



**KERJA PRAKTIK - EF234603**

**Rancang Bangun Dashboard Indeks Kinerja Emas Pada Website Kinerja UP3  
ITS**

**UP3 - ITS**

**Rektorat ITS Lantai 1 - Kampus ITS Sukolilo Surabaya,  
60117.**

**Periode: 6 Juli 2023 - 31 Desember 2023**

**Oleh:**

**Muhammad Amin**

**5025201251**

**Pembimbing Departemen**

**Dini Adni Navastara, S.Kom, M.Sc.**

**Pembimbing Lapangan**

**Dr. Eng Darlis Herumurti, S.Kom, M.Kom.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**



**KERJA PRAKTIK - EF234603**

**Rancang Bangun Dashboard Indeks Kinerja Emas Pada Website Kinerja  
UP3 ITS**

**UP3 - ITS**

**Gedung Pusat Riset Center Lantai 4 - Kampus ITS Sukolilo  
Surabaya, 60117.**

**Periode: 6 Juli 2023 - 31 Desember 2023**

**Oleh:**

**Muhammad Amin 5025201251**

**Pembimbing Departemen**

**Dini Adni Navastara, S.Kom., M.Sc.**

**Pembimbing Lapangan**

**Dr. Eng Darlis Herumurti, S.Kom., M.Kom.**

**DEPARTEMEN TEKNIK INFORMATIKA**

**Fakultas Teknologi Elektro dan Informatika Cerdas**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2023**

**LEMBAR PENGESAHAN  
KERJA PRAKTIK**

**Rancang Bangun Dashboard Indeks Kinerja Emas Pada Website Kinerja UP3 ITS**

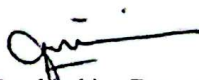
Oleh:

Muhammad Amin

5025201251

Disetujui oleh Pembimbing Kerja Praktik:

1. Dini Adni Navastara, S.Kom., M.Sc.  
NIP. 198510172015042001

  
(Pembimbing Departemen)

2. Dr. Eng Darlis Herumurti, S.Kom., M.Kom.  
NIP. 197712172003121001

  
(Pembimbing Lapangan)

**Surabaya, Agustus 2024**

## **Rancang Bangun Dashboard Indeks Kinerja Emas Pada Website Kinerja UP3 ITS**

Nama Mahasiswa : Muhammad Amin (5025201251)  
Departemen : Teknik Informatika FTEIC-ITS  
Pembimbing Departemen : Dini Adni Navastara, S.Kom., M.Sc.  
Pembimbing Lapangan : Reza Fuad Rachmadi, S.T., M.T., Ph.D.

### **ABSTRAK**

UP3 (Unit Pengembangan dan Pengendalian Program) ITS ingin membangun Dashboard Indeks Kinerja Emas (IKE)) yang memudahkan para dosen ITS untuk melihat, mencatat, serta melakukan analisis IKE.

Website Kinerja ITS merupakan portal atau tempat yang menyediakan layanan untuk melaporkan dan monitoring kinerja dosen ITS. Website ini juga menyediakan berbagai data integrasi yang dibutuhkan para dosen ITS untuk validasi kinerja serta melakukan analisis. Dashboard IKE dalam website Kinerja ITS dibuat dengan menggunakan framework Vue.js, Javascript, HTML, CSS dengan menggunakan SQL Server. Dashboard ini diharapkan dapat mempermudah pencatatan dan analisis data IKE.

**Kata kunci : UP3, Dashboard, IKE**

*[Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah Tuhan Yang Maha Esa karena atas berkat limpahan rahmat dan lindungan-Nya penulis dapat melaksanakan salah satu kewajiban sebagai mahasiswa Departemen Informatika, yakni Kerja Praktik (KP).

Penulis menyadari masih terdapat banyak kekurangan baik dalam pelaksanaan KP maupun penyusunan buku laporan ini. Namun, penulis berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Penulis mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan KP ini.

Melalui laporan ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan KP hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Orang tua penulis.
2. Bapak Dr. Eng Darlis Herumurti, S.Kom., M.Kom. selaku dosen pembimbing KP dan kepala unit UP3.
3. Bapak Ary Mazharuddin Shiddiqi, S.Kom., M.Comp.Sc., selaku koordinator KP.
4. Bapak Reza Fuad Rachmadi, S.T., M.T., Ph.D., selaku pembimbing lapangan saya di UP3.
5. Seluruh keluarga UP3 ITS yang senantiasa memberikan semangat dan dukungan kepada penulis selama melaksanakan KP
6. Teman-teman penulis yang senantiasa memberikan semangat dan dukungan kepada penulis selama melaksanakan KP.

Surabaya, Juli 2024  
Muhammad Amin

*[Halaman ini sengaja dikosongkan]*

## DAFTAR ISI

LEMBAR PENGESAHAN.....	4	
ABSTRAK.....	5	
KATA PENGANTAR.....	7	
DAFTAR ISI.....	1	
DAFTAR GAMBAR.....	3	
DAFTAR TABEL.....	5	
BAB I		
PENDAHULUAN.....	6	
1.1. Latar Belakang.....	6	
1.2. Tujuan.....	6	
1.3. Manfaat.....	6	
1.4. Rumusan Masalah.....	7	
1.5. Lokasi dan Waktu Kerja Praktik.....	7	
1.6. Metodologi Kerja Praktik.....	7	
1.7. Sistematika Laporan.....	8	
BAB II.....	10	
<b>PROFIL PERUSAHAAN.....</b>	<b>10</b>	
2.1 Sejarah Perusahaan.....	10	
2.2 Profil DPTSI.....	10	
BAB III TINJAUAN PUSTAKA.....		11
3.1. Pemrograman Web.....	11	
3.1.1. <i>HTML</i> .....	11	
3.1.2. <i>Javascript</i> .....	11	
3.1.3. <i>Microsoft SQL Server</i> .....	11	
3.1.4. <i>Vue.js</i> .....	11	
3.1.5. <i>CSS</i> .....	12	
3.1.6. <i>Bootstrap</i> .....	12	
3.1.7. <i>SQL Server</i> .....	12	
3.1.8. <i>Visual Studio Code</i> .....	12	
BAB IV		
ANALISIS DAN PERANCANGAN SISTEM.....	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.....	14	
4.3. Spesifikasi Kasus Penggunaan.....	14	
4.4. Diagram Aktivitas.....	19	
4.5. Conceptual Data Model.....	21	
4.6. Deskripsi Data.....	22	
BAB V		
IMPLEMENTASI SISTEM.....	23	
5.1. Implementasi Sistem.....	23	



5.2. Implementasi Arsitektur Sistem dan Lapisan Kontrol.....	23
5.3. Implementasi Antarmuka Pengguna.....	35
BAB VI PENGUJIAN DAN EVALUASI.....	41
6.1. Tujuan Pengujian.....	41
6.2. Kriteria Pengujian.....	41
6.3. Skenario Pengujian.....	41
6.4. Evaluasi Pengujian.....	42
BAB VII	
KESIMPULAN DAN SARAN.....	43
7.1. Kesimpulan.....	43
7.2. Saran.....	43
DAFTAR PUSTAKA.....	45
BIODATA PENULIS.....	47

## DAFTAR GAMBAR

Gambar 4.1 Diagram Use Case Dashboard IKE	13
Gambar 4.2 diagram aktivitas melihat IKE	18
Gambar 4.3 diagram aktivitas melakukan entry IKE	18
Gambar 4.4 diagram aktivitas mengedit data IKE	19
Gambar 4.5 diagram aktivitas melakukan analisis data IKE	19
Gambar 4.6 diagram aktivitas mengambil data IKE	20
Gambar 4.7 Conceptual Data Model.	20
Gambar 5.1 tampilan antarmuka dashboard IKE yang berisi grafik grafik per ike.	34
Gambar 5.2 tampilan dashboard IKE hasil filter grafik	34
Gambar 5.3 tampilan dashboard IKE hasil filter grafik	35
Gambar 5.4 tampilan excel data IKE di website Kinerja ITS	35
Gambar 5.5 kembalian json saat penarikan dengan mengakses “/api/ike”	36
Gambar 5.6 tampilan awal pada halaman “/admin/ikeentry”	36
Gambar 5.7 tampilan mode Edit	37

## DAFTAR TABEL

Tabel 4. 1 Kebutuhan Fungsional Dashboard IKE	12
Tabel 4. 2 Kebutuhan Non-Fungsional Dashboard IKE	13
Tabel 4. 3 Tabel Use Case Dashboard IKE Melihat Grafik IKE	14
Tabel 4. 4 Tabel Use Case Dashboard IKE Melakukan Entry IKE	14
Tabel 4. 5 Tabel Use Case Dashboard IKE Mengedit Data IKE	15
Tabel 4. 6 Tabel Use Case Dashboard IKE Melakukan Analisis Data IKE	16
Tabel 4. 7 Tabel Use Case Dashboard IKE Mengambil Data IKE	17
Tabel 4. 8 Deskripsi Data pada Tabel IKE	21
Tabel 6. 1 Hasil Evaluasi Pengujian Dashboard IKE	40

# **BAB I**

## **PENDAHULUAN**

### **1.1. Latar Belakang**

Institut Teknologi Sepuluh Nopember (ITS) Surabaya merupakan salah satu perguruan tinggi teknik terkemuka di Indonesia yang senantiasa berkomitmen untuk meningkatkan kualitas pendidikan, penelitian, dan pengabdian masyarakat. Dalam upaya untuk mewujudkan visi dan misi tersebut, ITS terus melakukan berbagai inovasi dalam pengelolaan dan pemantauan kinerja. Salah satu inisiatif strategis yang diluncurkan adalah pengembangan dashboard indikator kinerja Emas ITS.

Indikator Kinerja Emas ITS merupakan sistem evaluasi yang dirancang untuk mengukur dan menilai kinerja ITS dalam berbagai aspek utama, termasuk akademik, penelitian, pengabdian masyarakat, dan pengelolaan sumber daya. Sistem ini berfungsi sebagai alat bantu dalam pengambilan keputusan dan perencanaan strategis, dengan tujuan untuk meningkatkan kualitas dan efektivitas program-program yang dilaksanakan oleh ITS.

Namun, untuk memastikan bahwa indikator kinerja dapat diakses, dipantau, dan dianalisis secara efisien, diperlukan sebuah alat yang dapat menyajikan data secara real-time dan interaktif. Oleh karena itu, pembuatan dashboard indikator kinerja Emas ITS menjadi sangat penting. Dashboard ini diharapkan akan menjadi solusi yang efektif dalam menyediakan informasi yang terintegrasi dan terperinci mengenai kinerja ITS, memudahkan pengawasan, dan memberikan gambaran yang jelas tentang pencapaian dan area yang memerlukan perbaikan.

### **1.2. Tujuan**

Tujuan KP ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban 2 SKS. Selain itu juga untuk membantu dosen-dosen ITS untuk melakukan analisis terkait IKE.

Tujuan dari pembangunan dashboard tersebut antara lain:

1. Meningkatkan aksesibilitas data.
2. Menunjang proses pencatatan dan analisis IKE.

### **1.3. Manfaat**

Manfaat yang dapat diperoleh dengan adanya dashboard IKE antara lain adalah:

1. Mempermudah administrator untuk mencatat Indeks Kinerja Emas ITS.
2. Mempermudah analisis terkait IKE.

#### 1.4. Rumusan Masalah

Berikut ini rumusan masalah pada KP pembuatan Aplikasi MyITS License Management:

1. Bagaimana solusi efektif dalam pembuatan dashboard IKE?
2. Bagaimana cara pembuatan dashboard IKE agar menghasilkan dashboard yang dapat menunjang pengambilan keputusan.

