

The background features a large, abstract geometric design on the left side, consisting of a black triangle pointing downwards and a blue triangle pointing upwards, meeting at a diagonal line. The rest of the background is white, overlaid with a pattern of light blue and grey hexagons and lines, resembling a molecular or network structure. Some hexagons are solid, while others are outlines, and some are connected by thin lines.

# Manual Book **SQLearn**

Putra Prima Arhandi

Muhammad Shulhan Khairy

Muhammad Ilham Adhim

## DAFTAR ISI

<b>DAFTAR ISI</b> .....	<b>2</b>
<b>DAFTAR GAMBAR</b> .....	<b>3</b>
<b>Abstrak</b> .....	<b>4</b>
<b>1. Software Requirement</b> .....	<b>5</b>
<b>2. Menjalankan Aplikasi SQLearn</b> .....	<b>5</b>
<b>3. Source Code</b> .....	<b>28</b>
<b>4. Kesimpulan</b> .....	<b>47</b>

## DAFTAR GAMBAR

Gambar 1 URL Browser.....	5
Gambar 2 Login Page .....	5
Gambar 3 Halaman Dosen Admin .....	6
Gambar 4 Tambah Dosen Modal .....	6
Gambar 5 Edit Dosen Modal.....	7
Gambar 6 Modal Konfirmasi Delete Dosen .....	7
Gambar 7 Grading Rules Admin.....	8
Gambar 8 Edit Grading Rules .....	8
Gambar 9 Delete Grading Rules.....	9
Gambar 10 Threshold Setting.....	9
Gambar 11 Ubah Profile Admin.....	10
Gambar 12 Halaman Jadwal Dosen .....	10
Gambar 13 Add Jadwal Modal.....	11
Gambar 14 Edit Jadwal Modal .....	11
Gambar 15 Delete Jadwal Modal .....	12
Gambar 16 Daftar Kelas .....	12
Gambar 17 Menambahkan kelas baru .....	13
Gambar 18 Mengubah data kelas .....	13
Gambar 19 Menghapus data kelas.....	14
Gambar 20 Preview Detail Kelas .....	14
Gambar 21 Tambah Mahasiswa baru .....	14
Gambar 22 Daftar Studi Kasus.....	15
Gambar 23 Tambah Studi Kasus .....	15
Gambar 24 Menghapus Studi Kasus .....	16
Gambar 25 Preview Studi Kasus.....	16
Gambar 26 Daftar Paket Soal .....	17
Gambar 27 Preview Paket Soal Close-Ended .....	18
Gambar 28 Tambah Soal ke Paket Soal Close-Ended.....	18
Gambar 29 Modal Hapus Paket Soal.....	19
Gambar 30 Daftar Soal Close-Ended .....	19
Gambar 31 Pembuatan Soal Baru Close-Ended.....	21
Gambar 32 Hapus Soal .....	22
Gambar 33 Tampilan Daftar Nilai.....	22
Gambar 34 Log Aktivitas Mahasiswa .....	22
Gambar 35 Jadwal Tersedia .....	23
Gambar 36 Jadwal Selesai.....	24
Gambar 37 Kerjakan Jadwal .....	24
Gambar 38 Konfirmasi Kerjakan .....	24
Gambar 39 Pengerjaan Soal .....	25
Gambar 40 Hasil Query Jawaban Benar .....	26
Gambar 41 Modal Akhiri Ujian .....	27
Gambar 42 Modal Loading Penilaian.....	27
Gambar 43 Pengerjaan selesai.....	27
Gambar 44 Ubah Profile Mahasiswa.....	28

