User Baru.....	19
Tabel 4.8 Tabel Use Case Memvalidasi Data Baru.....	20
Tabel 4.9 Struktur Tabel Users.....	28
Tabel 4.10 Struktur Prestasi Dosen.....	29
Tabel 4.11 Struktur Tabel Prestasi.....	30
Tabel 4.12 Struktur Tabel Training.....	31
Tabel 4.13 Struktur Tabel Konferensi.....	32
Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Monitoring Kinerja FTEIC.....	70

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

1.1. Latar Belakang

Dari waktu ke waktu, setiap kegiatan yang dilaksanakan di Fakultas Teknologi Elektro dan Informatika Cerdas tidak tercatat dengan baik. Akibatnya, pencarian kegiatan-kegiatan tersebut sulit dilakukan di waktu mendatang.

Masalah tersebut memunculkan ide untuk membangun sebuah aplikasi untuk melakukan pengawasan terhadap seluruh catatan yang dapat disajikan secara terperinci dan terstruktur.

Aplikasi ini akan mengelompokkan kegiatan berdasarkan kategorinya dan untuk setiap kategori dapat diurutkan berdasarkan tahun, departemen, dan detail kegiatan lainnya.

1.2. Tujuan

Tujuan Kerja praktik kali ini adalah melakukan implementasi aplikasi monitoring kinerja FTEIC ITS sekaligus menuntaskan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember.

1.3. Manfaat

Manfaat dari implementasi aplikasi ini adalah memudahkan pencarian dan penyortiran catatan kegiatan yang ada pada FTEIC ITS yang akan berguna di waktu yang akan datang.

1.4. Rumusan Permasalahan

Berikut rumusan masalah dalam pelaksanaan kerja praktik implementasi aplikasi monitoring kinerja FTEIC ITS berbasis Web:

- Bagaimana membangun aplikasi monitoring kinerja FTEIC ITS berbasis web?
- Bagaimana mengelola aplikasi monitoring kinerja FTEIC ITS berbasis web agar lebih fleksibel dan optimal?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi : *Online*
Alamat : Perumdos Blok U no. 169, Kampus Sukolilo ITS Surabaya
Waktu : 1 Oktober 2020 – 31 Desember 2020
Hari Kerja : Setiap Hari
Jam kerja : Fleksibel

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1.6.1. Perumusan Masalah

Untuk mengetahui domain dan fungsionalitas, dijelaskan secara rinci bagaimana sistem yang harus dibuat. Penjelasan oleh pembimbing lapangan kerja praktik kali ini menghasilkan beberapa catatan mengenai gambaran secara garis besar tentang sistem berbasis website yang sebelumnya telah diterapkan. Setelah mendapatkan gambaran sistem, diskusi lebih lanjut dilakukan guna menentukan rancangan serta *tools* pendukung pembuatan sistem.

1.6.2 Studi Literatur

Pada tahap ini, setelah ditentukannya rancangan *database*, bahasa pemrograman sampai dengan teknologi beserta *tools* tambahan yang digunakan, dilakukan studi literatur lanjut mengenai bagaimana penggunaannya dalam membangun sistem sesuai yang diharapkan.

Dikarenakan aplikasi yang akan dibuat merupakan bagian dari sistem yang sudah terbangun, maka secara garis besar *tools* yang digunakan juga tidak jauh berbeda. Bahasa pemrograman yang

digunakan Python untuk *backend* dan *web server*, dengan bantuan kerangka kerja (framework) Django. Kemudian untuk *frontend*, digunakan javascript dengan bantuan kerangka kerja ReactJS.

1.6.3 Analisis dan Perancangan Sistem

Langkah ini meliputi penjelasan awal tentang sistem. Bagaimana cara kerja sistem dengan skenario tertentu. Dari penjelasan awal telah didapatkan beberapa kebutuhan fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas dan menspesifikkan kebutuhan- kebutuhan tersebut.

1.6.4 Implementasi

Implementasi sistem didasarkan oleh perancangan dan analisis sebelumnya. Semua didasari pada rancangan yang sudah ada sebelumnya dan penentuan *tools* yang telah dilakukan sebelumnya. Penentuan tipe data saat pembuatan table baru pada database disesuaikan juga dengan kebutuhan.

Pengerjaan dilakukan dengan progres setiap hari, dengan setiap harinya menargetkan perkembangan dari hari sebelumnya. Progres penyelesaian aplikasi terus dipantau oleh Pembimbing Lapangan.

1.6.5 Pengujian dan Evaluasi

Pengujian dilakukan oleh pembimbing lapangan setiap bagian dari fitur telah selesai dikerjakan untuk memberikan evaluasi ketika ada yang tidak sesuai, dan persetujuan apabila sudah sesuai.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari 7 bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Bab ini berisi tentang latar belakang masalah, tujuan, manfaat, rumusan masalah, lokasi dan waktu kerja praktik, metodologi, dan sistematika laporan.

1.7.2. Bab II Profil Perusahaan

Bab ini berisi sekilas tentang profil Fakultas Teknologi Elektro dan Informatika Cerdas.

1.7.3. Bab III Tinjauan Pustaka

Dalam bab ini dibahas mengenai konsep-konsep pembuatan aplikasi, dasar teori, teknologi yang dipakai dalam pembuatan aplikasi.

1.7.4. Bab IV Analisis dan Perancangan Sistem

Pada bab ini, dijelaskan hasil pembelajaran atau analisis terhadap apa saja yang diperlukan dan harus diperhatikan dalam pengembangan aplikasi yang dikerjakan selama KP.

1.7.5. Bab V Implementasi Sistem

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses implementasi aplikasi.

1.7.6. Bab VI Pengujian dan Evaluasi

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan KP.

1.7.7. Bab VII Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan KP.

[Halaman ini sengaja dikosongkan]

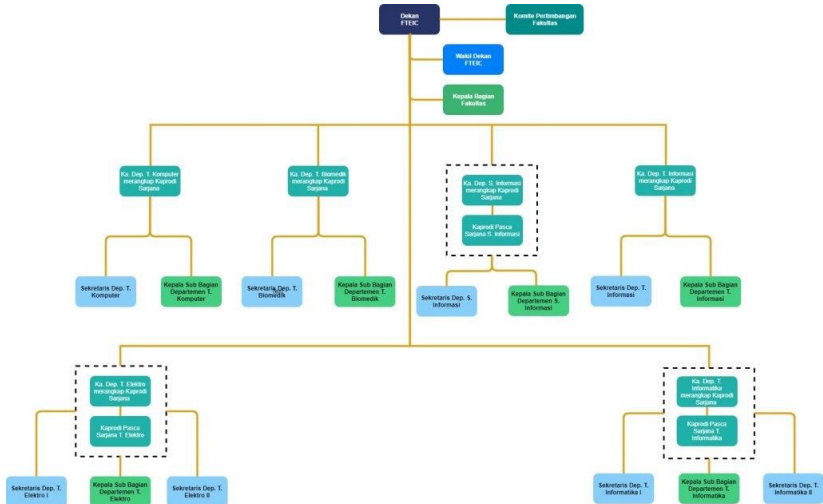
BAB II PROFIL INSTANSI

2.1 Profil Instansi

Fakultas Teknologi Elektro dan Informatika Cerdas (F-Electics) adalah fakultas unggulan di ITS yang didirikan sebagai gabungan dari Fakultas Teknologi Informasi dan Komunikasi (FTIK) dan Fakultas Teknologi Elektro (FTE) yang terdiri dari 6 Departemen dengan lebih dari 2500 mahasiswa baik di tingkat S1, S2 ataupun S3. F-Electics hadir dengan visi membentuk sumber daya manusia berkarakter, berbudi unggul serta berkelas dunia dalam bidang teknologi elektro, sistem informasi, biomedik, komputer, informatika dan teknologi informasi..

2.2 Struktur Organisasi

Struktur organisasi di FTEIC dapat dilihat pada Gambar 2.1.



Gambar 2.3: Struktur organisasi FTEIC

2.3 Lokasi Instansi

Penampakan depan salah satu departemen yang ada pada FTEIC ITS dapat dilihat pada Gambar 2.2.

Alamat : Jl. Teknik Kimia - Gedung Departemen Teknik Informatika

Kampus Institut Teknologi Sepuluh Nopember Surabaya
Jalan Raya ITS, Sukolilo, Surabaya 60111, Indonesia



Gambar 2.4: Foto Departemen Informatika FTEIC ITS

BAB III TINJAUAN PUSTAKA

5.3. Aplikasi Web

Dalam rekayasa perangkat lunak, suatu aplikasi web adalah suatu aplikasi yang diakses menggunakan penjelajah web melalui suatu jaringan seperti Internet atau intranet. Ia juga merupakan suatu aplikasi perangkat lunak komputer yang dikodekan dalam bahasa yang didukung penjelajah web dan bergantung pada penjelajah tersebut untuk menampilkan aplikasi.

5.4. Python

Python adalah Bahasa pemrograman [interpretatif](#) multiguna dengan filosofi perancangan yang berfokus pada tingkat keterbacaan kode. Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python juga didukung oleh komunitas yang besar.

Python mendukung multi paradigma pemrograman, utamanya; namun tidak dibatasi; pada [pemrograman berorientasi objek](#), pemrograman imperatif, dan pemrograman fungsional. Salah satu fitur yang tersedia pada python adalah sebagai [bahasa pemrograman dinamis](#) yang dilengkapi dengan manajemen memori otomatis. Seperti halnya pada bahasa pemrograman dinamis lainnya, python umumnya digunakan sebagai [bahasa skrip](#) meski pada praktiknya penggunaan bahasa ini lebih luas mencakup konteks pemanfaatan yang umumnya tidak dilakukan dengan menggunakan bahasa skrip. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform [sistem operasi](#).

5.5. Django

Django adalah web framework Python yang didesain untuk membuat aplikasi web yang dinamis, kaya fitur dan

aman. Django yang dikembangkan oleh Django Software Foundation terus mendapatkan perbaikan sehingga membuat web framework yang satu ini menjadi pilihan utama bagi banyak pengembang aplikasi web.

5.6. JavaScript

JavaScript adalah sebuah script yang memungkinkan kita mengimplementasikan hal-hal kompleks pada halaman web terutama yang bersifat interaktif. Javascript juga dapat digunakan untuk memanipulasi dan mengirim data pada browser pengguna.

5.7. HTML

Hyper Text Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web, menampilkan berbagai informasi di dalam sebuah penjelajah web Internet dan pemformatan hiperteks sederhana yang ditulis dalam berkas format ASCII agar dapat menghasilkan tampilan wujud yang terintegrasi.