#### 1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut :

Lokasi : UP3 - ITS  
Alamat : Rektorat ITS Lantai 1 – Kampus ITS Sukolilo  
Surabaya, 60117.  
Waktu: 6 Juli 2023 – 31 Desember 2023

#### 1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

##### 1. Perumusan Masalah

Untuk mengetahui permasalahan apa yang harus diselesaikan, diberikan penjelasan mengenai alasan mengapa dashboard ini dibutuhkan. Dijelaskan pula secara rinci mengenai bagaimana alur sistem itu akan berjalan. Penjelasan mengenai hal ini dijelaskan oleh pembimbing lapangan KP. Dari penjelasannya dihasilkan catatan-catatan penting mengenai gambaran dashboard yang akan diintegrasikan ke dalam website kinerja ITS ini akan dibuat. Dengan begitu dapat diputuskan untuk membuat dashboard berbasis web dengan menggunakan bahasa pemrograman Vue.js dan Javascript serta menggunakan database SQL Server. Hal ini dikarenakan bahasa pemrograman web jenis ini telah digunakan pada website Kinerja ITS untuk memudahkan integrasi..

##### 2. Studi Literatur

Setelah ditentukan database, bahasa pemrograman, serta tools tambahan yang akan digunakan, dilakukan studi literatur mengenai cara implementasinya dalam membangun sistem sesuai yang dibutuhkan. Pada tahap ini dilakukan proses pencarian, pembelajaran, pengumpulan dan pemahaman informasi serta literatur yang berkaitan untuk membantu dalam implementasi aplikasi ini. Informasi bisa didapat dari internet untuk istilah-istilah umum yang digunakan dalam mengimplementasikan suatu sistem informasi.

##### 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 dan non-fungsional secara garis besar. Kemudian dilanjutkan dengan memperjelas dan menspesifikkan kebutuhan-kebutuhan tersebut. Maka dibuat sebuah diagram kasus penggunaan yang mewakili skenario- skenario untuk penggunaan dashboard IKE dalam website Kinerja ITS.

##### 4. Implementasi Sistem

Implementasi sistem didasarkan pada perancangan dan analisis sebelumnya. Kasus penggunaan dan penentuan tools juga turut mendasari pengimplementasian sistem ini. Pada tahap ini setidaknya ada dua pekerjaan utama yang dilakukan, yakni

desain web pada tampilan atau front-end dan juga desain fungsi-fungsi yang bekerja dalam sistem atau dikenal sebagai back-end. Pengerjaan dilakukan dengan progres setiap hari, dengan setiap harinya menargetkan perkembangan dari hari sebelumnya. Progres penyelesaian dashboard terus dipantau oleh UP3 sebagai pelopor dashboard Indeks Kinerja Emas ITS.

## **5. Pengujian dan Evaluasi**

Pengujian dilakukan dengan menguji fitur-fitur yang telah dibuat. Kesesuaian sistem dengan kebutuhan akan menentukan keberhasilan dalam pengujian. Hal ini akan menghasilkan hasil evaluasi apakah sistem sudah sesuai dengan tujuan dan kebutuhan atau belum.

### **1.7. Sistematika Laporan**

Laporan KP ini terdiri dari tujuh bab dengan rincian sebagai berikut:

#### **1. Bab I Pendahuluan**

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan serta sistematika pengerjaan KP dan juga penulisan laporan KP.

#### **2. Bab II Profil Perusahaan**

Pada bab ini, dijelaskan secara rinci tentang profil perusahaan tempat melaksanakan KP, yakni UP3 (Unit Pengelolaan dan Pengendalian Program).

#### **3. Bab III Tinjauan Pustaka**

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian KP.

#### **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.

#### **5. Bab V Implementasi Sistem**

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

#### **6. Bab VI Pengujian dan Evaluasi**

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

#### **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 PERUSAHAAN**

#### **2.1 Sejarah Perusahaan**

Unit Pengelolaan dan Pengendalian Program (UP3) ITS di Institut Teknologi Sepuluh Nopember (ITS) Surabaya didirikan sebagai respons terhadap kebutuhan untuk meningkatkan efisiensi dan efektivitas dalam pengelolaan berbagai program di kampus. Sejak awal berdirinya ITS, unit ini berperan penting dalam memastikan bahwa kegiatan akademik dan non-akademik yang dilaksanakan dapat berjalan sesuai dengan rencana strategis dan tujuan yang telah ditetapkan.

Pada tahap awal, fokus utama UP3 ITS adalah pada pengelolaan anggaran dan pelaksanaan program. Namun, seiring dengan pertumbuhan jumlah program dan kompleksitasnya, fungsi UP3 ITS berkembang untuk mencakup pemantauan kinerja, evaluasi program, dan koordinasi antar unit. Perkembangan ini mencerminkan kebutuhan untuk pengawasan yang lebih sistematis dan terstruktur dalam menjalankan berbagai inisiatif kampus.

Dalam beberapa tahun terakhir, UP3 ITS telah mengintegrasikan teknologi informasi untuk meningkatkan efisiensi operasional. Penerapan sistem informasi manajemen dan alat bantu analisis data telah memungkinkan pemantauan yang lebih baik dan pengambilan keputusan berbasis data. Teknologi ini mempermudah akses informasi dan meningkatkan transparansi dalam pengelolaan program.

Selain itu, UP3 ITS telah menjalani berbagai inisiatif untuk meningkatkan kapasitas dan kualitas pengelolaannya, termasuk pelatihan bagi staf dan penyempurnaan proses internal. Unit ini kini memainkan peran strategis dalam perencanaan dan pengambilan keputusan tingkat tinggi di ITS, memastikan bahwa semua program berjalan sinergis dan mencapai hasil yang optimal. Sejarah UP3 ITS mencerminkan perjalanan ITS dalam beradaptasi dan berinovasi untuk memenuhi kebutuhan pengelolaan yang berkembang, serta mendukung pencapaian tujuan akademik dan strategis kampus.

#### **2.2 Profil DPTSI**

Unit Pengelolaan dan Pengendalian Program (UP3) bertugas untuk mengelola, memonitor, serta menganalisis seluruh program kerja yang ada di ITS. Terkait peran, UP3 berperan untuk mendukung administrasi dan pemantauan kinerja dosen di ITS, serta manajerial di lingkungan ITS dalam rangka membantu ITS mencapai visi misinya .



## **BAB III**

### **TINJAUAN PUSTAKA**

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses pengerjaan dan pengembangan aplikasi.

#### **3.1. Pemrograman Web**

Aplikasi berbasis web merupakan sebuah program atau perangkat lunak yang di akses melalui internet dengan menggunakan web browser [2]. Sedangkan pemrograman web adalah proses pembuatan aplikasi berbasis web [3].

##### **3.1.1. *HTML***

HyperText Markup Language (HTML) adalah sebuah bahasa markah yang digunakan untuk membuat sebuah halaman web. HTML merupakan kumpulan script yang bisa kita gunakan untuk membuat halaman web. Dari HTML ini, kita bisa menampilkan data baik berupa teks maupun gambar di situs web yang kita buat [4].

##### **3.1.2. *Javascript***

JavaScript adalah bahasa scripting yang digunakan pada Web pages untuk meningkatkan design, validate forms, detect browsers, create cookies, GUI dan sebagainya. Javascript tidak dapat berdiri sendiri dan harus didasari oleh HTML atau PHP [4].

##### **3.1.3. *Microsoft SQL Server***

Microsoft SQL Server adalah sebuah sistem manajemen basis data relasional (RDBMS) yang dikembangkan oleh Microsoft. RDBMS ini dirancang untuk menyimpan dan mengelola data dengan cara yang terstruktur dan efisien. SQL Server biasanya digunakan dalam berbagai aplikasi dan skenario, mulai dari aplikasi bisnis kecil hingga sistem perusahaan yang besar [?].

##### **3.1.4. *Vue.js***

Vue.js adalah sebuah framework JavaScript yang bersifat progresif untuk membangun antarmuka pengguna (UI) yang interaktif dan single-page applications (SPA). Dikembangkan oleh Evan You, Vue.js memiliki desain yang ringan dan mudah dipelajari, namun kuat dalam mengintegrasikan dengan proyek-proyek yang ada. Vue.js menggunakan pendekatan berbasis komponen, di mana aplikasi dibangun dari komponen-komponen yang dapat digunakan kembali. Hal ini memungkinkan pengembang untuk memisahkan logika aplikasi dari UI dan memfasilitasi pengembangan yang lebih terstruktur dan terkelola. Vue.js juga mendukung dua arah data binding, yang memungkinkan perubahan di model data terjadi secara langsung mempengaruhi tampilan UI, serta memiliki ekosistem plugin yang aktif dan dukungan yang luas dari komunitas pengembang[6].

### **3.1.5. CSS**

Cascading Style Sheet (CSS) merupakan aturan untuk mengendalikan beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. CSS dapat mengendalikan ukuran gambar, warna body teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri/kanan/atas/bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda [4].

### **3.1.6. Bootstrap**

Bootstrap adalah framework CSS yang dikhususkan untuk pengembangan front-end website. Bootstrap adalah salah satu framework yang memungkinkan developer dapat mengembangkan website dengan mudah dan cepat. Developer hanya perlu memanggil class tertentu untuk membuat tombol, panel, tabel, pesan peringatan, dan lain sebagainya [11].

### **3.1.7. SQL Server**

SQL Server adalah sebuah database management system (manajemen basis data) open source menggunakan perintah dasar SQL (Structured Query Language). SQL sendiri merupakan suatu bahasa yang dipakai di dalam pengambilan data pada relational database atau database yang terstruktur. Jadi SQL Server adalah database management system yang menggunakan bahasa SQL sebagai bahasa penghubung antara perangkat lunak aplikasi dengan database server [5].

### **3.1.8. Visual Studio Code**

Visual Studio Code adalah source code editor yang dikembangkan Microsoft yang bisa digunakan pada berbagai platform (Linux, Mac OS dan Microsoft). Visual Studio Code juga dapat digunakan sebagai editor source code berbagai bahasa pemrograman dari PHP, Java, Python, Javascript, Node.js, C/C++, Go, C# sampai editor untuk script web seperti HTML dan CSS. Visual Studio Code juga telah terintegrasi ke GitHub [

## **BAB IV ANALISIS DAN PERANCANGAN SISTEM**

### **4.1. Analisis Sistem**

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

#### **4.1.1. Definisi Umum Aplikasi**

Secara umum, Dashboard IKE ini merupakan dashboard berbasis web yang digunakan untuk melakukan analisis data IKE. Dashboard ini dapat memudahkan tendik dalam melakukan analisis data IKE yang bertujuan untuk membuat rencana strategis ke depannya. Adapun pengguna aplikasi ini didefinisikan sebagai berikut:

- Admin dari UP3 yang dapat melakukan input dan edit data IKE.
- Dosen dan Tendik yang dapat mengakses dashboard dan mengambil data IKE.

#### **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 itu bekerja. Kebutuhan fungsional dari Dashboard IKE pada website Kinerja ITS dijelaskan pada Tabel 4.1.