## Abstrak

Structured Query Language (SQL) umum digunakan untuk kebutuhan industri dan akademik, khususnya di bidang teknologi informasi. SQL menggunakan sintaks deklaratif untuk mendapatkan data dari basis data. Untuk membuat query yang efektif dalam mempelajari query SQL, murid diharuskan untuk memvisualisasikan berbagai elemen dalam konsep basis data dan memahami bagaimana cara memperoleh data tersebut, hal ini memungkinkan untuk adanya beban kognitif yang dialami oleh pelajar. Penelitian ini mengusulkan pendekatan close-ended dalam pemberian petunjuk komponen SQL, serta drag-and-drop sebagai metode baru untuk membantu mahasiswa membentuk query SQL yang valid. Dalam penerapannya, diimplementasikan dalam bentuk aplikasi latihan berbasis *web* bernama SQLearn. Untuk mengetahui dampak dari metode yang diajukan, dilakukan sebuah pengambilan data oleh 28 partisipan diarahkan untuk mengikuti pre-test, latihan query SQL di platform SQLearn, dan post-test. Data tersebut diolah menggunakan uji *Kolmogorov-Smirnov* untuk uji normalitas data pre-test dan post-test. Hasil pengujian normalitas mengindikasikan bahwa data pre-test dan post-test tidak terdistribusi normal. Karena data tidak terdistribusi normal, selanjutnya data akan diuji menggunakan non-parametrik yaitu *Wilcoxon Signed Ranks Test* guna mengetahui apakah terdapat perbedaan rata-rata yang signifikan dari data pre-test dan post-test mahasiswa. Hasil pengujian Wilcoxon menunjukkan bahwa nilai rata-rata nilai post-test (73.2143) lebih tinggi daripada rata-rata nilai pre-test (60.7143). Selain itu, hasil uji Wilcoxon juga memiliki nilai *Asymp. Sig. (2-tailed)* sebesar 0.000 ( $< 0.05$ ). Berdasarkan dasar pengambilan keputusan, dapat disimpulkan bahwa terdapat peningkatan skor pre-test dan post-test secara signifikan. Oleh karena itu, penerapan metode close-ended dan implementasi drag-and-drop memiliki dampak positif yang signifikan terhadap nilai post-test mahasiswa terkait konsep query SQL.

**Kata kunci**— *SQL Query, Drag-and-drop, Data Manipulation Language*

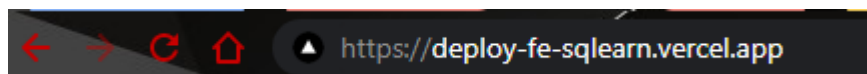
## 1. Software Requirement

Software yang dibutuhkan untuk dapat menjalankan aplikasi ini adalah sebagai berikut:

- a. PC atau Laptop dengan sistem operasi Windows atau Linux atau MacOS atau ChromeOS
- b. Web Browser minimal Google Chrome ver 102.0.5005, Mozilla Firefox ver 99.0, Microsoft Edge ver 101.0.1210.53, Apple Safari ver 15.4

## 2. Menjalankan Aplikasi SQLearn

Buka browser dan masukkan url <https://deploy-fe-sqllearn.vercel.app/> pada browser



*Gambar 1 URL Browser*

Pertama kali aplikasi dibuka akan muncul halaman login, terdapat 3 pilihan role login yaitu sebagai: mahasiswa, dosen, dan admin

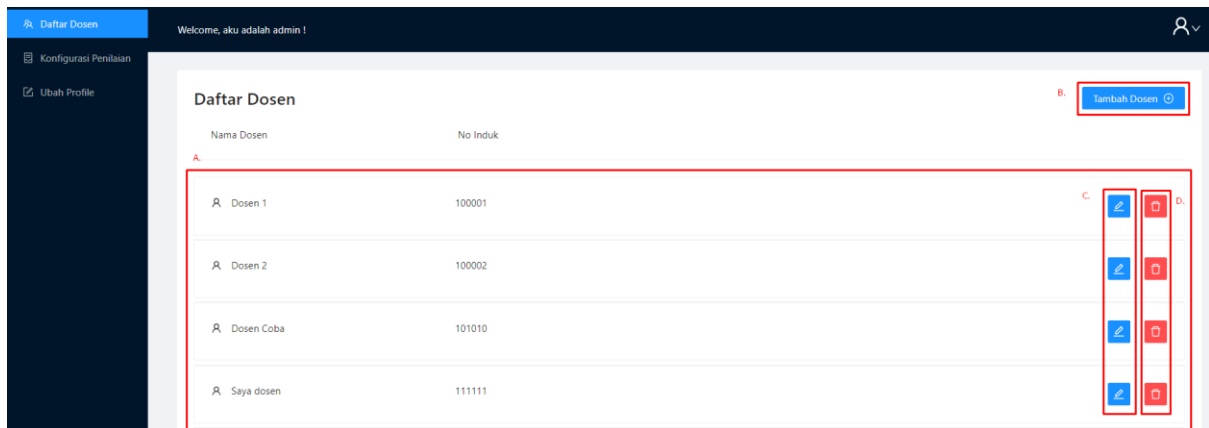
A screenshot of a login form. At the top left, it says 'Login Sebagai'. To its right is a dropdown menu with 'Mahasiswa' selected. Below this are two input fields: 'Username mahasiswa . . .' and 'Password'. There is a 'Remember me' checkbox which is checked. At the bottom is a blue 'Log in' button.