5.8. CSS

Cascading Style Sheets (CSS) merupakan mekanisme sederhana untuk menambahkan *style* (seperti warna tulisan, *font*, jarak tulisan) ke dalam dokumen web. Hampir semua *browser* dan banyak aplikasi yang ada saat ini telah menunjang penggunaan CSS.

5.9. React

React merupakan library javascript yang membantu dalam pembuatan antarmuka pengguna. React memungkinkan penggunaannya untuk membuat tampilan antarmuka dari kumpulan bagian kecil yang disebut komponen.

5.10. Web Server

Web Server adalah sebuah perangkat lunak server yang berfungsi menerima permintaan HTTP atau HTTPS dari klien yang dikenal dengan *web browser* dan mengirimkan kembali hasilnya dalam halaman-halaman web yang umumnya berbentuk dokumen HTML.

5.11. Visual Studio Code

Visual Studio Code adalah editor kode yang dikembangkan oleh Microsoft untuk Windows, Linux dan macOS. Visual Studio Code memberikan dukungan untuk debugging, kontrol Git tertanam, penyorotan sintaksis, penyelesaian kode cerdas, snippet, dan refactoring kode. Visual Studio Code juga dapat disesuaikan, sehingga pengguna dapat mengubah tema editor, pintasan keyboard, dan preferensi.

[Halaman ini sengaja dikosongkan]

BAB IV ANALISIS DAN PERANCANGAN SISTEM

4.1. Analisis Sistem

Pada bab ini akan dijelaskan mengenai tahapan dalam membangun Aplikasi Monitoring Kinerja FTEIC berupa analisis dari sistem yang akan dibangun. Hal tersebut dijelaskan ke dalam dua bagian, yaitu definisi umum aplikasi dan analisis kebutuhan fungsional.

4.1.1 Definisi Umum Aplikasi

Secara umum, Monitoring Kinerja FTEIC merupakan sistem berbasis web yang digunakan untuk memudahkan pendataan terkait beberapa hal yang ada dalam FTEIC seperti pendataan prestasi mahasiswa maupun dosen, lalu pelatihan yang diikuti oleh dosen, kuliah tamu, dan juga jurnal dosen serta konferensi yang diadakan oleh dosen FTEIC. Alasan utama pengembangan aplikasi ini ialah untuk memudahkan pendataan topik terkait yang ada di FTEIC.

Kategori Pengguna	Tugas	Hak Akses ke aplikasi
Admin	<ul style="list-style-type: none">Membuat user, memvalidasi data baru, dan menghapus atau merubah data	Modul <i>Scenario</i>
User	<ul style="list-style-type: none">Menambahkan data baru dan melihat daftar data	Modul <i>Scenario</i>

Tabel 4.1 Karakteristik Pengguna

4.1.2 Analisis Kebutuhan

Dalam aplikasi ini, terdapat fungsi-fungsi yang harus dipenuhi oleh sistem. Kebutuhan ini terbagi ke dalam dua jenis, yakni kebutuhan fungsional dan kebutuhan non- fungsional.

4.1.2.1. Kebutuhan Fungsional

Kebutuhan fungsional pada aplikasi ini menjelaskan bagaimana sistem ini bekerja. Kebutuhan fungsional dari Aplikasi Monitoring Kinerja FTEIC dijelaskan pada Tabel 4.2.

Kode Kebutuhan	Deskripsi kebutuhan	Modul
F01	Membuka Halaman Awal	Scenario
F02	Melakukan Login	Scenario
F03	Admin dan user dapat memasukan data baru	Scenario
F04	Admin dapat membuat user baru	Scenario
F05	Admin dapat memvalidasi data baru	Scenario

Tabel 4.2 Kebutuhan Fungsional

4.1.2.2. Kebutuhan Non-Fungsional

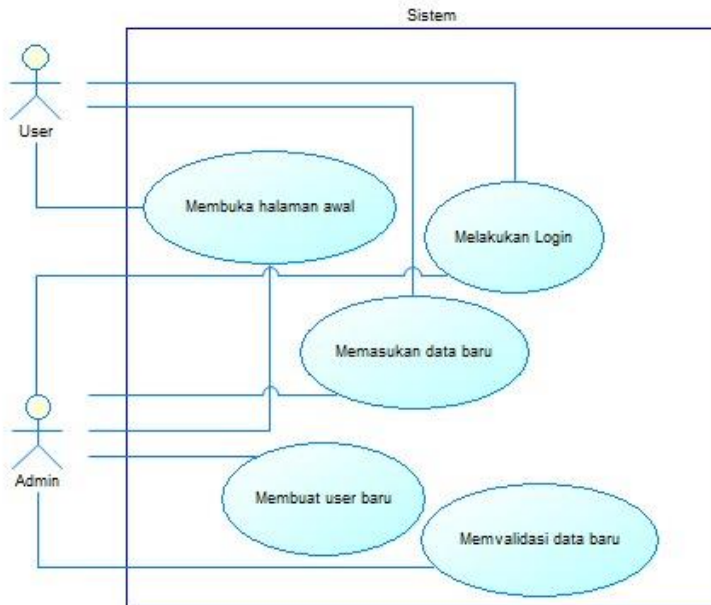
Kebutuhan non-fungsional adalah kebutuhan pengguna untuk mendefinisikan bagaimana Batasan dan karakteristik dari sebuah sistem yang dibangun. Kebutuhan non-fungsional dari aplikasi Monitoring Kinerja FTEIC secara umum serta khusus untuk modul Scenario terdapat pada Tabel 4.3.

Kode Kebutuhan	Deskripsi kebutuhan	Kualitas
NF01	Hanya akun pengguna yang teridentifikasi dan terotorisasi yang dapat menjalankan aplikasi	Security
NF02	Sistem dapat diakses 24 jam	Portability
NF03	Sistem dibuat dalam bentuk aplikasi berbasis web	Portability
NF04	Sistem hanya menampilkan data yang sudah tervalidasi	Reliability
NF05	Data hanya dapat diubah oleh admin	Security

Tabel 4.3 Kebutuhan Non-Fungsional

4.2 Diagram Kasus Penggunaan

Daftar kebutuhan fungsional dapat direpresentasikan menjadi diagram kasus penggunaan (Use Case Diagram) sehingga memudahkan untuk dipahami. Use Case Diagram yang telah dibuat dapat dilihat pada Gambar 4.1.



Gambar 4.1 Diagram Use Case Modul Scenario

4.3 Spesifikasi Kasus Penggunaan Aplikasi E-Learning

4.3.1 Membuka Halaman Awal

Tabel 4.4 berikut merupakan tabel use case dari Aplikasi Monitoring Kinerja FTEIC membuka halaman awal.

Nama	Membuka homepage
Kode	UC001
Deskripsi	Aktor dapat membuka homepage
Tipe	Fungsional

Pemicu	Aktor membuka homepage
Aktor	Seluruh pengguna (admin, user)
Kondisi Awal	-
Kondisi Akhir	Aktor dapat membuka homepage
Alur Kejadian Secara Normal	1. Aktor membuka homepage 2. Sistem menampilkan homepage
Alur Kejadian Alternatif	-
Pengecualian	-

Tabel 4.4 Tabel Use Case Membuka Halaman Awal

4.3.2 Melakukan Login

Tabel 4.5 berikut merupakan tabel use case dari Aplikasi Monitoring Kinerja FTEIC melakukan login.

Nama	Melakukan login
Kode	UC002
Deskripsi	Aktor dapat masuk ke halaman awal dengan akun sesuai role yang dimaksud
Tippe	Fungsional
Pemicu	Aktor menekan tombol ‘sign in’ setelah mengisi email dan password akun untuk admin dan user
Aktor	Seluruh pengguna (admin, user)
Kondisi Awal	Form login ditampilkan
Kondisi Akhir	Aktor memasuki halaman awal (homepage)
Alur Kejadian Secara Normal	1. Aktor mengisi form login 2. Sistem memeriksa field kosong pada form login 3. Sistem mencocokkan data login dengan database 4. Sistem menampilkan halaman awal
Alur Kejadian Alternatif	1. Sistem menampilkan pesan email atau password salah 2. Sistem menampilkan form login 3. Aktor mengisi kembali form login

	4. Sistem mencocokkan data dengan data pengguna pada basis data 5. Jika cocok, sistem menampilkan halaman awal untuk pengguna
Pengecualian	-

Tabel 4.5 Tabel Use Case Melakukan Login

4.3.3 Memasukan data baru

Tabel 4.6 berikut merupakan tabel use case dari Aplikasi Monitoring Kinerja FTEIC memasukan data baru.

Nama	Memasukan Data Baru
Kode	UC003
Deskripsi	Aktor dapat menambahkan data baru
Tipe	Fungsional
Pemicu	Aktor menekan tombol ' <i>Tambah</i> ' di salah satu pilihan halaman, mengisi form dan menekan tombol ' <i>Submit</i> '
Aktor	Seluruh pengguna (admin, user)
Kondisi Awal	Halaman Daftar Data ditampilkan
Kondisi Akhir	Aktor dapat menambahkan data baru
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Aktor menekan tombol '<i>Tambah</i>' di salah satu pilihan halaman 2. Sistem akan menampilkan form data baru 3. Aktor mengisi form data baru 4. Aktor menekan tombol '<i>Submit</i>' 5. Sistem memeriksa field kosong pada form data baru 6. Sistem menyimpan data yang telah diisi pada basis data 7. Sistem menampilkan pesan data baru telah berhasil di entry 8. Sistem menampilkan daftar data
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Sistem menampilkan form data baru kembali

	<ol style="list-style-type: none"> 2. Aktor mengisi kembali form pembuatan scenario 3. Aktor menekan tombol '<i>Submit</i>' 4. Sistem memeriksa field kosong pada form data baru 5. Sistem menyimpan data yang telah diisi pada basis data 6. Sistem menampilkan pesan data baru telah berhasil di entry 7. Sistem menampilkan daftar data
Pengecualian	-

Tabel 4.6 Tabel Use Case Memasukan Data Baru

4.3.4 Membuat User Baru

Tabel 4.7 berikut merupakan tabel use case dari Aplikasi Monitoring Kinerja FTEIC membuat user baru.