*Tabel 4. 1 Kebutuhan Fungsional Dashboard IKE*

<b>Kode Kebutuhan</b>	<b>Deskripsi Kebutuhan</b>
F-001	Melihat Grafik IKE
F-002	Melakukan Entry IKE
F-003	Mengedit Data IKE
F-004	Melakukan Analisis Data IKE
F-005	Mengambil Data IKE

##### **4.1.2.2. Kebutuhan Non-Fungsional**

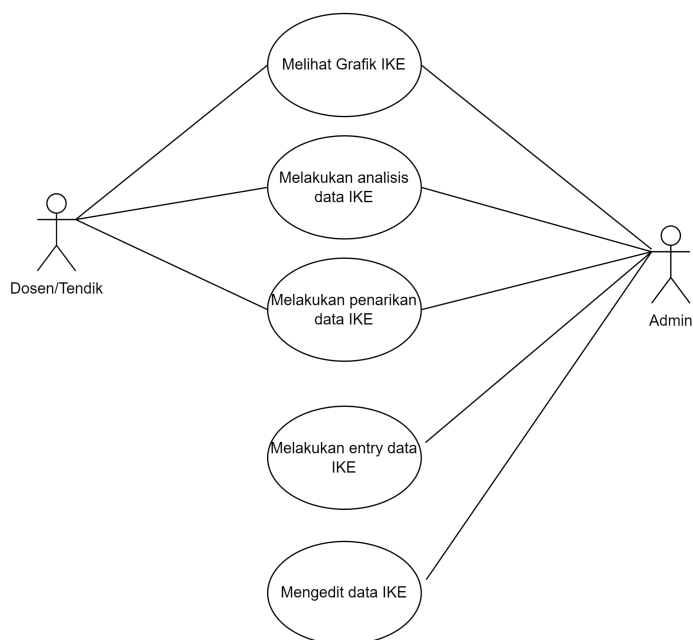
Kebutuhan non-fungsional adalah kebutuhan pengguna untuk mendefinisikan bagaimana batasan dan karakteristik dari sebuah sistem yang dibangun. Kebutuhan non-fungsional dari Aplikasi MyITS License Management terdapat pada Tabel 4.2.

Tabel 4. 2 Kebutuhan Non-Fungsional Dashboard IKE

Kode Kebutuhan	Deskripsi Kebutuhan
NF-001	Sistem dapat diakses oleh pengguna
NF-002	Sistem memudahkan pengguna untuk melihat grafik yang diinginkan
NF-003	Sistem memiliki tampilan antarmuka yang mudah dipahami

#### 4.2. Diagram Kasus Penggunaan

Pembahasan dengan pembimbing lapangan tentang fitur-fitur yang perlu ada dalam Dashboard IKE pada website Kinerja ITS menghasilkan beberapa fitur yang dijadikan 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 Dashboard IKE

#### 4.3. Spesifikasi Kasus Penggunaan

##### 4.3.1. Melihat Grafik IKE

Tabel 4.3 berikut merupakan tabel use case dari Aplikasi MyITS License Management Melakukan Login.

Tabel 4. 3 Tabel Use Case Dashboard IKE Melihat Grafik IKE

Nama	Melihat Grafik IKE
Kode	UC001
Deskripsi	Aktor dapat mengakses grafik IKE
Tipe	Fungsional
Pemicu	Aktor telah login pada website kinerja ITS dan mengetikkan link “/dashboard/ike”
Aktor	dosen, tendik, admin
Kondisi Awal	Berbagai Grafik IKE diperlihatkan
Kondisi Akhir	Aktor dapat melihat grafik IKE untuk kebutuhan analisis
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor mengakses “/dashboard/ike”</li> <li>2. Sistem menampilkan halaman utama</li> </ol>
Alur Kejadian Alternatif	<ol style="list-style-type: none"> <li>1. Aktor mengakses “/dashboard/ike”</li> <li>2. Sistem menampilkan halaman utama</li> <li>3. Aktor memilih filter grafik yang diinginkan</li> <li>4. Sistem menampilkan list grafik IKE yang telah terfilter</li> </ol>
Pengecualian	-

#### 4.3.2. Melakukan Entry IKE

Tabel 4.4 berikut merupakan tabel use case dari Dashboard IKE untuk melakukan Entry IKE

Tabel 4. 4 Tabel Use Case Dashboard IKE Melakukan Entry IKE

Nama	Melakukan Entry IKE
Kode	UC002
Deskripsi	Aktor dapat melakukan entry data IKE
Tipe	Fungsional
Pemicu	Aktor mengakses link “/ikeentry”

Aktor	Admin
Kondisi Awal	Aktor ingin melakukan entry data IKE
Kondisi Akhir	Aktor berhasil melakukan entry data IKE
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor mengakses link “admin/ikeentry”</li> <li>2. Sistem akan menampilkan form entry IKE</li> <li>3. Aktor mengisi form dan submit</li> <li>4. Sistem akan menyimpan data IKE</li> </ol>
Alur Kejadian Alternatif	
Pengecualian	-

#### 4.3.3. Mengedit Data IKE

Tabel 4.5 berikut merupakan tabel use case dari Dashboard IKE untuk mengedit Data IKE

*Tabel 4. 5 Tabel Use Case Dashboard IKE Mengedit Data IKE*

Nama	Mengedit data IKE
Kode	UC003
Deskripsi	Admin dapat mengedit data IKE
Tipe	Fungsional
Pemicu	Aktor mengklik tombol “edit” saat berada di halaman “/admin/ikeentry”
Aktor	Admin
Kondisi Awal	Aktor ingin mengedit data IKE
Kondisi Akhir	Aktor berhasil mengedit data IKE
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor menekan tombol mode edit</li> <li>2. Sistem menampilkan halaman edit IKE</li> <li>3. Aktor mengedit data IKE dan menekan tombol save</li> <li>4. Sistem menyimpan hasil edit data IKE</li> </ol>

Alur Kejadian Alternatif	-
Pengecualian	-

#### 4.3.4. Melakukan Analisis Data IKE

Tabel 4.6 berikut merupakan tabel use case dari Dashboard IKE untuk melakukan analisis Data IKE

*Tabel 4. 6 Tabel Use Case Dashboard IKE Melakukan Analisis Data IKE*

Nama	Melakukan Analisis Data IKE
Kode	UC004
Deskripsi	Aktor dapat melakukan analisis data IKE dengan menerapkan berbagai fungsi yang dibutuhkan
Tipe	Fungsional
Pemicu	Aktor mengakses link “/integrasi/ike”
Aktor	Dosen, Tendik, Admin
Kondisi Awal	Aktor ingin melakukan analisis data IKE
Kondisi Akhir	Aktor berhasil melakukan analisis data IKE
Alur Kejadian Secara Normal	<ol style="list-style-type: none"> <li>1. Aktor mengakses link “/integrasi/ike”</li> <li>2. Sistem menampilkan excel embedded IKE</li> <li>3. Aktor mengaktifkan fungsi atau filter yang tersedia</li> <li>4. Sistem menampilkan data IKE berdasarkan fungsi atau filter yang diaktifkan</li> </ol>
Alur Kejadian Alternatif	
Pengecualian	-

#### 4.3.5. Mengambil Data IKE

Tabel 4.7 berikut merupakan tabel use case dari Dashboard IKE Mengambil data IKE.

*Tabel 4. 7 Tabel Use Case Dashboard IKE Mengambil Data IKE*

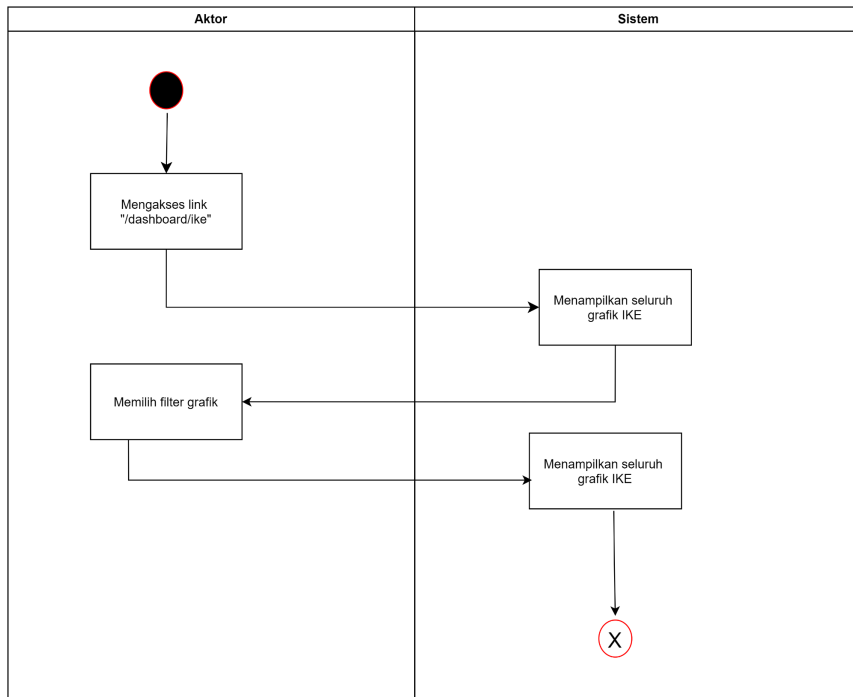
Nama	Mengambil Data IKE
Kode	UC005
Deskripsi	Aktor dapat melakukan penarikan data IKE
Tipe	Fungsional
Pemicu	Aktor mengakses link “/api/ike” pada website Kinerja
Aktor	Dosen, Tendik, dan Admin
Kondisi Awal	Aktor belum melakukan penarikan data IKE
Kondisi Akhir	Aktor mendapatkan data IKE
Alur Kejadian Secara Normal	<ol style="list-style-type: none"><li>1. Aktor mengakses link “/api/IKE”</li><li>2. Sistem mengembalikan data IKE</li></ol>
Alur Kejadian Alternatif	
Pengecualian	-



#### 4.4. Diagram Aktivitas

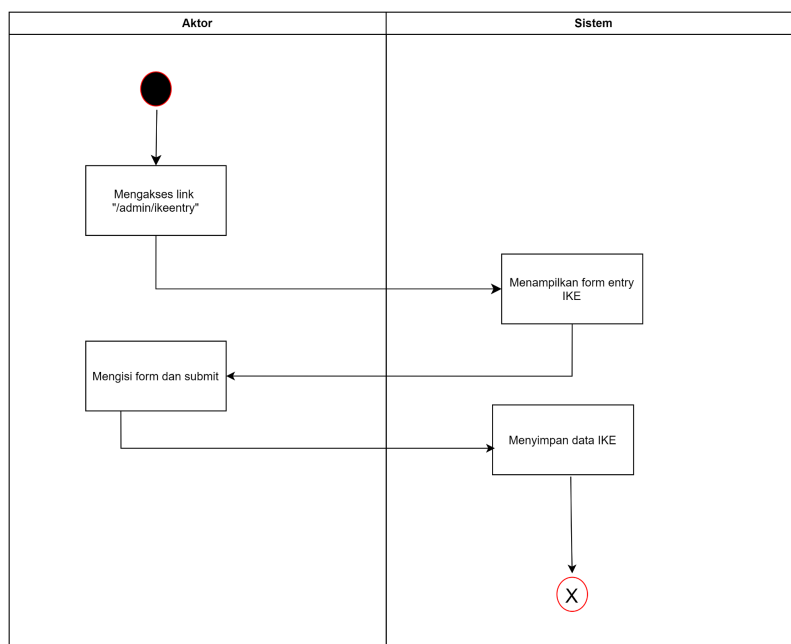
##### 4.4.1. Melihat Grafik IKE

Gambar 4.2 berikut merupakan diagram aktivitas dari Dashboard IKE untuk melihat IKE