*Gambar 2 Login Page*

Agar dapat login dibutuhkan username dan password dan memilih role yang sesuai.

### 2.1. Role Admin

#### 2.1.1. Dosen

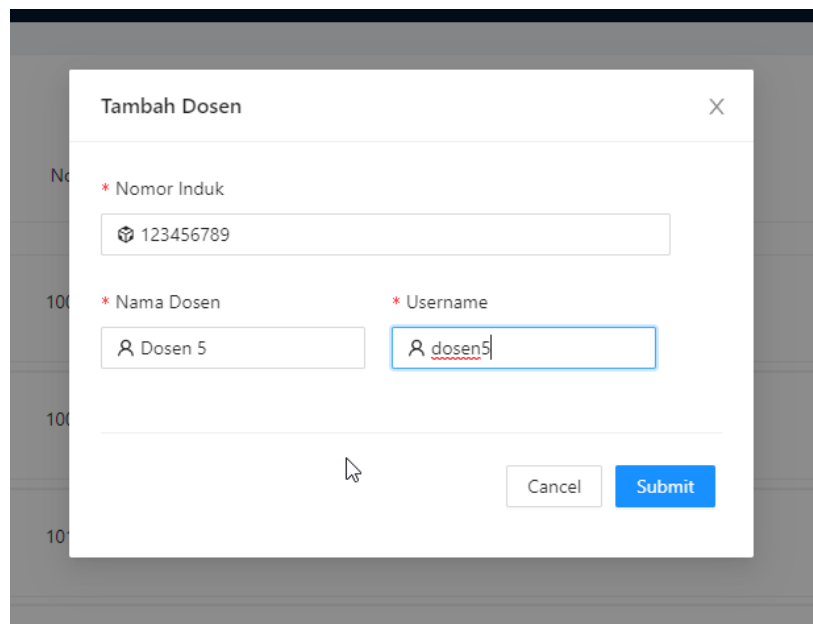


**Gambar 3 Halaman Dosen Admin**

Setelah berhasil login sebagai admin akan diarahakan ke halaman dosen, admin dapat

- a) Melihat daftar dosen
- b) Menambahkan dosen

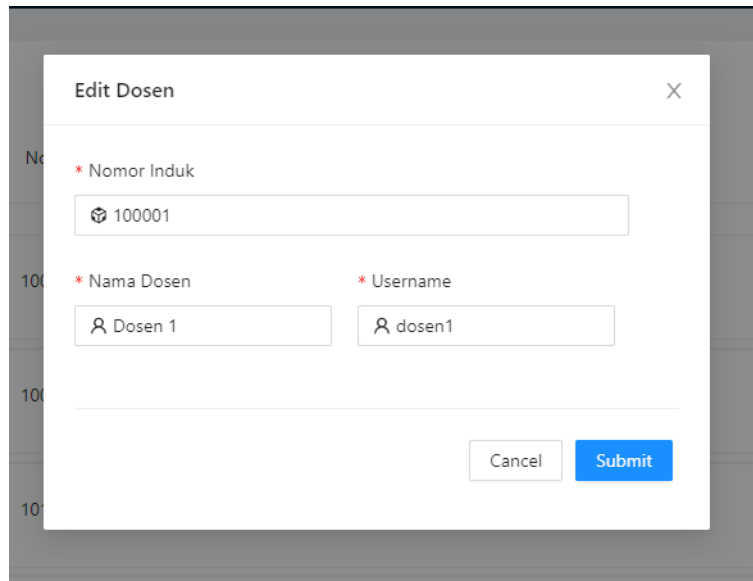
Untuk menambahkan dosen dapat menekan tombol tambah dosen, kemudian akan muncul modal dimana dapat diisi nomor induk dosen, nama dosen, dan username dosen yang akan digunakan untuk login, secara default password dosen adalah sama dengan username, tekan tombol submit untuk menyimpan data.