Nama	Membuat User Baru
Kode	UC004
Deskripsi	Aktor dapat membuat user baru
Tipe	Fungsional
Pemicu	Aktor menekan tombol ' <i>Tambah Pengguna Baru</i> ' di halaman user
Aktor	Admin
Kondisi Awal	Halaman Awal ditampilkan
Kondisi Akhir	Aktor dapat menambah pengguna baru
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Aktor menekan tombol '<i>Tambah Pengguna Baru</i>' di halaman user 2. Sistem akan menampilkan form pembuatan user baru 3. Aktor mengisi form pembuatan user baru 4. Aktor menekan tombol '<i>Tambah</i>' 5. Sistem memeriksa field kosong pada form pembuatan user baru

	6. Sistem menyimpan data yang telah diisi pada basis data 7. Sistem menampilkan pesan pembuatan user baru telah berhasil 8. Sistem menampilkan halaman awal
Alur Kejadian Alternatif	1. Menampilkan form pembuatan user baru 2. Aktor mengisi kembali form pembuatan user baru 3. Aktor menekan tombol ' <i>Tambah</i> 4. Sistem memeriksa field kosong pada form pembuatan user baru 5. Sistem menyimpan data yang telah diisi pada basis data 6. Sistem menampilkan pesan pembuatan user baru telah berhasil 7. Sistem menampilkan halaman awal
Pengecualian	-

Tabel 4.7 Tabel Use Case Membuat User Baru

4.3.5 Memvalidasi Data Baru

Tabel 4.8 berikut merupakan tabel use case dari Aplikasi Monitoring Kinerja FTEIC memvalidasi data baru.

Nama	Memvalidasi Data Baru
Kode	UC005
Deskripsi	Aktor dapat memvalidasi data baru
Tipe	Fungsional
Pemicu	Aktor menekan tombol ' <i>Validasi</i> ' di salah satu halaman data
Aktor	Admin
Kondisi Awal	Halaman data ditampilkan
Kondisi Akhir	Aktor memasuki halaman data

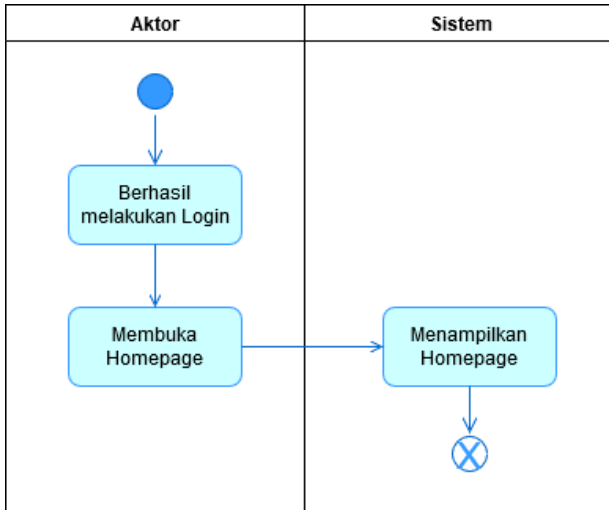
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> 1. Sistem akan menampilkan data yang akan divalidasi 2. Aktor menekan tombol ‘Validasi’ di halaman data 3. Aktor menekan tombol ‘Ya’ 4. Sistem menampilkan pesan data sudah tervalidasi 5. Sistem menampilkan halaman data
Alur Kejadian Alternatif	<ol style="list-style-type: none"> 1. Aktor menutup halaman data 2. Sistem menampilkan halaman data
Pengecualian	-

Tabel 4.8 Tabel Use Case Memvalidasi Data Baru

4.4 Diagram Aktivitas

4.4.1 Membuka Halaman Awal

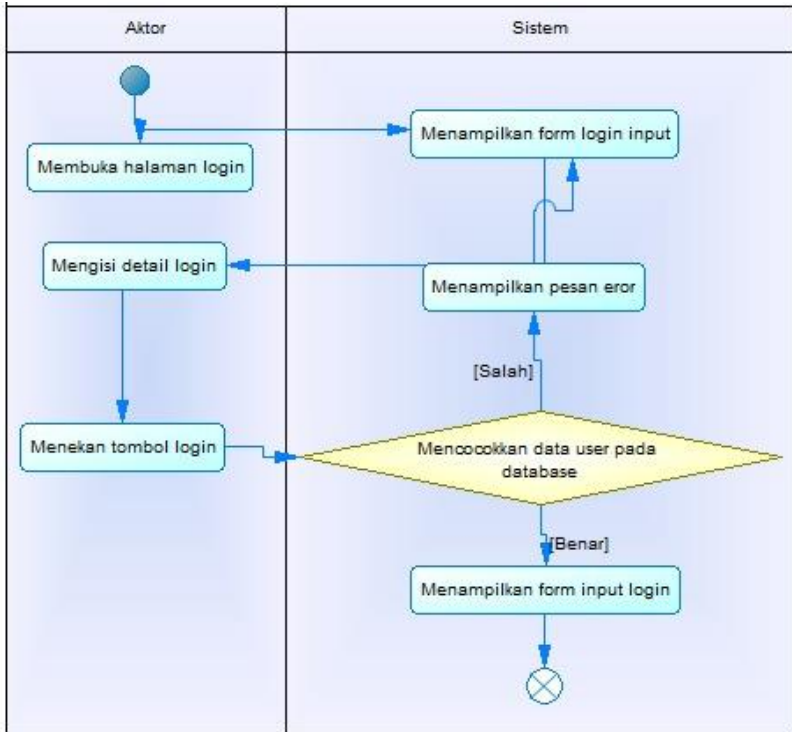
Gambar 4.2 berikut merupakan Diagram Aktivitas dari Aplikasi Monitoring Kinerja FTEIC untuk membuka halaman awal.



Gambar 4.2 Diagram Aktivitas Membuka Halaman Awal

4.4.2 Melakukan Login

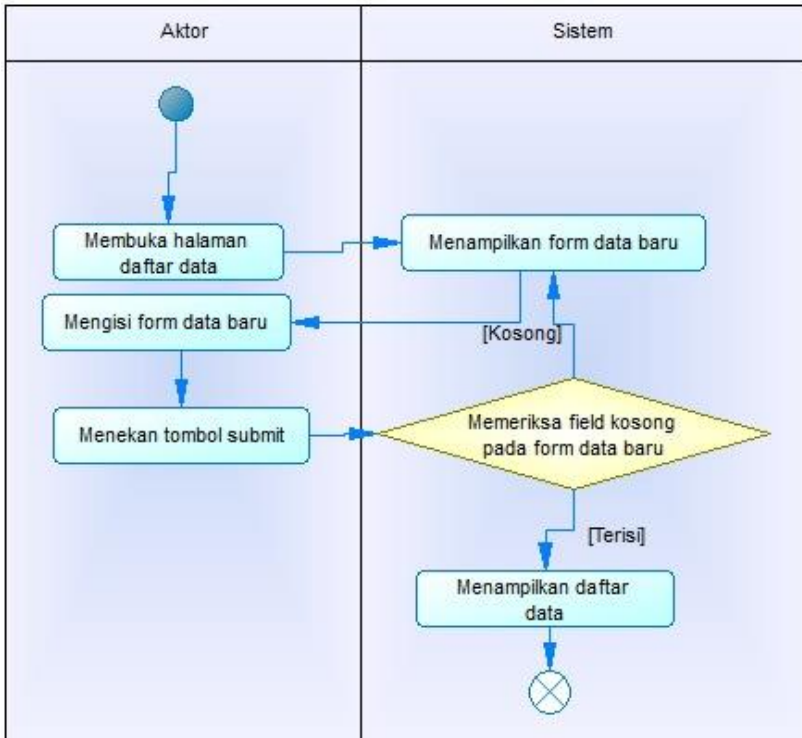
Gambar 4.3 berikut merupakan Diagram Aktivitas dari Aplikasi Monitoring Kinerja FTEIC untuk melakukan login.



Gambar 4.3 Diagram Aktivitas Melakukan Login

4.4.3 Memasukan Data Baru

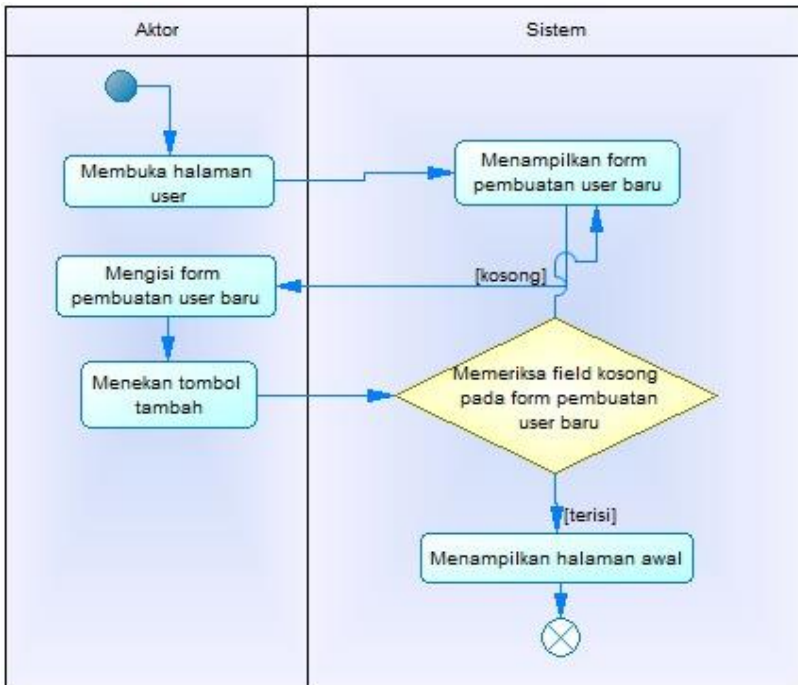
Gambar 4.4 berikut merupakan Diagram Aktivitas dari Aplikasi Monitoring Kinerja FTEIC untuk memasukan data baru.