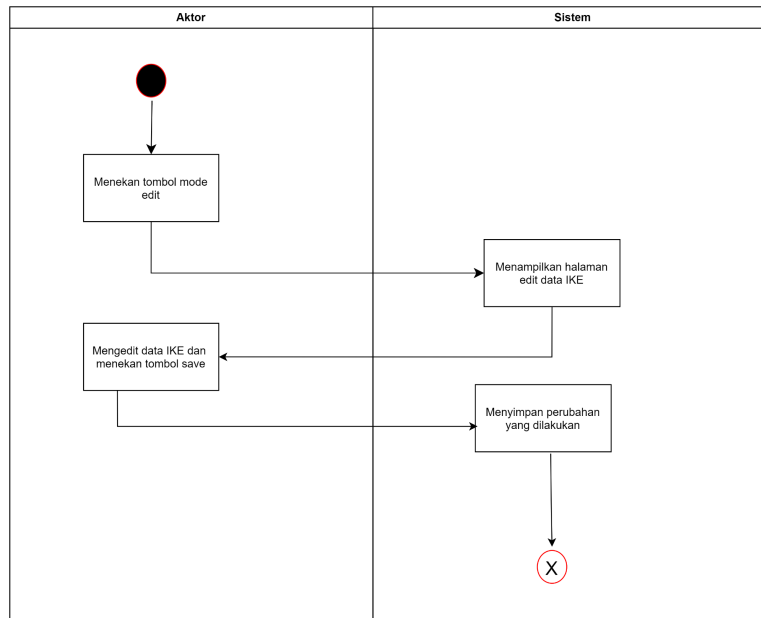
##### 4.4.2. Melakukan Entry IKE

Gambar 4.3 berikut merupakan diagram aktivitas dari melakukan entry IKE



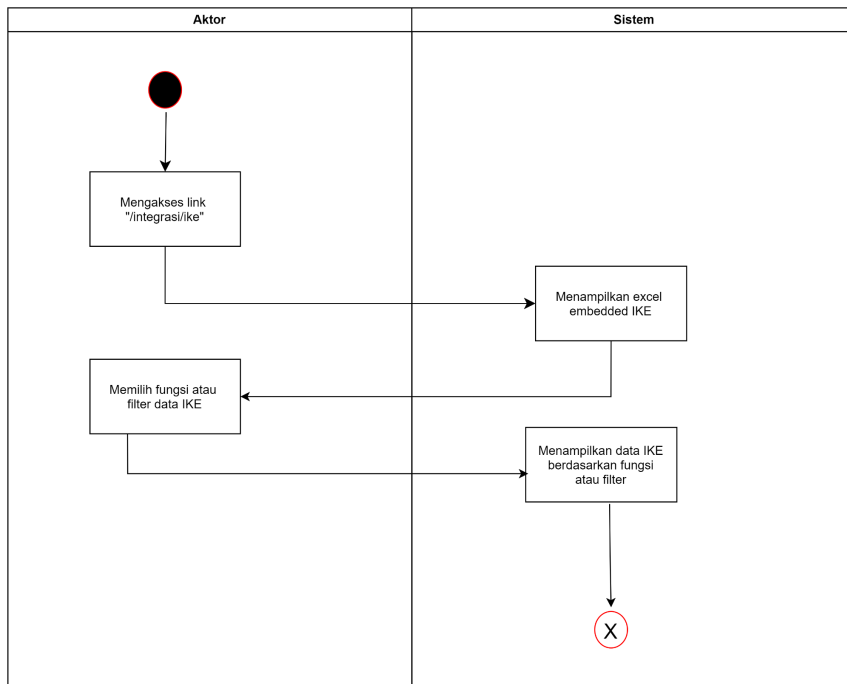
#### 4.4.3. Mengedit Data IKE

Gambar 4.4 berikut merupakan diagram aktivitas dari Dashboard IKE untuk mengedit data IKE



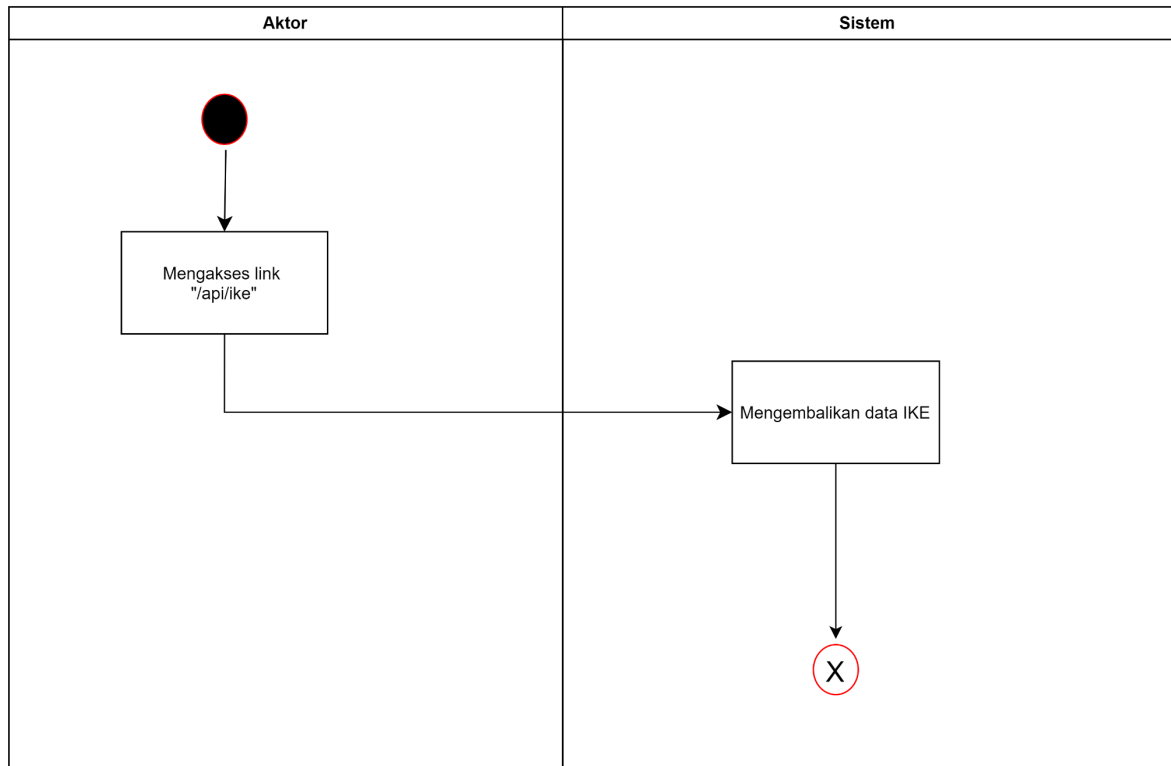
#### 4.4.4. Melakukan Analisis Data IKE

Gambar 4.5 berikut merupakan diagram aktivitas dari Dashboard IKE untuk melakukan analisis data IKE



#### 4.4.5. Mengambil Data IKE

Gambar 4.6 berikut merupakan diagram aktivitas dari Dashboard IKE untuk mengambil data IKE



#### 4.5. Conceptual Data Model

Gambar 4.7 berikut adalah Conceptual Data Model dari Dashboard IKE pada website Kinerja.

IKE		
kode	Varchar(%)	M
jenis	Varchar(%)	M
indikator	Varchar(%)	M
bobot	Float(%)	M
target	Float(%)	M
capaian	Float(%)	M
persen	Float(%)	M
persen_max	Float(%)	M
persen_terbobot	Float(%)	M
persen_terbobot_	Float(%)	M
tahun	Float(%)	M



#### 4.6. Deskripsi Data

Berikut adalah deskripsi data pada Aplikasi MyITS License Management meliputi nama tabel, nama atribut, dan tipe data. Tabel 4.21 adalah deskripsi data pada tabel tipe lisensi.

*Tabel 4. 8 Deskripsi Data pada Tabel IKE*

IKE		
No	Nama Atribut	Tipe Data
1.	kode	Variable characters
2.	jenis	Variable characters
3.	indikator	Variable characters
4.	bobot	float
5.	target	float
6.	capaian	float
7.	persen	float
8.	persen_max	float
9.	persen_terbobot	float
10.	persen_terbobot_max	float
11.	tahun	integer

## BAB V IMPLEMENTASI SISTEM

Bab ini membahas tentang implementasi dari perancangan sistem dan pengaplikasian sistem dalam bentuk situs web.

### 5.1. Implementasi Sistem

Sistem yang dibuat merupakan sebuah dashboard IKE yang di integrasikan ke dalam website Kinerja ITS yang dikelola oleh UP3. Sistem dapat melihat grafik IKE, membuka sebuah excel embedding untuk dilakukan analisis data IKE lebih lanjut, melakukan entry dan edit data IKE, serta melakukan penarikan data IKE.

Aplikasi ini dibuat dengan menggunakan Framework Vue.js dengan DBMS yang digunakan adalah SQL Server. Bahasa pemrograman web yang digunakan adalah Javascript, HTML, dan CSS. Sedangkan library yang digunakan adalah mssql.

Semua fungsi ini terbagi dalam banyak fitur seperti yang telah dipaparkan pada bagian sebelumnya. User dalam hal ini meliputi Dosen, Tendik serta pegawai UP3 dapat mengakses aplikasi ini melalui login dengan kredensial website Kinerja ITS.

### 5.2. Implementasi Arsitektur Sistem dan Lapisan Kontrol

Terdapat implementasi lapisan kontrol yang berisi logika yang digunakan dashboard IKE.

#### 5.2.1. Lapisan Kontrol Halaman IKE Entry dan Edit

```
1. import vSelect from 'vue-select'
2. import 'vue-select/dist/vue-select.css'
3.
4. import axios from 'axios'
5. import jexcel from 'jexcel'
6. import 'jexcel/dist/jexcel.css'
7. import { host } from '../././config/var.js'
8.
9. export default {
10.   components: {
11.     vSelect
12.   },
13.   name: 'App',
14.   props: ['user'],
15.   data() {
16.     return {
17.       isInsert: true,
18.       isEdit: false,
19.       ike: [],
20.       jenis: [],
21.       tahun: [],
22.       items: {},
23.       itemList: [],
24.       tahunList: [],
25.       jexcel_instance: null,
26.       jexcel_spreadsheet: null,
```

```

27.     tahun_choice: null,
28.     changedRow: [],
29.     form: {
30.         kode: null,
31.         jenis: null,
32.         indikator: null,
33.         target: 0,
34.         capaian: 0,
35.         persen: 0,
36.         tahun: new Date().getFullYear()
37.     },
38.     options: {
39.         data: [{ 'kode': '', 'jenis': '', 'indikator': '', 'target': 0, 'capaian': 0,
40. 'persen': 0, 'tahun': 0, 'periode': '' }],
41.         colAlignments: ['center', 'center', 'center', 'center', 'center', 'center',
42. 'center', 'center', 'center', 'center', 'center', 'center'],
43.         columns: [
44.             {type: 'text', title: 'kode', readOnly: true, width: '75px'},
45.             {type: 'text', title: 'jenis', readOnly: true, width: '125px'},
46.             {type: 'text', title: 'indikator', readOnly: false, width: '350px'},
47.             {type: 'numeric', title: 'bobot', readOnly: false, width: '75px'},
48.             {type: 'numeric', title: 'target', readOnly: false, width: '75px'},
49.             {type: 'numeric', title: 'capaian', readOnly: false, width: '75px'},
50.             {type: 'hidden', title: 'persen', readOnly: true, width: '75px'},
51.             {type: 'hidden', title: 'persen_max', readOnly: true, width: '75px'},
52.             {type: 'hidden', title: 'persen_terbobot', readOnly: true, width: '75px'},
53.             {type: 'hidden', title: 'persen_terbobot_max', readOnly: true, width: '75px'},
54.             {type: 'numeric', title: 'tahun', readOnly: true, width: '75px'}
55.             // {type: 'hidden', title: 'periode', readOnly: true, width: '75px'}
56.         ],
57.         editable: true,
58.         allowInsertRow: true,
59.         allowInsertColumn: false,
60.         allowDeleteRow: false,
61.         allowDeleteColumn: false,
62.         allowRenameColumn: false,
63.         allowComments: false,
64.         loadingSpin: true,
65.         wordWrap: true,
66.         tableOverflow: true,
67.         tableWidth: '100%',
68.         tableHeight: '300px',
69.         search: true,
70.         onchange: this.changed,
71.         onselection: this.selected
72.     }
73. },
74. mounted: function() {
75.     if (this.user.hak_akses !== 'up3' && this.user.hak_akses !== 'admin' &&
76. this.user.hak_akses !== 'validator') { this.$router.go(-1) }

```

```

75.   this.init()
76. },
77. computed: {
78.   calculatePersen() {
79.     console.log('persen calculated')
80.     if (this.form.target !== 0) {
81.       return ((this.form.capaian / this.form.target) * 100).toFixed(2)
82.     }
83.     return 0
84.   }
85. },
86. methods: {
87.   init_jexcel() {
88.     this.changedRow = []
89.     try {
90.       this.jexcel_instance.destroy()
91.     } catch (e) {
92.     }
93.     this.$nextTick(() => {
94.       axios.get(host + 'api/ike').then(response => {
95.         this.options.data = response.data
96.         try {
97.           this.jexcel_spreadsheet = document.getElementById('spreadsheet')
98.           this.jexcel_instance = jexcel(this.jexcel_spreadsheet, this.options)
99.         } catch (e) {
100.           }
101.         })
102.       })
103.     },
104.     init_form() {
105.       this.form = {}
106.       this.form.choice = null
107.       this.form.kode = null
108.       this.form.jenis = null
109.       this.form.indikator = null
110.       this.form.target = null
111.       this.form.capaian = null
112.       this.form.persen = 0
113.       this.form.tahun = null
114.     },
115.     init_list() {
116.       this.ike.forEach(kode => {
117.         if (this.items[kode].tahun.length !== this.tahun.length) {
118.           this.itemList.push(kode + ' | ' + this.items[kode].jenis + ' | ' +
this.items[kode].indikator)
119.         } else {
120.           this.itemList.push('(Data Lengkap) ' + kode + ' | ' + this.items[kode].jenis +
' | ' + this.items[kode].indikator)
121.         }
122.       })

```

```

123.     this.itemList = this.unique(this.itemList)
124. },
125. init() {
126.     this.ike = []
127.     this.jenis = []
128.     this.tahun = []
129.     this.itemList = []
130.     this.items = {}
131.     axios.get(host + 'api/ike').then(response => {
132.         response.data.forEach(item => {
133.             console.log(item)
134.             this.ike.push(item.kode)
135.             this.jenis.push(item.jenis)
136.             this.tahun.push(parseInt(item.tahun))
137.             // this.itemList.push(item.kode + ' | ' + item.jenis + ' | ' + item.indikator)
138.             if (this.items.hasOwnProperty(item.kode)) {
139.                 this.items[item.kode].capaian.push(parseFloat(item.capaian))
140.                 this.items[item.kode].target.push(parseFloat(item.target))
141.                 this.items[item.kode].tahun.push(parseInt(item.tahun))
142.                 this.items[item.kode].indikator = item.indikator
143.             } else {
144.                 this.items[item.kode] = {
145.                     kode: item.kode,
146.                     jenis: item.jenis,
147.                     indikator: item.indikator,
148.                     capaian: [parseFloat(item.capaian)],
149.                     target: [parseFloat(item.target)],
150.                     tahun: [parseInt(item.tahun)]
151.                 }
152.             }
153.         })
154.         this.ike = this.unique(this.ike)
155.         this.jenis = this.unique(this.jenis)
156.         this.tahun.push(new Date().getFullYear())
157.         this.tahun = this.unique(this.tahun.sort())
158.         this.init_list()
159.         this.init_form()
160.         this.init_jexcel()
161.     })
162. },
163. unique(arr) {
164.     return [...new Set(arr)]
165. },
166. insertMode() {
167.     this.isInsert = true
168.     this.isHome = false
169.     this.isEdit = false
170.     this.init_list()
171. },
172. editMode() {

```



```

173.     var prev = this.isEdit
174.     this.isEdit = true
175.     this.isHome = false
176.     this.isInsert = false
177.     if (!prev) this.init_jexcel()
178. },
179. changed(instance, cell, x, y, value) {
180.     x = parseInt(x)
181.     y = parseInt(y)
182.     // var capaian = parseFloat(this.jexcel_instance.getValueFromCoords(4,
183.     y)).toFixed(3)
184.     // var target = parseFloat(this.jexcel_instance.getValueFromCoords(3,
185.     y)).toFixed(3)
186.     // var persen = ((capaian / target) * 100).toFixed(3)
187.     // this.jexcel_instance.setValueFromCoords(5, y, persen)
188.     // this.options.data[y][5] = persen
189.     this.changedRow.push(y)
190.     // console.log('change : ' + x + ', ' + y)
191. },
192. selected(instance, x1, y1, x2, y2, origin) {
193.     // console.log(x1 + ', ' + y1 + ' | ' + x2 + ', ' + y2)
194. },
195. formSelectKode(newKode) {
196.     // this.form.choice = newKode
197.     this.form.kode = null
198.     this.form.jenis = null
199.     this.form.indikator = null
200.     var pattern = /IKE\d+/
201.     try {
202.         this.form.kode = newKode.match(pattern)[0]
203.     } catch (e) {
204.         return null
205.     }
206.     this.form.jenis = this.items[this.form.kode].jenis
207.     this.form.indikator = this.items[this.form.kode].indikator
208.     // console.log(this.tahun)
209.     try {
210.         this.tahunList = this.tahun.filter(tahun =>
211.         !this.items[this.form.kode].tahun.includes(tahun))
212.     } catch (e) {
213.         this.tahunList = this.tahun
214.         console.log(this.tahunLweist)
215.     }
216.     if (this.tahunList.length > 0) {
217.         this.form.tahun = this.tahunList[0]
218.     } else {
219.         this.form.tahun = 'Data ' + this.form.kode + ' sudah lengkap'
220.     }
221. },
222. formSelectTahun(newTahun) {
223.     this.form.tahun = newTahun

```

```

221.     },
222.     newIke() {
223.         this.form.tahun = this.tahun
224.     },
225.     // updatePersen() {
226.     //     if (this.form.target !== 0 && this.form.choice !== null) {
227.     //         this.form.persen = ((this.form.capaian / this.form.target) * 100).toFixed(3)
228.     //     } else {
229.     //         this.form.persen = 0
230.     //     }
231.     // },
232.     postIke() {
233.         if (this.form.kode === null || this.form.jenis === null || this.form.indikator ===
null || this.form.target === null || this.form.capaian === null || this.form.persen ===
null || this.form.tahun === null) {
234.             console.log('Please fill in all required fields.')
235.         } else {
236.             var bobot = parseFloat(this.form.bobot)
237.             var target = parseFloat(this.form.target)
238.             var capaian = parseFloat(this.form.capaian)
239.             var persen = capaian / target
240.             var persenMax = (5 / 4 < persen) ? 5 / 4 : persen
241.             var dataToPost = {
242.                 kode: this.form.kode,
243.                 jenis: this.form.jenis,
244.                 indikator: this.form.indikator,
245.                 bobot: bobot,
246.                 target: target,
247.                 capaian: capaian,
248.                 persen: capaian / target,
249.                 persenMax: persenMax,
250.                 persenTerbobot: bobot * persen,
251.                 persenTerbobotMax: bobot * persenMax,
252.                 tahun: this.form.tahun
253.             }
254.             axios({
255.                 method: 'post',
256.                 url: host + 'api/ike',
257.                 data: dataToPost
258.             })
259.             // this.getList()
260.             this.init_form()
261.         }
262.     },
263.     updateIke() {
264.         this.changedRow = this.unique(this.changedRow)
265.         this.changedRow.forEach(row => {
266.             var capaian = parseFloat(this.jexcel_instance.getValueFromCoords(5,
row)).toFixed(3)
267.             var bobot = parseFloat(this.jexcel_instance.getValueFromCoords(3,
row)).toFixed(3)

```

```

268.         var target = parseFloat(this.jexcel_instance.getValueFromCoords(4,
row)).toFixed(3)
269.         var persen = ((capaian / target) * 100).toFixed(3)
270.         var persenMax = 0.0
271.         if (persen < 5 / 4) {
272.             persenMax = persen
273.         } else {
274.             persenMax = 5 / 4
275.         }
276.         var persenTerbobot = bobot * persen
277.         var persenTerbobotMax = bobot * persenMax
278.         var dataToPost = {
279.             kode: this.jexcel_instance.getValueFromCoords(0, row),
280.             jenis: this.jexcel_instance.getValueFromCoords(1, row),
281.             indikator: this.jexcel_instance.getValueFromCoords(2, row),
282.             target: target,
283.             capaian: capaian,
284.             persen: persen,
285.             persenMax: persenMax,
286.             persenTerbobot: persenTerbobot,
287.             persenTerbobotMax: persenTerbobotMax,
288.             tahun: this.jexcel_instance.getValueFromCoords(10, row)
289.         }
290.         axios({
291.             method: 'put',
292.             url: host + 'api/ike',
293.             data: dataToPost
294.         })
295.         // console.log(dataToPost)
296.     })
297.     this.init_jexcel()
298. },
299. showTahun(tahunChoice) {
300.     this.jexcel_instance.search(tahunChoice.toString())
301.     this.jexcel_instance.searchInput.dispatchEvent(new Event('keyup'))
302. }
303. }
304. }

```

setelah inisialisasi, pada mounted function, mengecek terlebih dahulu apakah hak akses up3 atau tidak, setelah itu akan menjalankan init() yang akan menginisialisasi excel IKE untuk proses edit. terdapat fungsi postIKE dan updateIKE yang masing masing untuk menyimpan data baru dan update data IKE.

```

1. var express = require('express');
2. var router = express.Router();
3. var sql = require("mssql");
4. const { executeQuery } = require("../Connection");
5.
6. //API untuk Halaman IKE
7. var routes = function () {