Gambar 4.4 Diagram Aktivitas Memasukan Data Baru

4.4.4 Membuat User Baru

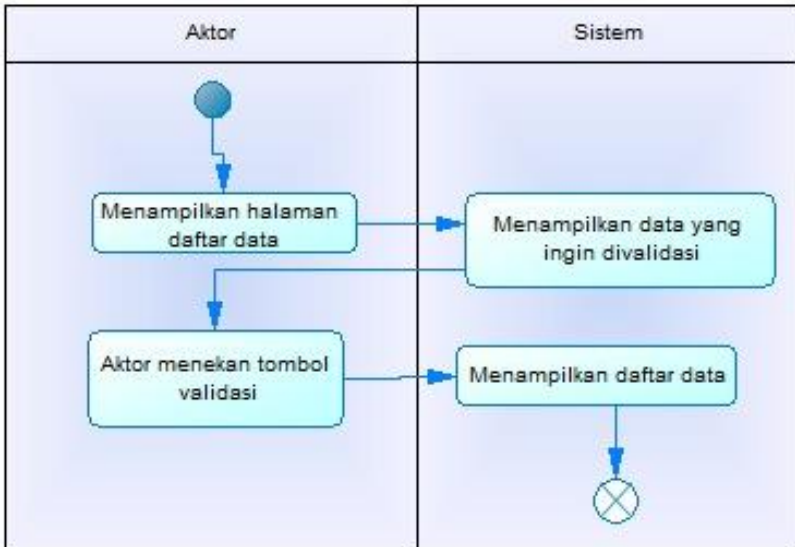
Gambar 4.5 berikut merupakan Diagram Aktivitas dari Aplikasi Monitoring Kinerja FTEIC untuk membuat user baru.



Gambar 4.5 Diagram Aktivitas Membuat User Baru

4.4.5 Memvalidasi Data Baru

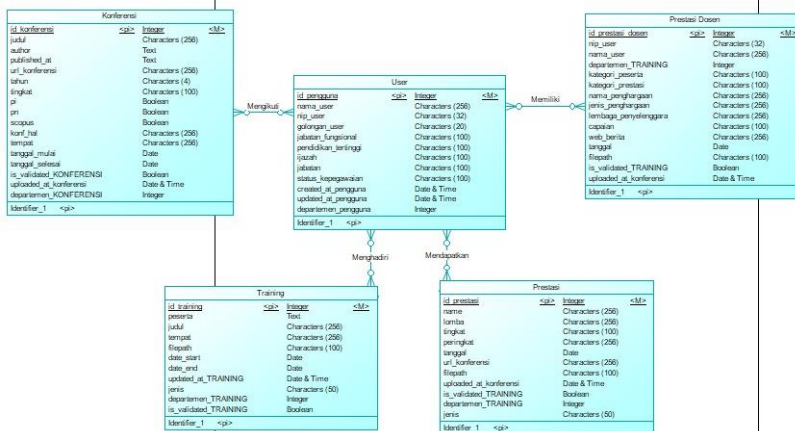
Gambar 4.6 berikut merupakan Diagram Aktivitas dari Aplikasi Monitoring Kinerja FTEIC untuk memvalidasi data baru.



Gambar 4.6 Diagram Aktivitas Memvalidasi Data Baru

4.5 Conceptual Data Model

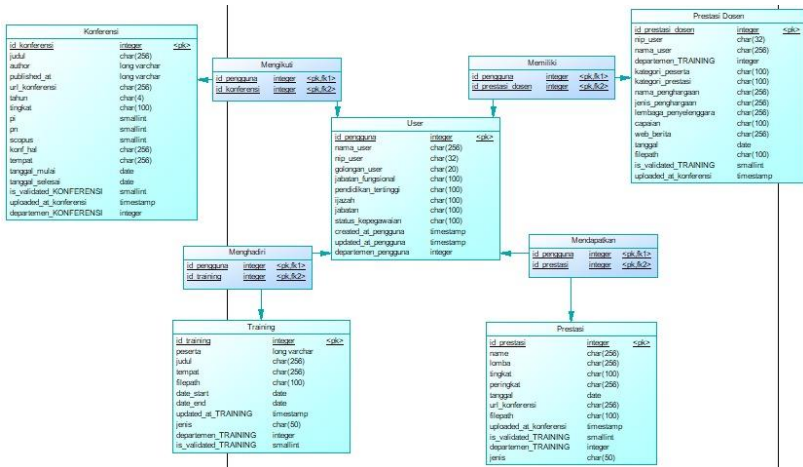
Gambar 4.7 berikut adalah *Conceptual Data Model* dari Aplikasi Monitoring Kinerja FTEIC.



Gambar 4.7 Conceptual Data Model Aplikasi Monitoring Kinerja FTEIC

4.6 Physical Data Model

Gambar 4.8 berikut adalah *Physical Data Model* dari Aplikasi Monitoring Kinerja FTEIC



Gambar 4.8 Physical Data Model Aplikasi Monitoring Kinerja FTEIC

4.7 Struktur Tabel

Berikut adalah struktur tabel pada Aplikasi Monitoring Kinerja FTEIC meliputi nama tabel, nama atribut, tipe data dan keterangan atribut.

Tabel 4.9 berikut adalah struktur tabel users.

Users			
No	Nama Atribut	Tipe Data	Keterangan
1.	user_id	Integer	Primary key dan auto increment
2.	name	Variable Character (255)	Nama pengguna
3.	NIP	Variable Character (32)	NIP pengguna
4.	golongan	Variable Character (32)	Waktu email diverifikasi
5.	jabatan_fungsional	Variable Character (100)	Jabatan fungsional
6.	pendidikan_tertinggi	Variable Character (100)	Pendidikan tertinggi
7.	ijazah	Variable Character (100)	Ijazah
8.	jabatan	Variable Character (100)	Jabatan

9.	status_kepegawaian	Variable Character (100)	Status kepegawaian
10.	created_at	Date Time	Pembuatan row
11.	updated_at	Date Time	Row diupdate
12.	departemen	Integer	Foreign key

Tabel 4.9 Struktur Tabel Users

Tabel 4.10 berikut adalah struktur tabel test.

Test			
No	Nama Atribut	Tipe Data	Keterangan
1.	id	Integer	Primary key dan auto increment
2.	NIP	Variable Character (32)	Nama pengguna
3.	nama	Variable Character (255)	Email pengguna
4.	departemen	Integer	Foreign key
5.	kategori_peserta	Variable Character (100)	Kategori peserta
6.	kategori_prestasi	Variable Character (100)	Kategori prestasi
7.	nama_penghargaan	Variable Character (255)	Nama penghargaan

8.	jenis_penghargaan	Variable Character (255)	Jenis penghargaan
9.	lembaga_penyelenggara	Variable Character (255)	Lembaga penyelenggara
10.	capaian	Variable Character (100)	Capaian
11.	web_berita	Variable Character (255)	Web berita
12.	tanggal	Date	Tanggal prestasi
13.	filepath	Variable Character (100)	Letak file
14.	is_validated	Boolean	Tervalidasi
15.	uploaded_at	Date Time	Tanggal diupload

Tabel 4.10 Struktur Prestasi Dosen

Tabel 4.11 berikut adalah struktur tabel prestasi.

Session			
No	Nama Atribut	Tipe Data	Keterangan
1.	id_prestasi	Integer	Primary key dan auto increment
2.	name	Variable Character (255)	Nama pemenang lomba

3.	lomba	Variable Character (255)	Nama perlombaan
4.	tingkat	Variable Character (100)	Tingkat perlombaan
5.	peringkat	Variable Character (255)	Peringkat pada lomba
6.	tanggal	Date	Tanggal perlombaan
7.	departemen	Integer	Foreign key
8.	url	Variable Character (255)	url web prestasi
9.	filepath	Variable Character (100)	Letak file
10.	is_validated	Boolean	Tervalidasi
11.	uploaded_at	Date Time	Tanggal diupload
12.	jenis	Variable Character (100)	Jenis prestasi

Tabel 4.11 Struktur Tabel Prestasi

Tabel 4.12 berikut adalah struktur tabel Training.

Session			
No	Nama Atribut	Tipe Data	Keterangan
1.	id_training	Integer	Primary key dan auto increment
2.	peserta	Text	Nama peserta training
3.	jenis	Variable Character (50)	Jenis training
4.	judul	Variable Character (255)	Judul training

5.	date_start	Date	Tanggal mulai training
6.	date_end	Date	Tanggal selesai training
7.	tempat	Variable Character (255)	Tempat training
8.	departemen	Integer	Foreign key
9.	filepath	Variable Character (100)	Letak file
10.	is_validated	Boolean	Tervalidasi
11.	uploaded_at	Date Time	Tanggal diupload

Tabel 4.12 Struktur Tabel Training

Tabel 4.13 berikut adalah struktur tabel konferensi.

Participant			
No	Nama Atribut	Tipe Data	Keterangan
1.	id_konferensi	Integer	Primary key dan auto increment
2.	judul	Variable Character (255)	Judul konferensi
3.	author	Text	Pemilik konferensi
4.	published_at	Text	Publish konferensi
5.	url	Variable Character (255)	url web konferensi
6.	tahun	Variable Character (4)	Tahun konferensi
7.	tingkat	Variable Character (50)	Tingkat konferensi

8.	pi	Boolean	Pi dari konferensi
9.	pn	Boolean	Pn dari konferensi
10.	scopus	Boolean	Scopus dari konferensi
11.	konf_hal	Variable Character (255)	Konferensi halaman
12.	tempat	Variable Character (255)	Tempat konferensi
13.	tanggal_mulai	Date	Tanggal mulai konferensi
14.	tanggal_selesai	Date	Tanggal selesai konferensi
15.	is_validated	Boolean	Tervalidasi
16.	uploaded_at	Date Time	Tanggal diupload
17.	departemen	Integer	Foreign key

Tabel 4.13 Struktur Tabel Konferensi

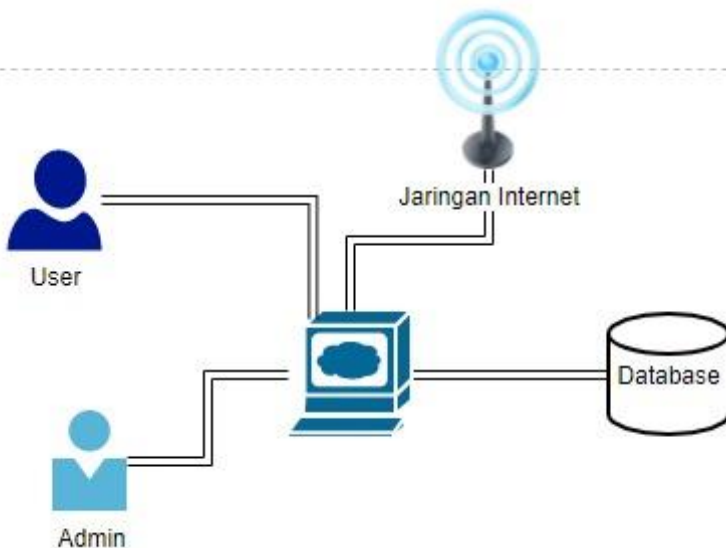
BAB V

IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari perancangan sistem dan pengaplikasian sistem dalam bentuk situs web. Implementasi ini terbagi menjadi 4, yaitu implementasi arsitektur sistem, implementasi lapisan control dan implementasi antarmuka pengguna.