```

```

8.  router.route('/').get(function (req, res) {
9.    var query = "SELECT * FROM dbo.IKE ORDER BY tahun";
10.   executeQuery(res, query);
11. });
12. router.route('/').post(function (req, res) {
13.   var param = [
14.     { name: 'kode', sqltype: sql.VarChar, value: req.body.kode },
15.     { name: 'jenis', sqltype: sql.VarChar, value: req.body.jenis },
16.     { name: 'indikator', sqltype: sql.VarChar, value: req.body.indikator },
17.     { name: 'bobot', sqltype: sql.VarChar, value: req.body.bobot },
18.     { name: 'target', sqltype: sql.Float, value: req.body.target },
19.     { name: 'capaian', sqltype: sql.Float, value: req.body.capaian },
20.     { name: 'persen', sqltype: sql.Float, value: req.body.persen },
21.     { name: 'persenMax', sqltype: sql.Float, value: req.body.persen_max },
22.     { name: 'persenTerbobot', sqltype: sql.Float, value: req.body.persen_terbobot },
23.     { name: 'persenTerbobotMax', sqltype: sql.Float, value:
req.body.persen_terbobot_max },
24.     { name: 'tahun', sqltype: sql.Int, value: req.body.tahun }
25.   ];
26.   var query = "INSERT INTO
dbo.IKE(kode,jenis,indikator,bobot,target,capaian,persen,persen_max,persen_terbobot,persen_
terbobot_max,tahun)
VALUES(@kode,@jenis,@indikator,@bobot,@target,@capaian,@persen,@persenMax,@persenTerbobot,@
persenTerbobotMax,@tahun)";
27.   executeQuery(res, query, param);
28. });
29. router.route('/').put(function (req, res) {
30.   var param = [
31.     { name: 'kode', sqltype: sql.VarChar, value: req.body.kode },
32.     { name: 'jenis', sqltype: sql.VarChar, value: req.body.jenis },
33.     { name: 'indikator', sqltype: sql.VarChar, value: req.body.indikator },
34.     { name: 'bobot', sqltype: sql.VarChar, value: req.body.bobot },
35.     { name: 'target', sqltype: sql.Float, value: req.body.target },
36.     { name: 'capaian', sqltype: sql.Float, value: req.body.capaian },
37.     { name: 'persen', sqltype: sql.Float, value: req.body.persen },
38.     { name: 'persenMax', sqltype: sql.Float, value: req.body.persen_max },
39.     { name: 'persenTerbobot', sqltype: sql.Float, value: req.body.persen_terbobot },
40.     { name: 'persenTerbobotMax', sqltype: sql.Float, value: req.body.persen_terbobot_max
},
41.     { name: 'tahun', sqltype: sql.Int, value: req.body.tahun }
42.   ];
43.   var query = "UPDATE dbo.IKE SET kode=@kode, jenis=@jenis, indikator=@indikator,
bobot=@bobot, target=@target, capaian=@capaian, persen=@persen, persen_max=@persenMax,
persen_terbobot=@persenTerbobot, persen_terbobot_max=@persenTerbobotMax WHERE kode=@kode
AND tahun=@tahun";
44.   executeQuery(res, query, param);
45. });
46. router.route('/:kode').get(function(req, res) {
47.   var param = [
48.     {name: 'kode', sqltype: sql.VarChar, value: req.params.kode}
49.   ];
50.   var query = "SELECT * FROM dbo.IKE WHERE kode=@kode ORDER BY tahun";

```

```

51.   executeQuery(res, query, param)
52. });
53. router.route('/:kode/:tahun').get(function(req, res) {
54.   var param = [
55.     { name: 'kode', sqltype: sql.VarChar, value: req.params.kode },
56.     { name: 'tahun', sqltype: sql.Int, value: req.body.tahun }
57.   ];
58.   var query = "SELECT * FROM dbo.IKE WHERE kode=@kode AND tahun=@tahun";
59.   executeQuery(res, query, param)
60. });
61.
62.
63. return router
64. };
65. module.exports = routes;

```

Implementasi untuk API untuk penarikan data IKE, terdapat beberapa endpoint untuk menarik semua data, melakukan post atau entry data, edit IKE, serta tambahan api filter kode dan tahun.

```

1. import vSelect from 'vue-select'
2. import 'vue-select/dist/vue-select.css'
3.
4. import Chart from 'chart.js'
5. import { api } from '../././config/var.js'
6. var host = api
7.
8. export default {
9.   components: {
10.    vSelect
11.   },
12.   name: 'ike',
13.   data() {
14.     return {
15.       start_year: 2020,
16.       ike: [],
17.       tahun: [],
18.       jenis: [],
19.       items: {},
20.       itemList: [ ' ' ],
21.       searchTerm: '',
22.       searchTermLabel: '',
23.       searchTermPrevious: '',
24.       chartToShow: []
25.     }
26.   },
27.   async mounted() {
28.     var response = await this.fetchData()
29.     this.processData(response)
30.     this.chartToShow = this.ike

```

```

31.   this.$nextTick(() => {
32.     this.makeAllChart()
33.   })
34. },
35. methods: {
36.   async fetchData() {
37.     const response = await fetch(host + 'api/integrasi/ike')
38.     if (!response.ok) throw new Error('Something error while fetching IKE data')
39.     var res = await response.json()
40.     return res
41.   },
42.   processData(response) {
43.     var ike = []
44.     var tahun = []
45.     var jenis = []
46.     for (let i = 0; i < response.length; i++) {
47.       ike.push(response[i].kode)
48.       tahun.push(response[i].tahun)
49.       jenis.push(response[i].jenis)
50.       if (this.items.hasOwnProperty(response[i].kode)) {
51.         this.items[response[i].kode].capaian.push(parseFloat(response[i].capaian))
52.         this.items[response[i].kode].target.push(parseFloat(response[i].target))
53.       } else {
54.         this.items[response[i].kode] = {
55.           kode: response[i].kode,
56.           jenis: response[i].jenis,
57.           indikator: response[i].indikator,
58.           capaian: [parseFloat(response[i].capaian)],
59.           target: [parseFloat(response[i].target)]
60.         }
61.       }
62.     }
63.     this.ike = this.unique(ike)
64.     this.tahun = this.unique(tahun)
65.     this.jenis = this.unique(jenis)
66.     for (const kode of this.ike) {
67.       var label = kode + ' | ' + this.items[kode].jenis + ' | ' +
this.items[kode].indikator
68.       this.items[kode].label = label
69.       this.itemList.push(label)
70.     }
71.   },
72.   unique(arr) {
73.     return [...new Set(arr)]
74.   },
75.   makeAllChart() {
76.     for (const kode of this.chartToShow) {
77.       const ctx = this.$refs['chartCanvas-' + kode][0].getContext('2d')
78.       this.items[kode].chart = this.makeChart(ctx, this.items[kode].capaian, this.tahun,
this.items[kode].kode)

```

```

79.     this.items[kode].chart.render()
80.   }
81. },
82. makeChart(ctx, data, labels, label) {
83.   var chart = new Chart(ctx, {
84.     type: 'line',
85.     data: {
86.       labels: labels,
87.       datasets: [{
88.         label: label,
89.         data: data,
90.         borderWidth: 1,
91.         borderColor: 'rgba(0,0,0,0.9)',
92.         pointRadius: 6,
93.         pointHoverRadius: 9,
94.         backgroundColor: [
95.           'rgba(0, 123, 255, 0.9)'
96.           // 'rgba(54, 162, 235, 0.7)',
97.           // 'rgba(255, 206, 86, 0.7)',
98.           // 'rgba(75, 192, 192, 0.7)',
99.           // 'rgba(153, 102, 255, 0.7)',
100.          // 'rgba(255, 159, 64, 0.7)'
101.        ],
102.        lineTension: 0,
103.        fill: false
104.        // borderWidth: 10
105.      }]
106.    },
107.    options: {
108.      plugins: {},
109.      scales: {
110.        y: {
111.          beginAtZero: true,
112.          min: 0
113.        }
114.      }
115.    }
116.  })
117.  return chart
118. },
119. setSearchTerm(newSearchTerm) {
120.   this.searchTerm = newSearchTerm
121.   this.filterChart()
122. },
123. filterChart() {
124.   this.searchTermLabel = this.searchTerm
125.   if (this.searchTerm === null || this.searchTerm === '') {
126.     this.chartToShow = this.ike
127.     this.searchTerm = ''
128.     this.onSearch()

```

```

129.     } else {
130.         var searchTerm = this.searchTerm.toLowerCase()
131.         searchTerm = searchTerm.trim()
132.         this.searchTerm = ''
133.         this.onSearch()
134.         this.chartToShow = []
135.         var kode, jenis, indikator, label
136.         for (let i = 0; i < this.ike.length; i++) {
137.             kode = this.ike[i].toLowerCase()
138.             jenis = this.items[this.ike[i]].jenis.toLowerCase()
139.             indikator = this.items[this.ike[i]].indikator.toLowerCase()
140.             label = this.items[this.ike[i]].label.toLowerCase()
141.             label = label.trim()
142.             if (kode === searchTerm || jenis.includes(searchTerm) ||
indikator.includes(searchTerm) || label === searchTerm) {
143.                 this.chartToShow.push(this.ike[i])
144.             }
145.         }
146.     }
147.     this.$nextTick(() => {
148.         this.makeAllChart()
149.     })
150. },
151. onSearch() {
152.     this.itemList.splice(0, 1)
153.     this.itemList.unshift(this.searchTerm)
154. },
155. createOptionHandler() {
156.     return true
157. }
158. }
159. }

```

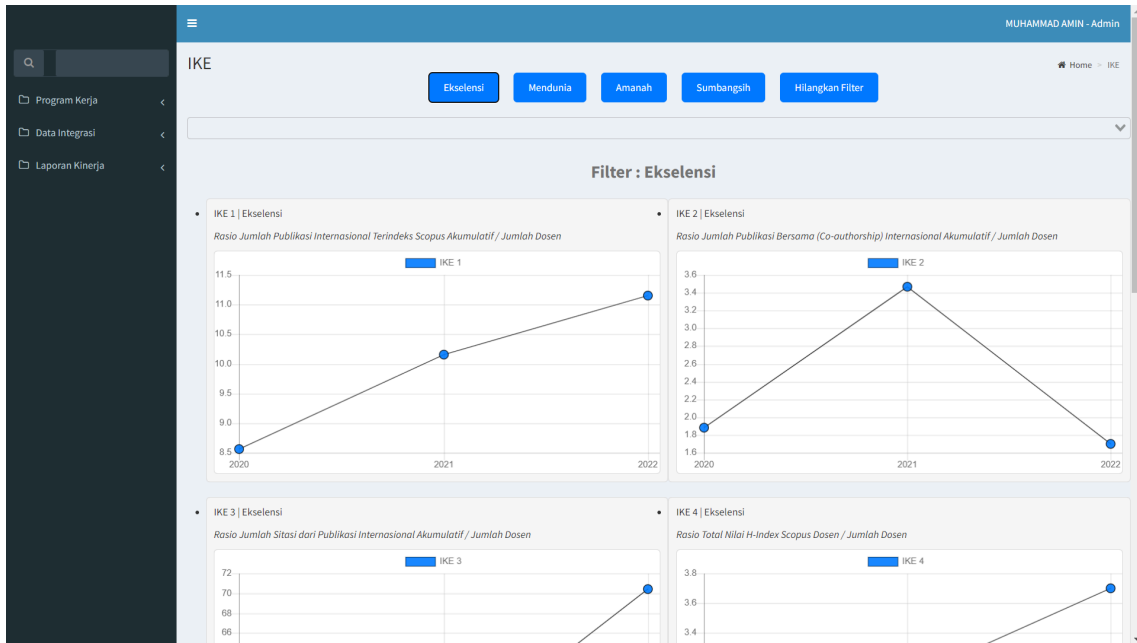
Implementasi dashboard IKE, tidak ada pengecekan hak akses. Saat mounted akan langsung menginisialisasi seluruh grafik. Setiap penggunaan filter akan memuat ulang grafik sesuai dengan filter yang sedang aktif.