5.1. Implementasi Arsitektur Sistem

Pada bagian ini akan digambarkan arsitektur sistem. Adapun diagram arsitektur sistem yang diterapkan pada Aplikasi Monitoring Kinerja FTEIC terlihat pada gambar 5.1.



Gambar 5.1 Diagram Arsitektur Sistem

5.2. Implementasi Lapisan Kontrol dan Lapisan Model

Implementasi lapisan kontrol berisi logika yang digunakan aplikasi seperti kontrol untuk memasukkan data ke database.

5.2.1 Lapisan Kontrol Aplikasi Monitoring Kinerja FTEIC

5.2.1.1 Konferensi Controller

Lapisan kontrol ini berguna untuk mengatur data pada konferensi. Berikut adalah potongan kode konferensi controller.

```
class KonferensiViewSet (viewsets.ViewSet):
    parser_classes = (MultiPartParser, FormParser)

    def get_all_objects(self, request):
        if request.user.is_admin:
            queryset = Konferensi.objects.all()
        else:
            queryset =
Konferensi.objects.filter(is_validated=True)
        return queryset

    def get_permissions(self):
        if self.action == 'destroy':
            permission_classes = [isAdminPermission]
        else:
            permission_classes = [IsAuthenticated]
        return [permission() for permission in
permission_classes]

    def list(self, request):
        queryset = self.get_all_objects(request)
        serializer = KonferensiSerializer(queryset, many=True)
        return Response(serializer.data)

    def create(self, request):
        serializer = KonferensiSerializer(data=request.data)

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)

        return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)
```

```

def create(self, request):
    serializer = KonferensiSerializer(data=request.data)

    if serializer.is_valid():
        serializer.save()
        return Response(serializer.data,
status=status.HTTP_201_CREATED)

    return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)

def retrieve(self, request, pk=None):
    queryset = self.get_all_objects(request)
    konferensi = get_object_or_404(queryset, pk=pk)
    serializer = KonferensiSerializer(konferensi)
    return Response(serializer.data)

def update(self, request, pk=None):
    pass

def partial_update(self, request, pk=None):
    pass

def destroy(self, request, pk=None):
    queryset = Konferensi.objects.all()
    konferensi = get_object_or_404(queryset, pk=pk)
    konferensi.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)

@api_view(['POST'])
@permission_classes([isAdminPermission])
def set_validate(request, pk):
    if request.method == 'POST':
        queryset = Konferensi.objects.all()
        konferensi = get_object_or_404(queryset, pk=pk)
        if not konferensi.is_validated:
            konferensi.is_validated = True
            konferensi.save()

    return Response(status=status.HTTP_200_OK)

class KonferensiList(ListAPIView):
    permission_classes = (IsAuthenticated, )
    serializer_class = KonferensiSerializer
    queryset = Konferensi.objects.all()
    filter_backends = [DjangoFilterBackend]
    filter_fields = [
        'judul',
        'author',
        'published_at',
        'url',
        'tahun',

```

```

        'tingkat',
        'pi',
        'pn',
        'scopus',
        'konf_hal',
        'tempat',
        'tanggal_mulai',
        'tanggal_selesai',
        'departemen'
    ]

# Export Excel API
@api_view(['GET'])
@permission_classes([IsAuthenticated])
def export_data(request):
    response =
    HttpResponse(content_type='application/vnd.openxmlformats-
    officedocument.spreadsheetml.sheet')
        response['Content-Disposition'] = 'attachment;
        filename={date}-
        konferensi.xlsx'.format(date=datetime.now().strftime('%d-%m-
        %Y'))

    if request.user.is_admin:
        queryset = Konferensi.objects.all()
    else:
        queryset =
Konferensi.objects.filter(is_validated=True)

    param = request.GET
    if 'tahun' in param:
        queryset = queryset.filter(tahun=param['tahun'])
    if 'departemen' in param:
        queryset =
queryset.filter(departemen=param['departemen'])

    workbook = Workbook()

    # Get active worksheet/tab
    worksheet = workbook.active
    worksheet.title = 'Konferensi'

    # Define the titles for columns
    columns = [
        'Judul',
        'Author',
        'Dipublish di',
        'Url',
        'Tahun',
        'Tingkat',
        'PI',
        'PN',

```

```

        'Scopus',
        'Konf Hal.',
        'Tempat',
        'Tanggal Mulai',
        'Tanggal Selesai',
        'Departemen'
    ]
    row_num = 1

    # Assign the titles for each cell of the header
    for col_num, column_title in enumerate(columns, 1):
        cell = worksheet.cell(row=row_num, column=col_num)
        cell.value = column_title

    # Iterate through all kuliah tamu data
    for data in queryset:
        row_num += 1

        # Define the data for each cell in the row
        row = [
            data.judul,
            data.author,
            data.published_at,
            data.url,
            data.tahun,
            data.tingkat,
            int(data.pi),
            int(data.pn),
            int(data.scopus),
            data.konf_hal,
            data.tempat,
            data.tanggal_mulai,
            data.tanggal_selesai,
            data.departemen.nama
        ]

        # Assign the data for each cell of the row
        for col_num, cell_value in enumerate(row, 1):
            cell = worksheet.cell(row=row_num,
            column=col_num)
            cell.value = cell_value

    workbook.save(response)

    return response

```

5.2.1.2 Prestasi Controller

Lapisan kontrol ini berguna untuk mengatur data pada prestasi. Berikut adalah potongan kode prestasi controller.

```
class PrestasiViewSet (viewsets.ViewSet):
    parser_classes = (MultiPartParser, FormParser)

    def get_all_objects(self, request):
        if request.user.is_admin:
            queryset = Prestasi.objects.all()
        else:
            queryset =
Prestasi.objects.filter(is_validated=True)
        return queryset

    def get_permissions(self):
        if self.action == 'destroy':
            permission_classes = [isAdminPermission]
        else:
            permission_classes = [IsAuthenticated]
        return [permission() for permission in
permission_classes]

    def list(self, request):
        queryset = self.get_all_objects(request)
        serializer = PrestasiSerializer(queryset, many=True)
        return Response(serializer.data)

    def create(self, request):
        serializer = PrestasiSerializer(data=request.data)

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)

        return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)

    def retrieve(self, request, pk=None):
        queryset = self.get_all_objects(request)
        prestasi = get_object_or_404(queryset, pk=pk)
        serializer = PrestasiSerializer(prestasi)
        return Response(serializer.data)

    def update(self, request, pk=None):
```



```

        pass

    def partial_update(self, request, pk=None):
        pass

    def destroy(self, request, pk=None):
        queryset = Prestasi.objects.all()
        prestasi = get_object_or_404(queryset, pk=pk)
        prestasi.delete()
        return Response(status=status.HTTP_204_NO_CONTENT)

# Validated API
@api_view(['POST'])
@permission_classes([IsAdminPermission])
def set_validate(request, pk):
    if request.method == 'POST':
        queryset = Prestasi.objects.all()
        prestasi = get_object_or_404(queryset, pk=pk)
        if not prestasi.is_validated:
            prestasi.is_validated = True
            prestasi.save()
        return Response(status=status.HTTP_200_OK)

# Filtering API
class PrestasiList(ListAPIView):
    permission_classes = (IsAuthenticated, )
    serializer_class = PrestasiSerializer
    queryset = Prestasi.objects.all()
    filter_backends = [DjangoFilterBackend]
    filter_fields = ['name', 'lomba', 'tingkat',
'peringkat', 'tanggal', 'url', 'jenis', 'departemen']

# Export Excel API
@api_view(['GET'])
@permission_classes([IsAuthenticated])
def export_data(request):
    response =
HttpResponse(content_type='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet')
    response['Content-Disposition'] = 'attachment;
filename={date}-
prestasi.xlsx'.format(date=datetime.now().strftime('%d-%m-
%Y'))

    if request.user.is_admin:
        queryset = Prestasi.objects.all()
    else:
        queryset =
Prestasi.objects.filter(is_validated=True)

    param = request.GET
    if 'tahun' in param:

```

```

        queryset =
queryset.filter(tanggal__year=param['tahun'])
    if 'departemen' in param:
        queryset =
queryset.filter(departemen=param['departemen'])
    workbook = Workbook()
    # Get active worksheet/tab
    worksheet = workbook.active
    worksheet.title = 'Prestasi'
    # Define the titles for columns
    columns = [
        'Nama',
        'Lomba',
        'Tingkat',
        'Peringkat',
        'Tanggal',
        'Url',
        'Jenis',
        'Departemen'
    ]
    row_num = 1

    # Assign the titles for each cell of the header
    for col_num, column_title in enumerate(columns, 1):
        cell = worksheet.cell(row=row_num, column=col_num)
        cell.value = column_title

    # Iterate through all kuliah tamu data
    for data in queryset:
        row_num += 1

        # Define the data for each cell in the row
        row = [
            data.name,
            data.lomba,
            data.tingkat,
            data.peringkat,
            data.tanggal,
            data.url,
            data.jenis,
            data.departemen.nama
        ]

        # Assign the data for each cell of the row
        for col_num, cell_value in enumerate(row, 1):
            cell = worksheet.cell(row=row_num,
column=col_num)
                cell.value = cell_value

        workbook.save(response)

    return response

```

5.2.1.3 Prestasi Dosen Controller

Lapisan kontrol ini berguna untuk mengatur data pada prestasi dosen. Berikut adalah potongan kode prestasi dosen controller.

```
class PrestasiDosenViewSet(viewsets.ViewSet):
    parser_classes = (MultiPartParser, FormParser)

    def get_all_objects(self, request):
        if request.user.is_admin:
            queryset = PrestasiDosen.objects.all()
        else:
            queryset =
PrestasiDosen.objects.filter(is_validated=True)

        return queryset

    def get_permissions(self):
        if self.action == 'destroy':
            permission_classes = [isAdminPermission]
        else:
            permission_classes = [IsAuthenticated]
        return [permission() for permission in
permission_classes]

    def list(self, request):
        queryset = self.get_all_objects(request)
        serializer = PrestasiDosenSerializer(queryset,
many=True)
        return Response(serializer.data)

    def create(self, request):
        serializer =
PrestasiDosenSerializer(data=request.data)

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)

        return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)

    def retrieve(self, request, pk=None):
        queryset = self.get_all_objects(request)
        data = get_object_or_404(queryset, pk=pk)
        serializer = PrestasiDosenSerializer(data)
        return Response(serializer.data)

    def update(self, request, pk=None):
        pass
```

```

def partial_update(self, request, pk=None):
    pass

def destroy(self, request, pk=None):
    queryset = PrestasiDosen.objects.all()
    data = get_object_or_404(queryset, pk=pk)
    data.delete()
    return Response(status=status.HTTP_204_NO_CONTENT)

# Validated API
@api_view(['POST'])
@permission_classes([isAdminPermission])
def set_validate(request, pk):
    if request.method == 'POST':
        queryset = PrestasiDosen.objects.all()
        prestasi = get_object_or_404(queryset, pk=pk)
        if not prestasi.is_validated:
            prestasi.is_validated = True
            prestasi.save()
        return Response(status=status.HTTP_200_OK)

# Filtering API
class PrestasiDosenList(ListAPIView):
    permission_classes = (IsAuthenticated, )
    serializer_class = PrestasiDosenSerializer
    queryset = PrestasiDosen.objects.all()
    filter_backends = [DjangoFilterBackend]
    filter_fields = '__all__'

# Export Excel API
@api_view(['GET'])
# @permission_classes([IsAuthenticated])
def export_data(request):
    response =
    HttpResponseRedirect(content_type='application/vnd.openxmlformats-
    officedocument.spreadsheetml.sheet')
    response['Content-Disposition'] = 'attachment;
    filename={date}-prestasi-
    dosen.xlsx'.format(date=datetime.now().strftime('%d-%m-%Y'))

    # if request.user.is_admin:
    queryset = PrestasiDosen.objects.all()
    # else:
    #     queryset =
    PrestasiDosen.objects.filter(is_validated=True)

    param = request.GET
    if 'tahun' in param:
        queryset =
    queryset.filter(tanggal__year=param['tahun'])
    if 'departemen' in param:
        queryset =

```

```

queryset.filter(departemen=param['departemen'])

workbook = Workbook()
# Get active worksheet/tab
worksheet = workbook.active
worksheet.title = 'Prestasi Dosen'
# Define the titles for columns
columns = [
    'Nama',
    'NIP',
    'Departemen',
    'KategoriPeserta',
    'KategoriPrestasi',
    'NamaPenghargaan',
    'JenisPenghargaan',
    'LembagaPenyelenggara',
    'Capaian',
    'WebBerita',
    'Tanggal',
]
row_num = 1
# Assign the titles for each cell of the header
for col_num, column_title in enumerate(columns, 1):
    cell = worksheet.cell(row=row_num, column=col_num)
    cell.value = column_title
# Iterate through all kuliah tamu data
for data in queryset:
    row_num += 1
    # Define the data for each cell in the row
    row = [
        data.nama,
        data.NIP,
        data.departemen.nama,
        data.kategori_peserta,
        data.kategori_prestasi,
        data.nama_penghargaan,
        data.jenis_penghargaan,
        data.lembaga_penyelenggara,
        data.capaian,
        data.web_berita,
        data.tanggal
    ]
    # Assign the data for each cell of the row
    for col_num, cell_value in enumerate(row, 1):
        cell = worksheet.cell(row=row_num,
column=col_num)
        cell.value = cell_value

workbook.save(response)

return response

```

5.2.1.4 Training Controller

Lapisan kontrol ini berguna untuk mengatur data pada training. Berikut adalah potongan kode training controller.

```
class TrainingAPIView(APIView):
    parser_classes = (MultiPartParser, FormParser)
    permission_classes = (IsAuthenticated,)

    def get(self, request):
        trains_dosen = Training.objects.all()
        serializer = TrainingSerializer(trains_dosen,
many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = TrainingSerializer(data=request.data)

        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data,
status=status.HTTP_201_CREATED)

        return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)

class TrainingDetailsAPIView(APIView):
    parser_classes = (MultiPartParser, FormParser)
    permission_classes = (IsAuthenticated,)

    def get_object(self, pk):
        try:
            return Training.objects.get(pk=pk)
        except Training.DoesNotExist:
            raise NotFound

    def get(self, request, pk):
        train_dosen = self.get_object(pk)
        serializer = TrainingSerializer(train_dosen)
        return Response(serializer.data)

    def put(self, request, pk):
        train_dosen = self.get_object(pk)
        serializer = TrainingSerializer(train_dosen,
data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data)

        return Response(serializer.errors,
status=status.HTTP_400_BAD_REQUEST)
```

```

def delete(self, request, pk):
    train_dosen = self.get_object(pk)
    if isAdminPermission:
        train_dosen.delete()
        return
Response(status=status.HTTP_204_NO_CONTENT)

# Validated API
@api_view(['POST'])
@permission_classes([isAdminPermission])
def set_validate(request, pk):
    if request.method == 'POST':
        queryset = Training.objects.all()
        train = get_object_or_404(queryset, pk=pk)
        if not train.is_validated:
            train.is_validated = True
            train.save()
        return Response(status=status.HTTP_200_OK)

class TrainingList(ListAPIView):
    permission_classes = (IsAuthenticated, )
    serializer_class = TrainingSerializer
    queryset = Training.objects.all()
    filter_backends = [DjangoFilterBackend]
    filter_fields = [
        'judul',
        'peserta',
        'jenis',
        'date_start',
        'date_end',
        'tempat',
        'departemen',
    ]

# Export Excel API
@api_view(['GET'])
@permission_classes([IsAuthenticated])
def export_data(request):
    response =
HttpResponse(content_type='application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet')
    response['Content-Disposition'] = 'attachment;
filename={date}-
training.xlsx'.format(date=datetime.now().strftime('%d-%m-
%Y'))

    # if request.user.is_admin:
    queryset = Training.objects.all()
    # else:
    #     queryset =
Training.objects.filter(is_validated=True)

```

```

param = request.GET
if 'tahun' in param:
    queryset =
queryset.filter(date_end__year=param['tahun'])
    if 'departemen' in param:
        queryset =
queryset.filter(departemen=param['departemen'])

workbook = Workbook()

# Get active worksheet/tab
worksheet = workbook.active
worksheet.title = 'Training'

# Define the titles for columns
columns = [
    'Judul',
    'Peserta',
    'Jenis',
    'Tempat',
    'Tanggal Mulai',
    'Tanggal Berakhir',
    'Departemen',
]
row_num = 1
# Assign the titles for each cell of the header
for col_num, column_title in enumerate(columns, 1):
    cell = worksheet.cell(row=row_num, column=col_num)
    cell.value = column_title
# Iterate through all kuliah tamu data
for data in queryset:
    row_num += 1
    # Define the data for each cell in the row
    row = [
        data.judul,
        data.peserta,
        data.jenis,
        data.tempat,
        data.date_start,
        data.date_end,
        data.departemen.nama
    ]

    # Assign the data for each cell of the row
    for col_num, cell_value in enumerate(row, 1):
        cell = worksheet.cell(row=row_num,
column=col_num)
        cell.value = cell_value

workbook.save(response)

return response

```


5.2.2 Lapisan Model Aplikasi Monitoring Kinerja FTEIC

5.2.2.1 Konferensi Model

Lapisan ini berguna untuk mengelola konferensi. Berikut adalah potongan code dari Konferensi Model.

```
from django.db import models

from ..masters.models import Departemen

# Create your models here.
class Konferensi(models.Model):
    judul = models.CharField(max_length=255)
    author = models.TextField()
    published_at = models.TextField()
    url = models.CharField(max_length=255)
    tahun = models.CharField(max_length=4)
    tingkat = models.CharField(max_length=50)
    pi = models.BooleanField(default=False)
    pn = models.BooleanField(default=False)
    scopus = models.BooleanField(default=False)
    konf_hal = models.CharField(max_length=255)
    tempat = models.CharField(max_length=255)
    tanggal_mulai = models.DateField()
    tanggal_selesai = models.DateField()
    is_validated = models.BooleanField(default=False)
    uploaded_at = models.DateTimeField(auto_now=True)
    departemen = models.ForeignKey(to=Departemen,
    on_delete=models.DO_NOTHING, null=True)

    def __str__(self):
        return self.judul
```

5.2.2.2 Prestasi Model

Lapisan ini berguna untuk mengelola prestasi. Berikut adalah potongan code dari Prestasi Model.

```
from django.db import models
from ..masters.models import Departemen

# Create your models here.
class Prestasi(models.Model):
    name = models.CharField(max_length=255)
    lomba = models.CharField(max_length=255)
    tingkat = models.CharField(max_length=100)
    peringkat = models.CharField(max_length=255)
    tanggal = models.DateField()
    departemen = models.ForeignKey(to=Departemen,
on_delete=models.DO_NOTHING,null=True)
    url = models.CharField(max_length=255)
    filepath =
models.FileField(upload_to='prestasi/',null=True)
    uploaded_at = models.DateTimeField(auto_now=True)
    is_validated = models.BooleanField(default=False)
    jenis = models.CharField(max_length=100)

    def __str__(self):
        return self.name
```

5.2.2.3 Prestasi Dosen Model

Lapisan ini berguna untuk mengelola prestasi dosen. Berikut adalah potongan code dari Prestasi Dosen Model.

```
from django.db import models
from ..masters.models import Departemen
from ..masters.models import MasterDosen

# Create your models here.
class PrestasiDosen(models.Model):
    NIP = models.CharField(max_length=32, default=None)
    nama = models.CharField(max_length=255, default=None)
    departemen = models.ForeignKey(Departemen,
on_delete=models.CASCADE, null=True)
    kategori_peserta = models.CharField(max_length=100)
    kategori_prestasi = models.CharField(max_length=100)
    nama_penghargaan = models.CharField(max_length=255)
    jenis_penghargaan = models.CharField(max_length=255)
    lembaga penyelenggara = models.CharField(max_length=255)
    capaian = models.CharField(max_length=100)
    web_berita = models.CharField(max_length=255)
    tanggal = models.DateField()
    filepath = models.FileField(upload_to='prestasi/dosen/',
null=True, blank=True)
    is_validated = models.BooleanField(default=False)
    uploaded_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.nama_penghargaan
```

5.2.2.4 Training Model

Lapisan ini berguna untuk mengelola training. Berikut adalah potongan code dari Training Model.

```
from django.db import models
from ..masters.models import Departemen

# Create your models here.
class Training(models.Model):
    peserta = models.TextField()
    jenis = models.CharField(max_length=50)
    judul = models.CharField(max_length=255)
    date_start = models.DateField()
    date_end = models.DateField()
    tempat = models.CharField(max_length=255)
    departemen = models.ForeignKey(to=Departemen,
on_delete=models.DO_NOTHING,null=True)
    filepath =
models.FileField(upload_to='training/',null=True)
    is_validated = models.BooleanField(default=False)
    updated_at = models.DateTimeField(auto_now=True)

    def __str__(self):
        return self.judul
```

5.3 Implementasi Antarmuka Pengguna

5.3.1 Halaman Konferensi

Berikut tampilan antarmuka konferensi, dengan menampilkan opsi menambahkan, melihat data konferensi, merubah, dan menghapus.