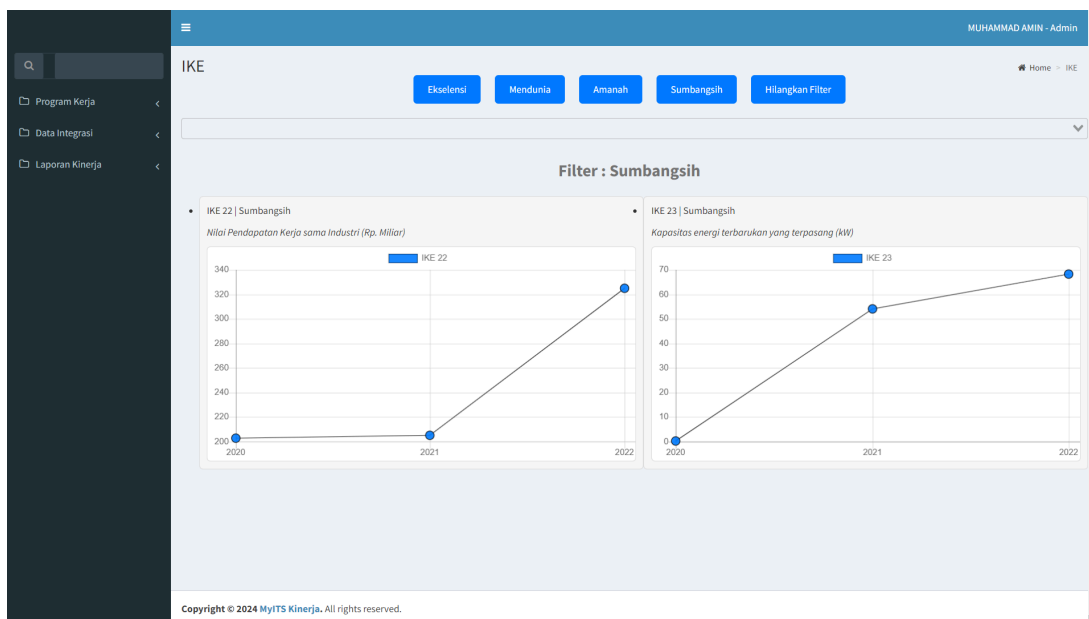
### 5.3. Implementasi Antarmuka Pengguna

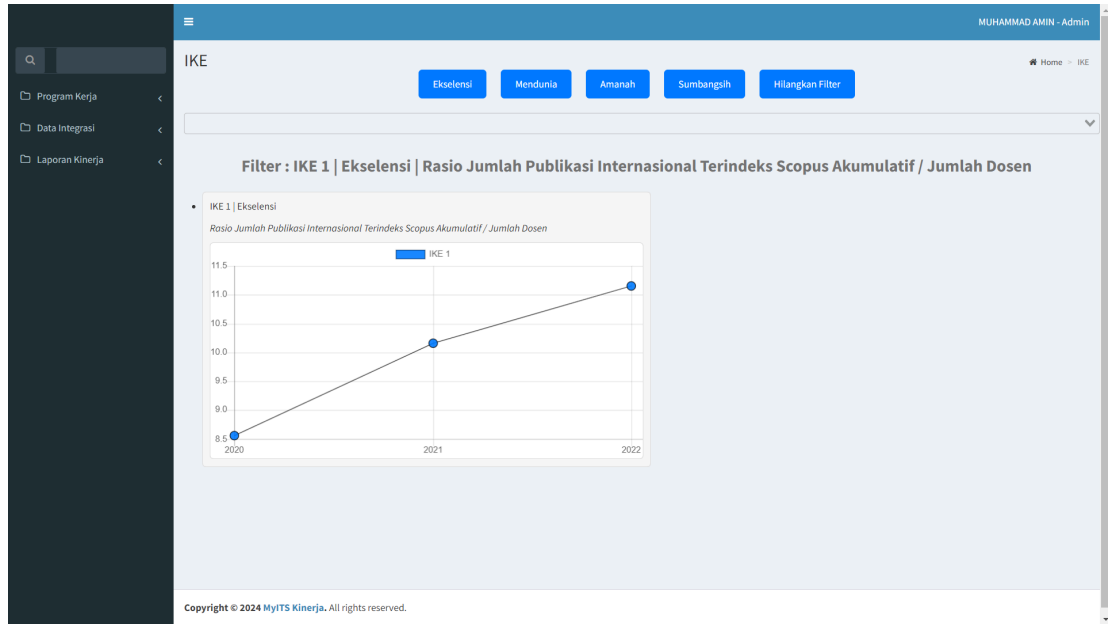
#### a. Halaman untuk user

Gambar 5.1 berisi tampilan antarmuka dashboard IKE yang berisi grafik grafik per ike.



Gambar 5.2 dan 5.3 merupakan tampilan hasil filter grafik





Gambar 5.4 berisi tampilan jika user ingin melakukan analisis data IKE di website Kinerja ITS

The screenshot shows a data table for 'IKE' with the following columns: 1. KODE, 2. JENIS, 3. INDIKATOR, 4. BOBOT, 5. TARGET, 6. CAPAIAN, 7. PERSEN, 8. PERSEN\_MAX. The table contains 12 rows of data for various indicators.

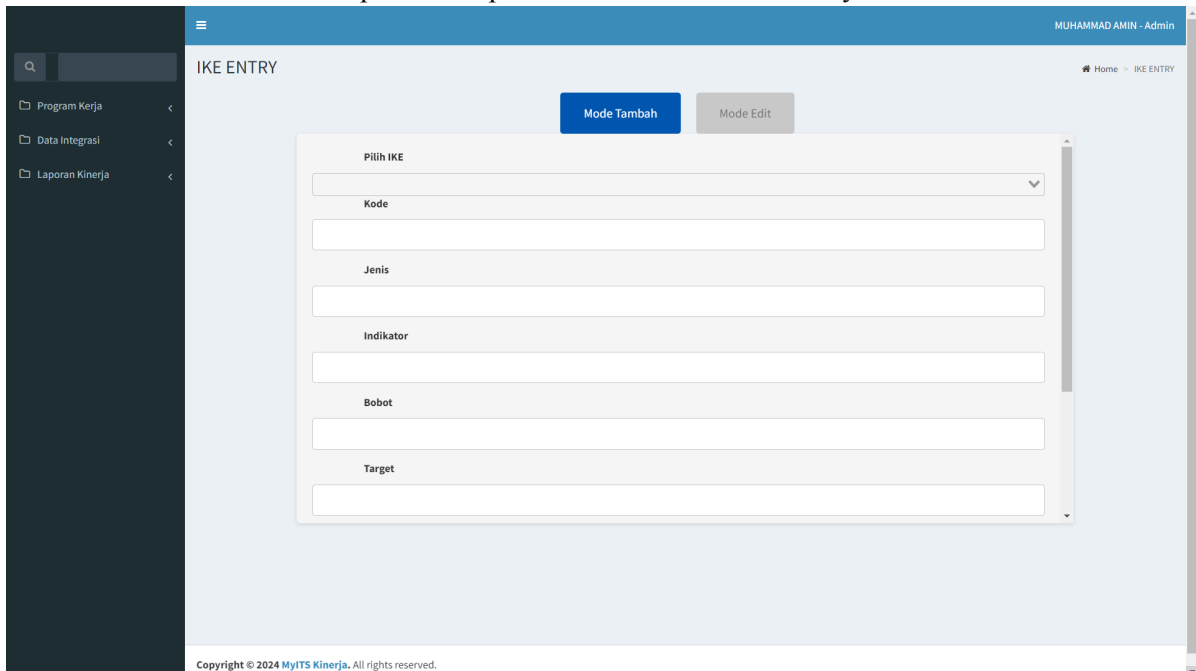
1	2	3	4	5	6	7	8
KODE	JENIS	INDIKATOR	BOBOT	TARGET	CAPAIAN	PERSEN	PERSEN_MAX
IKE 1	Ekselesi	Rasio Jumlah Publikasi Internasional Terindeks Scopus Akumulatif / Jumlah Dosen	0.061	6.9	8.56	1.24057971	1.24057971
IKE 2	Ekselesi	Rasio Jumlah Publikasi Bersama (Co-authorship) Internasional Akumulatif / Jumlah D...	0.061	1.09	1.88	1.724770642	1.25
IKE 3	Ekselesi	Rasio Jumlah Sitasi dari Publikasi Internasional Akumulatif / Jumlah Dosen	0.061	45.1	60.87	1.349667406	1.25
IKE 4	Ekselesi	Rasio Total Nilai H-Index Scopus Dosen / Jumlah Dosen	0.056	2.2	2.62	1.190909091	1.190909091
IKE 5	Ekselesi	Rasio Jumlah Judul Penelitian / Jumlah Dosen	0.05	0.9	0.78	0.866666667	0.866666667
IKE 6	Ekselesi	Rasio Jumlah Mahasiswa Pascasarjana / Jumlah Mahasiswa	0.044	0.1302	0.14	1.075268817	1.075268817
IKE 7	Ekselesi	Jumlah Kejuaraan Ranking 1 di Lomba Tingkat Nasional yang diraih Mahasiswa atau ...	0.056	55	90	1.636363636	1.25
IKE 8	Mendunia	Jumlah Program Studi Terakreditasi Internasional	0.055	10	0.53	0.053	0.053
IKE 9	Mendunia	Jumlah Kejuaraan di Lomba Tingkat Internasional yang Dijuarai Mahasiswa	0.05	20	54	2.7	1.25
IKE 10	Mendunia	Rasio Jumlah Mahasiswa Internasional / Jumlah Mahasiswa	0.049	0.03	0.01	0.333333333	0.333333333
IKE 11	Mendunia	Rasio Jumlah Dosen Internasional / Jumlah Dosen	0.047	0.13	0.03	0.230769231	0.230769231

Gambar 5.5 berisi kembalian berupa json jika user melakukan penarikan dengan mengakses “/api/ike”

```
Pretty-print
[
  {
    "kode": "IKE 1",
    "jenis": "Ekselensi",
    "indikator": "Rasio Jumlah Publikasi Internasional Terindeks Scopus Akumulatif / Jumlah Dosen",
    "bobot": 0.061,
    "target": 0.9,
    "capaian": 8.56,
    "persen": 1.24057971,
    "persen_max": 1.24057971,
    "persen_terbobot": 0.0757,
    "persen_terbobot_max": 0.0757,
    "tahun": "2020"
  },
  {
    "kode": "IKE 2",
    "jenis": "Ekselensi",
    "indikator": "Rasio Jumlah Publikasi Bersama (Co-authorship) Internasional Akumulatif / Jumlah Dosen",
    "bobot": 0.061,
    "target": 1.09,
    "capaian": 1.88,
    "persen": 1.724770642,
    "persen_max": 1.25,
    "persen_terbobot": 0.1052,
    "persen_terbobot_max": 0.0763,
    "tahun": "2020"
  },
  {
    "kode": "IKE 3",
    "jenis": "Ekselensi",
    "indikator": "Rasio Jumlah Sitasi dari Publikasi Internasional Akumulatif / Jumlah Dosen",
    "bobot": 0.061,
    "target": 45.1,
    "capaian": 60.87,
    "persen": 1.349667406,
    "persen_max": 1.25,
    "persen_terbobot": 0.0823,
    "persen_terbobot_max": 0.0763,
    "tahun": "2020"
  },
  {
    "kode": "IKE 4",
    "jenis": "Ekselensi",
    "indikator": "Rasio Total Nilai H-Index Scopus Dosen / Jumlah Dosen",
    "bobot": 0.056,
    "target": 2.2,
    "capaian": 2.62,
    "persen": 1.190909091,
    "persen_max": 1.190909091,
    "persen_terbobot": 0.0667,
    "persen_terbobot_max": 0.0667,
    "tahun": "2020"
  },
  {
    "kode": "TKP 5"
  }
]
```

## b. Halaman untuk admin

Gambar 5.6 berisi tampilan awal pada halaman “/admin/ikeentry”



Gambar 5.7 berisi tampilan jika user mengklik mode Edit

The screenshot displays the 'IKE ENTRY' application in 'Mode Edit'. The interface includes a sidebar on the left with navigation options: 'Program Kerja', 'Data Integrasi', and 'Laporan Kinerja'. The top navigation bar shows the user's name 'MUHAMMAD AMIN - Admin' and the current page 'IKE ENTRY'. The main content area features a table with the following data:

	kode	jenis	indikator	bobot	target	capaian	tahun
1	IKE 1	Ekselensi	Rasio Jumlah Publikasi Internasional Terindeks Scopus Akumulatif / Jumlah Dosen	0.061	6.9	8.56	2020
2	IKE 2	Ekselensi	Rasio Jumlah Publikasi Bersama (Co-authorship) Internasional Akumulatif / Jumlah Dosen	0.061	1.09	1.88	2020
3	IKE 3	Ekselensi	Rasio Jumlah Sitasi dari Publikasi Internasional Akumulatif / Jumlah Dosen	0.061	45.1	60.87	2020
4	IKE 4	Ekselensi	Rasio Total Nilai H-Index Scopus Dosen / Jumlah Dosen	0.056	2.2	2.62	2020
5	IKE 5	Ekselensi	Rasio Jumlah Judul Penelitian / Jumlah Dosen	0.05	0.9	0.78	2020
6	IKE 6	Ekselensi	Rasio Jumlah Mahasiswa Pascasarjana / Jumlah Mahasiswa	0.044	0.1302	0.14	2020
7	IKE 7	Ekselensi	Jumlah Kejuaraan Ranking 1 di Lomba Tingkat Nasional yang diraih Mahasiswa atau Tim Mahasiswa	0.056	55	90	2020
8	IKE 8	Mendunia	Jumlah Program Studi Terakreditasi Internasional	0.055	10	0.53	2020
-	----	..	Jumlah Kejuaraan di Lomba Tingkat Internasional yang	---	--	----	

Below the table, there are two buttons: 'Simpan Perubahan' (Save Changes) and 'Hapus Perubahan' (Delete Changes). The interface also includes a 'Mode Edit' button at the top and a 'Select Year' dropdown menu.