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Konferensi > Konferensi

Select konferensi to change ADD KONFERENSI +

Action: Go 0 of 1 selected

<input type="checkbox"/>	JUDUL	AUTHOR	PUBLISHED AT	TAHUN	TINGKAT	URL	TEMPAT	KONF HAL	SCOPUS	PI	PN	DEPARTEMEN
<input type="checkbox"/>	Konferensi(1)	Pemilik konferensi	Dipublikasi	2000	tingkat	alamat_url	tempat	halaman	●	●	●	Informatika

1 konferensi

FILTER

By tahun

All

2000

By tingkat

All

tingkat

Gambar 5.2 Halaman Awal Konferensi

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Konferensi > Konferensi > Add konferensi

Add konferensi

Judul:

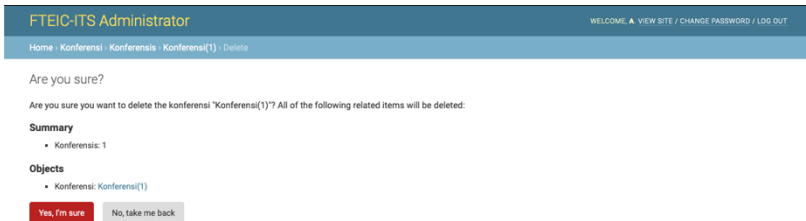
Author:

Published at:

Url:

Tahun:

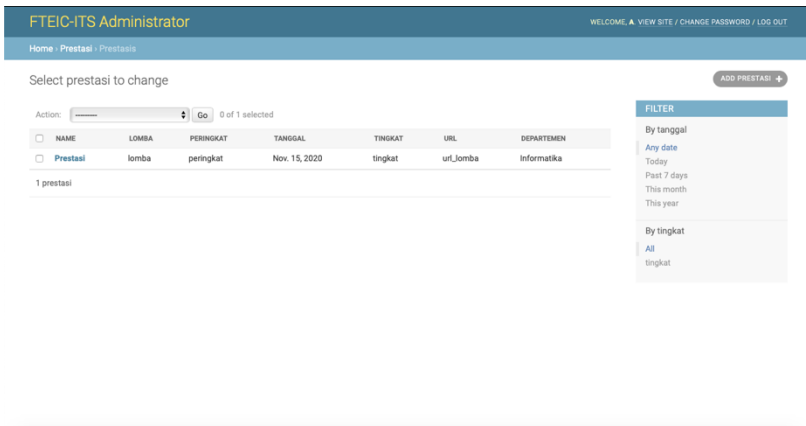
Gambar 5.3 Form Menambahkan Konferensi



Gambar 5.4 Halaman Hapus Konferensi

5.3.2 Halaman Prestasi

Berikut tampilan antarmuka prestasi, dengan menampilkan opsi menambahkan, melihat data prestasi, merubah, dan menghapus.



Gambar 5.5 Halaman Awal Prestasi

The screenshot shows the 'Add prestasi' form in the FTEIC-ITS Administrator interface. The form includes the following fields and elements:

- Name:** A text input field.
- Lomba:** A text input field.
- Tingkat:** A text input field.
- Peringkat:** A text input field.
- Tanggal:** A date picker set to 'Today' with a calendar icon. A note below reads: 'Note: You are 7 hours ahead of server time.'
- Departemen:** A dropdown menu with a plus sign and a close icon.
- Url:** A text input field.
- Filepath:** A file selection button labeled 'Pilih File' and a message 'tidak ada file yang dipilih'.
- Is validated:** A checkbox.
- Jenis:** A text input field.

Gambar 5.6 Form Menambahkan Prestasi

The screenshot shows the 'Are you sure?' confirmation page for deleting a prestasi. The page includes the following elements:

- Header:** 'FTEIC-ITS Administrator' and 'WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT'.
- Breadcrumbs:** 'Home > Prestasi > Prestasi > Prestasi > Delete'.
- Message:** 'Are you sure?' followed by 'Are you sure you want to delete the prestasi "Prestasi"? All of the following related items will be deleted:'.
- Summary:** A list containing 'Prestasi: 1'.
- Objects:** A list containing 'Prestasi: Prestasi'.
- Buttons:** A red button labeled 'Yes, I'm sure' and a grey button labeled 'No, take me back'.

Gambar 5.7 Halaman Hapus Prestasi

5.3.3 Halaman Prestasi Dosen

Berikut tampilan antarmuka prestasi dosen, dengan menampilkan opsi menambahkan, melihat data prestasi dosen, merubah, dan menghapus.

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Prestasidosen > Prestasi dosen

Select prestasi dosen to change ADD PRESTASI DOSEN +

Action: Go 0 of 1 selected

<input type="checkbox"/>	NIP	NAMA	DEPARTEMEN	NAMA PENGHARGAAN	CAPAIAN	TANGGAL	WEB BERITA
<input type="checkbox"/>	NIP	nama	Informatika	penghargaan	capaian	Jan. 10, 2021	web

1 prestasi dosen

FILTER

By tanggal

- Any date
- Today
- Past 7 days
- This month
- This year

By capaian

- All
- capaian

Gambar 5.8 Halaman Awal Prestasi Dosen

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Prestasidosen > Prestasi dosen > Add prestasi dosen

Add prestasi dosen

NIP:

Nama:

Departemen: +

Kategori peserta:

Kategori prestasi:

Nama penghargaan:

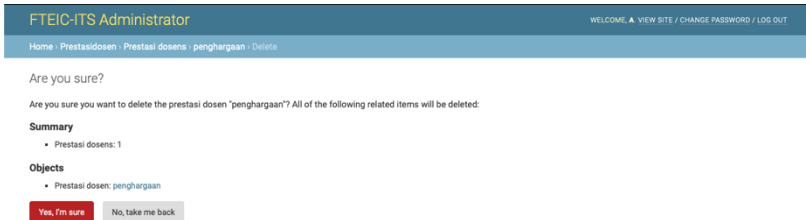
Jenis penghargaan:

Lembaga penyelenggara:

Capaian:

Web berita:

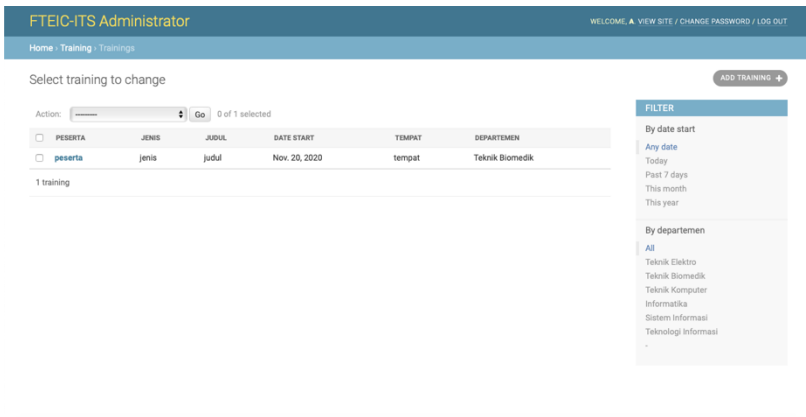
Gambar 5.9 Form Menambahkan Prestasi Dosen



Gambar 5.10 Halaman Hapus Prestasi Dosen

5.3.4 Halaman Training

Berikut tampilan antarmuka training, dengan menampilkan opsi menambahkan, melihat data training, merubah, dan menghapus.



Gambar 5.11 Halaman Awal Training

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Training > Trainings > Add training

Add training

Peserta:

Jenis:

Judul:

Date start: Today Note: You are 7 hours ahead of server time.

Date end: Today Note: You are 7 hours ahead of server time.

Tempat:

Departemen:

Gambar 5.12 Form Menambahkan Training

FTEIC-ITS Administrator WELCOME | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Training > Trainings > a > Delete

Are you sure?

Are you sure you want to delete the training "a"? All of the following related items will be deleted:

Summary

- Trainings: 1

Objects

- Training: a

Gambar 5.13 Halaman Hapus Training

[Halaman ini sengaja dikosongkan]

BAB VI

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dilakukan terhadap Aplikasi Monitoring Kinerja FTEIC. Pengujian dilakukan untuk memastikan kualitas perangkat lunak yang dibangun dan kesesuaian hasil eksekusi perangkat lunak dengan analisis dan perancangan perangkat lunak.

6.1 Tujuan Pengujian

Pengujian dilakukan terhadap Aplikasi Monitoring Kinerja FTEIC guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi.

6.2 Skenario Pengujian

6.2.1 Membuka Halaman Konferensi

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol Konferensi.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat halaman konferensi yang berisi daftar konferensi, *input* untuk pencarian, *filter* pencarian, dan tombol tambah.

6.2.2 Membuka Halaman Detail Konferensi

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol Konferensi.
- Menekan salah satu daftar konferensi.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat

halaman detail konferensi yang berisi judul kegiatan, departemen, URL, konferensi, *check box* scopus, tanggal selesai, author, dipublish di, tingkat, tempat, tanggal mulai, tombol hapus, dan tombol validasi jika konferensi belum divalidasi.

6.2.3 Menghapus Konferensi

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol konferensi.
- Menekan salah satu daftar konferensi.
- Menekan tombol hapus.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah terhapus, dan data sudah tidak ada di halaman konferensi.

6.2.4 Memvalidasi Konferensi

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol konferensi.
- Menekan salah satu daftar konferensi.
- Menekan tombol validasi.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah tervalidasi, dan tombol validasi hilang dari halaman detail konferensi.

6.2.5 Menambah Daftar Konferensi

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan salah satu tombol konferensi.
- Menekan tombol TAMBAH.
- Mengisi *form*.
- Menekan tombol SUBMIT.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat pemberitahuan data berhasil di *entry*, dan data konferensi terdapat pada halaman konferensi.