Copyright © 2024 MyITS Kinerja. All rights reserved.

*[Halaman ini sengaja dikosongkan]*

## **BAB VI PENGUJIAN DAN EVALUASI**

Bab ini menjelaskan tahap uji coba dilakukan terhadap Dashboard IKE. 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 Dashboard IKE guna menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi

### **6.2. Kriteria Pengujian**

Penilaian atas pencapaian tujuan pengujian didapatkan dengan memerhatikan beberapa hasil yang diharapkan berikut ini:

- a. Kemampuan aplikasi menampilkan berbagai form mulai dari form login, form pengajuan peminjaman lisensi, perubahan durasi peminjaman lisensi, hingga form perubahan kuota upload dokumen.
- b. Kemampuan aplikasi menampilkan berbagai tabel mulai dari hasil pengajuan peminjaman lisensi, history peminjaman lisensi, antrian pengajuan peminjaman lisensi, lisensi yang sedang terpakai dari sisi admin, hingga history pengunggahan dokumen
- c. Kemampuan aplikasi menampilkan detail lisensi yang sedang dipinjam, melakukan revoke antrian baik dari sisi user maupun admin, melakukan penggantian password.
- d. Kemampuan aplikasi untuk menyimpan progress pengisian data dengan secara otomatis pengguna dapat melanjutkan pengisian data baru sementara pengisian sebelumnya masih tersimpan dan juga dapat diperbarui.
- e. Kemampuan aplikasi untuk menampilkan grafik pemakaian peminjaman lisensi baik skala fakultas maupun skala departemen
- f. Kemampuan aplikasi memenuhi kebutuhan lainnya, yaitu mencari data dengan filter pada tabel, mengurutkan data pada kolom tabel tertentu, serta mengunduh data.
- g. Kesesuaian dalam memenuhi kebutuhan non-fungsional aplikasi, yaitu:
  - Pengaksesan sistem dari area yang terhubung internet
  - Sistem memiliki tampilan (antarmuka) yang mudah dipahami

### **6.3. Skenario Pengujian**

Skenario pengujian dilakukan dengan melakukan peran sebagai admin, dosen atau tendik, serta mahasiswa yang akan menjalankan fitur-fitur dan seluruh kebutuhan fungsional dari sistem. Langkah-langkah untuk setiap kebutuhan fungsional yaitu sebagai berikut:

#### **6.3.1. Admin**

- a. Admin melakukan login.
- b. Admin melakukan penggantian role.
- c. Admin mengisi formulir pengajuan peminjaman lisensi Adobe.
- d. Admin melihat grafik peminjaman lisensi Adobe
- e. Admin mengisi kredensial Lisensi Adobe
- f. Admin melihat history peminjaman lisensi
- g. Admin melihat antrian dari request pengajuan peminjaman lisensi Adobe.
- h. Admin melihat detail dari peminjaman lisensi yang sedang dipinjam
- i. Admin melakukan revoke pada lisensi yang sedang aktif dipakai oleh User
- j. Admin membatalkan antrian user pada pengajuan peminjaman lisensi Adobe
- k. Admin mengubah durasi peminjaman lisensi
- l. Admin melihat daftar dokumen yang telah terunggah ke iThenticate
- m. Admin melihat detail dokumen yang telah terunggah ke iThenticate

- n. Admin mengunggah dokumen ke iThenticate.
- o. Admin mengubah kuota unggahan dokumen
- p. Admin melihat kuota unggahan dokumen.

**6.3.2. Dosen, Tendik, dan Mahasiswa**

- a. Dosen, Tendik, dan Mahasiswa melakukan login.
- b. Dosen, Tendik, dan Mahasiswa melakukan penggantian role.
- c. Dosen, Tendik, dan Mahasiswa mengisi formulir pengajuan peminjaman lisensi Adobe.
- d. Dosen, Tendik, dan Mahasiswa melihat history peminjaman lisensi
- e. Dosen, Tendik, dan Mahasiswa melihat antrian dari request pengajuan peminjaman lisensi Adobe.
- f. Dosen, Tendik, dan Mahasiswa melihat detail dari peminjaman lisensi yang sedang dipinjam
- g. Dosen, Tendik, dan Mahasiswa melakukan revoke pada lisensi yang sedang dipinjam
- h. Dosen, Tendik, dan Mahasiswa membatalkan antrian pada pengajuan peminjaman lisensi Adobe
- i. Dosen, Tendik, dan Mahasiswa melihat daftar dokumen yang telah terunggah ke iThenticate
- j. Dosen, Tendik, dan Mahasiswa melihat detail dokumen yang telah terunggah ke iThenticate
- k. Dosen, Tendik, dan Mahasiswa mengunggah dokumen ke iThenticate.
- l. Dosen, Tendik, dan Mahasiswa melihat kuota unggahan dokumen.

**6.4. Evaluasi Pengujian**

Hasil pengujian dilakukan terhadap pengamatan mengenai perilaku sistem Aplikasi MyITS License Management terhadap kasus skenario uji coba. Pengujian dilakukan oleh pihak pengembang, pengguna, dan pembimbing lapangan. Tabel 6.1 menjelaskan hasil uji coba terhadap aplikasi yang telah dibuat.

*Tabel 6. 1 Hasil Evaluasi Pengujian Dashboard IKE*

<b>Kriteria Pengujian</b>	<b>Hasil Pengujian</b>
Melihat Grafik IKE	Terpenuhi
Melakukan Entry Data IKE	Terpenuhi
Melakukan Penggantian Role	Terpenuhi
Mengisi Formulir Pengajuan Peminjaman Lisensi Adobe (Admin dan User)	Terpenuhi
Mengisi Kredensial Lisensi (Admin)	Terpenuhi

## **BAB VII**

### **KESIMPULAN DAN SARAN**

#### **7.1. Kesimpulan**

Kesimpulan yang didapat setelah melakukan pengembangan dashboard pada kegiatan KP di UP3 adalah sebagai berikut:

- Dashboard yang dibangun berguna untuk keperluan analisis Indeks Kinerja Emas ITS.
- Dengan adanya dashboard Indeks Kinerja Emas pada website Kinerja ITS, administrator bisa melakukan pencatatan IKE ITS tiap tahunnya.

#### **7.2. Saran**

Berikut ini adalah saran yang penulis berikan untuk arah perkembangan selanjutnya, yaitu sebaiknya ada kasus pergantian deskripsi dan penambahan IKE karena susunan IKE dapat berganti di masa depan.



*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- [1] Elisa, Edi, 2016. *Aplikasi Berbasis Web*. Diakses pada 08 Juli 2023 dari: <http://edel.staff.unja.ac.id/blog/artikel/Aplikasi-Berbasis-Web.html>, Juli 2023.
- [2] Guntoro, 2020. *Panduan Awal Belajar Pemrograman Web Dalam 10 Menit*. Diakses pada 08 Juli 2023 dari: <https://badoystudio.com/belajar-pemrograman-web/>, Juli 2020.
- [3] Alvian, 2019. *Panduan Awal Belajar Pemrograman Web Dalam 10 Menit*. Diakses pada 08 Juli 2023 dari: <https://sis.binus.ac.id/2019/02/25/hubungan-dan-perbedaan-javascript-html-css-jquery-dan-php-di-dalam-web-development/>, Juli 2023.
- [4] Yasin, 2019. *Pengertian MySQL, Fungsi, dan Cara Kerjanya (Lengkap)*. Diakses pada 08 Juli 2023 dari: <https://www.niagahoster.co.id/blog/mysql-adalah/>, Juli 2023.
- [5] Wijaya, Ketut Krisna, 2016. *Visual Studio Code: Aplikasi Editor Kode dari Microsoft untuk Windows, Linux, dan OS X*. Diakses pada 08 Juli 2023 dari: <https://id.techinasia.com/visual-studio-code-editor-kode-microsoft>, Juli 2023.
- [6] Yasin, 2018. *Apa Itu Web Server dan Fungsinya?*. Diakses pada 08 Juli 2023 dari: <https://www.niagahoster.co.id/blog/web-server-adalah/>, Juli 2023.
- [7] *Perbedaan Web Hosting, VPS, dan Cloud Server yang Perlu Diketahui Sebelum Membangun Website*. Diakses pada 08 Juli 2023 dari: <https://blog.gamatechno.com/perbedaan-web-hosting-vps-cloud-server/>, Juli 2023.
- [8] Yasin, 2019. *Pengertian jQuery Serta Fungsi dan Contohnya*. Diakses pada 08 Juli 2023 dari: <https://www.niagahoster.co.id/blog/jquery-adalah/>, Juli 2023.
- [9] Ham, Hanry. *Kelebihan Menggunakan Laravel Web Development*. Diakses pada 08 Juli 2023 dari: <https://socs.binus.ac.id/2018/12/13/kelebihan-menggunakan-laravel-web-development/>, Juli 2023.
- [10] Yasin, 2019. *Cara Menggunakan Bootstrap 4 Untuk Membuat Website*. Diakses pada 08 Juli 2023 dari: <https://www.niagahoster.co.id/blog/cara-menggunakan-bootstrap/>, Juli 2023.

*[Halaman ini sengaja dikosongkan]*

## BIODATA PENULIS



Muhammad Amin dilahirkan di Balikpapan, 16 September 2002, merupakan anak bungsu dari 8 bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Kartika Balikpapan, SDN 001 Balikpapan Tengah, SMPN 3 Balikpapan dan SMAN 1 Balikpapan. Setelah lulus dari SMA tahun 2020. Penulis mengikuti SBMPTN dan diterima di Departemen Teknik Informatika FTEIC - ITS pada tahun 2020 dan terdaftar dengan NRP 5025201251 Di Departemen Teknik Informatika Penulis sempat aktif di Schematics NPC 2021 dan 2022 yang diselenggarakan oleh Mahasiswa Teknik Informatika, dan aktif sebagai Asisten dosen di beberapa mata kuliah seperti: pemrograman dasar, struktur data, kecerdasan buatan, kecerdasan komputasional, pembelajaran mesin, otomata, pemrograman berbasis kerangka kerja, sistem operasi, dan jaringan komputer. Penulis juga aktif di UKM Bridge ITS dan meraih banyak juara mulai dari tingkat kota hingga nasional.