6.2.6 Membuka Halaman Prestasi Dosen

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol prestasi dosen.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat halaman prestasi dosen yang berisi daftar prestasi dosen, *input* untuk pencarian, *filter* pencarian, dan tombol tambah.

6.2.7 Membuka Halaman Detail Prestasi Dosen

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol prestasi dosen.
- Menekan salah satu daftar prestasi dosen.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat halaman detail prestasi dosen yang berisi nama penghargaan, pelaku, kategori peserta, lembaga penyelenggara, capaian, URL, NIP, departemen, jenis

prestasi, tingkat, tanggal, tombol hapus, dan tombol validasi jika prestasi dosen belum divalidasi.

6.2.8 Menghapus Prestasi Dosen

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol prestasi dosen.
- Menekan salah satu daftar prestasi dosen.
- Menekan tombol hapus.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah terhapus, dan data sudah tidak ada di halaman prestasi dosen.

6.2.9 Memvalidasi Prestasi Dosen

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol prestasi dosen.
- Menekan salah satu daftar prestasi dosen.
- Menekan tombol validasi.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah tervalidasi, dan tombol validasi hilang dari halaman detail prestasi dosen.

6.2.10 Menambah Daftar Prestasi Dosen

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan salah satu tombol prestasi dosen.
- Menekan tombol TAMBAH.
- Mengisi *form*.

- Menekan tombol SUBMIT.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat pemberitahuan data berhasil di *entry*, dan data prestasi dosen terdapat pada halaman prestasi dosen.

6.2.11 Membuka Halaman Training

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol training.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat halaman training yang berisi daftar training, *input* untuk pencarian, *filter* pencarian, dan tombol tambah.

6.2.12 Membuka Halaman Detail Training

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol training.
- Menekan salah satu daftar prestasi training.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat halaman detail training yang berisi judul kegiatan, pelaku, tempat, tanggal selesai, jenis, departemen, tanggal mulai, gambar, tombol hapus, dan tombol validasi jika training belum divalidasi.

6.2.13 Menghapus Training

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol training.
- Menekan salah satu daftar training.

- Menekan tombol hapus.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah terhapus, dan data sudah tidak ada di halaman training.

6.2.14 Memvalidasi Training

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan tombol training.
- Menekan salah satu daftar training.
- Menekan tombol validasi.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu muncul pemberitahuan bahwa data sudah tervalidasi, dan tombol validasi hilang dari halaman detail training.

6.2.15 Menambah Daftar Training

Skenario pengujian aplikasi adalah sebagai berikut:

- Membuka /dashboard pada browser.
- Menekan salah satu tombol training.
- Menekan tombol TAMBAH.
- Mengisi *form*.
- Menekan tombol SUBMIT.
- Menekan tombol YA pada *modal* yang muncul.
- Memastikan kebutuhan-kebutuhan berikut sudah berfungsi sesuai prosedur yang disepakati yaitu melihat pemberitahuan data berhasil di *entry*, dan data training terdapat pada halaman prestasi dosen.

6.3 Evaluasi Pengujian

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem Aplikasi Monitoring Kinerja FTEIC. Hasil evaluasi pengujian dapat dilihat pada Tabel 6.1

No.	Kebutuhan	Uji Coba	Status
1.	Membuka Halaman Konferensi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Konferensi. Pengguna melihat halaman konferensi.	Berhasil
2.	Membuka Halaman Detail Konferensi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Konferensi. Pengguna menekan salah satu data konferensi. Pengguna melihat halaman detail konferensi.	Berhasil
3.	Menghapus Konferensi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Konferensi. Pengguna menekan salah satu data konferensi. Pengguna menekan tombol hapus. Pengguna menekan tombol ya pada modal.	Berhasil

		Data konferensi terhapus.	
4.	Memvalidasi Konferensi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Konferensi. Pengguna menekan salah satu data konferensi. Pengguna menekan tombol validasi. Pengguna menekan tombol ya pada modal. Data konferensi tervalidasi, tombol validasi hilang.	Berhasil
5.	Menambah Daftar Konferensi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Konferensi. Pengguna menekan tombol tambah. Pengguna mengisi data konferensi pada form. Pengguna menekan tombol submit. Pengguna menekan tombol ya pada modal. Data konferensi berhasil ditambah.	Berhasil
6.	Membuka Halaman Prestasi Dosen	Pengguna membuka halaman dashboard. Pengguna menekan	Berhasil

		tombol Prestasi Dosen. Pengguna melihat halaman prestasi dosen.	
7.	Membuka Halaman Detail Prestasi Dosen	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi Dosen. Pengguna menekan salah satu data prestasi dosen. Pengguna melihat halaman detail prestasi dosen.	Berhasil
8.	Menghapus Prestasi Dosen	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi Dosen. Pengguna menekan salah satu data prestasi dosen. Pengguna menekan tombol hapus. Pengguna menekan tombol ya pada modal. Data prestasi dosen terhapus.	Berhasil
9.	Memvalidasi Prestasi Dosen	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi Dosen. Pengguna menekan salah satu data prestasi dosen. Pengguna menekan tombol	Berhasil

		validasi. Pengguna menekan tombol ya pada modal. Data prestasi dosen tervalidasi, tombol validasi hilang.	
10.	Menambah Daftar Prestasi Dosen	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi Dosen. Pengguna menekan tombol tambah. Pengguna mengisi data prestasi dosen pada form. Pengguna menekan tombol submit. Pengguna menekan tombol ya pada modal. Data prestasi dosen berhasil ditambah.	Berhasil
11.	Membuka Halaman Training	Pengguna membuka halaman dashboard. Pengguna menekan tombol Training. Pengguna melihat halaman training.	Berhasil
12.	Membuka Halaman Detail Training	Pengguna membuka halaman dashboard. Pengguna menekan tombol Training. Pengguna menekan salah satu data sertifikasi	Berhasil

		Pengguna melihat halaman detail training.	
13.	Menghapus Training	Pengguna membuka halaman dashboard. Pengguna menekan tombol Training. Pengguna menekan salah satu data training. Pengguna menekan tombol hapus. Pengguna menekan tombol ya pada modal. Data training terhapus.	Berhasil
14.	Memvalidasi Training	Pengguna membuka halaman dashboard. Pengguna menekan tombol Training. Pengguna menekan salah satu data training. Pengguna menekan tombol validasi. Pengguna menekan tombol ya pada modal. Data training tervalidasi, tombol validasi hilang.	Berhasil
15.	Menambah Daftar Training	Pengguna membuka halaman dashboard. Pengguna menekan tombol Training. Pengguna menekan tombol tambah. Pengguna mengisi data	Berhasil

		<p>training pada form. Pegguna menekan tombol submit. Pegguna menekan tombol ya pada modal. Data training berhasil ditambah.</p>	
16.	Membuka Halaman Prestasi	<p>Pegguna membuka halaman dashboard. Pegguna menekan tombol Prestasi. Pegguna melihat halaman prestasi.</p>	Berhasil
17.	Membuka Halaman Detail Prestasi	<p>Pegguna membuka halaman dashboard. Pegguna menekan tombol Prestasi. Pegguna menekan salah satu data prestasi. Pegguna melihat halaman detail prestasi dosen.</p>	Berhasil
18.	Menghapus Prestasi	<p>Pegguna membuka halaman dashboard. Pegguna menekan tombol Prestasi. Pegguna menekan salah satu data prestasi. Pegguna menekan tombol hapus. Pegguna menekan tombol ya pada</p>	Berhasil

		modal. Data prestasi terhapus.	
19.	Memvalidasi Prestasi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi. Pengguna menekan salah satu data prestasi. Pengguna menekan tombol validasi. Pengguna menekan tombol ya pada modal. Data prestasi tervalidasi, tombol validasi hilang.	Berhasil
20.	Menambah Daftar Prestasi	Pengguna membuka halaman dashboard. Pengguna menekan tombol Prestasi. Pengguna menekan tombol tambah. Pengguna mengisi data prestasi pada form. Pengguna menekan tombol submit. Pengguna menekan tombol ya pada modal. Data prestasi berhasil ditambah.	Berhasil

Tabel 6.1 Hasil Evaluasi Pengujian Aplikasi Monitoring Kinerja FTEIC

Dengan hasil pengujian yang telah ditunjukkan, dapat disimpulkan bahwa secara keseluruhan Aplikasi Monitoring

Kinerja FTEIC telah memenuhi kriteria-kriteria yang sudah disebutkan pada bagian-bagian sebelumnya sehingga mampu melewati tahap pengujian yang telah dilakukan.

[Halaman ini sengaja dikosongkan]

BAB VII

KESIMPULAN

7.1 Kesimpulan

Kesimpulan yang didapat setelah melakukan pengembangan Aplikasi Monitoring Kinerja FTEIC adalah sebagai berikut:

- Aplikasi yang dibangun telah sesuai dengan permintaan dan dapat dengan mudah dioperasikan oleh pengguna.
- Dengan adanya Aplikasi Monitoring Kinerja FTEIC, memudahkan pengguna untuk melihat data yang ada di Fakultas FTEIC ITS.

7.2 Saran

Berikut ini adalah beberapa saran yang penulis berikan untuk arah perkembangan selanjutnya:

- Membuat dan atau merubah dokumen standar simulasi aplikasi sesuai dengan penambahan fitur-fitur baru.

[Halaman ini sengaja dikosongkan]

DAFTAR PUSTAKA

- [1] Janner Simarmata, 2010. *Rekayasa Web*. [online] Available at: https://books.google.co.id/books?id=J8JpLoPUHGAC&prints_ec=frontcover#v=onepage&q&f=false [Accessed at 5 September 2019].
- [2] Mark Lutz, 2009. *Learning Python Fourth Edition*. [online] Available at: https://cfm.ehu.es/ricardo/docs/python/Learning_Python.pdf [Accessed at 7 January 2021]
- [3] Alfin F, 2020. *Berkenalan Dengan Django Rest Framework*. [online] Available at: <https://halovina.com/berkenalan-dengan-django-rest-framework/> [Accessed at 7 January 2021]
- [4] Elza Fitriana S., 2018. *Pengembangan Aplikasi Menggunkakan Metode Prototyping: Tata Tertib Lalu Lintas dan Keselamatan Berkendara*. [online] Available at: <https://dspace.uui.ac.id/handle/123456789/12428> Accessed at 24 September 2020]

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Arif Darma Althia, lahir pada tanggal 13 Maret 1999 di Pekanbaru. Sebagai mahasiswa yang sedang menempuh studi di Departemen Informatika, Institut Teknologi Sepuluh Nopember (ITS).