**Gambar 4 Tambah Dosen Modal**

- c) Mengubah dosen

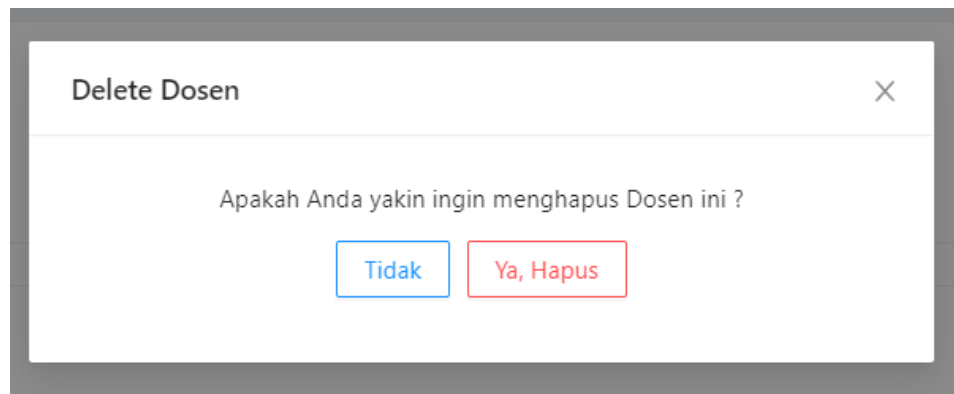
Untuk Mengubah dosen dapat menekan tombol edit dosen yaitu huruf c pada gambar 4, kemudian akan muncul modal untuk edit data dosen, masukkan perubahan yang dibutuhkan kemudian tekan submit.



*Gambar 5 Edit Dosen Modal*

d) Menghapus dosen

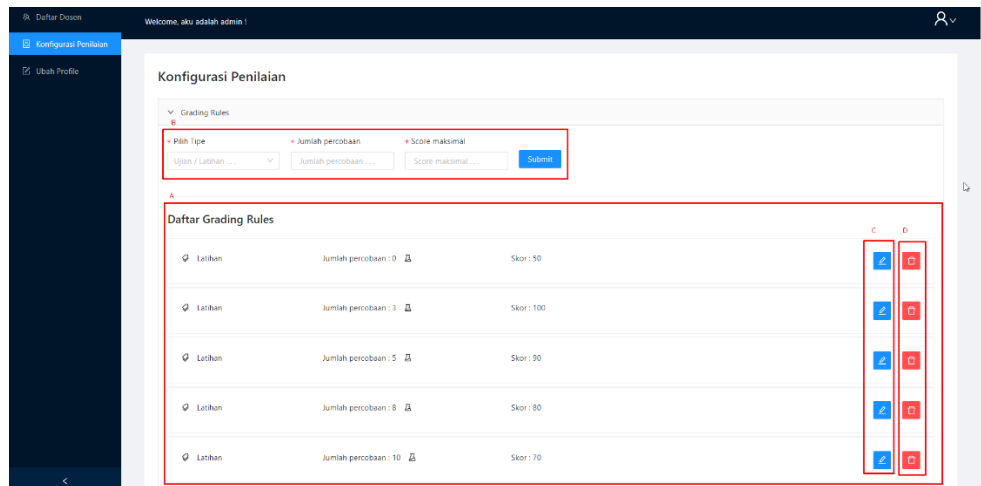
Untuk menghapus dosen dapat menekan tombol hapus dosen yaitu huruf d pada gambar 4, kemudian akan muncul modal konfirmasi untuk menghapus dosen, klik Ya, Hapus untuk menghapus data.



*Gambar 6 Modal Konfirmasi Delete Dosen*

## 2.1.2. Konfigurasi Penilaian

### 2.1.2.1. Grading Rules



**Gambar 7 Grading Rules Admin**

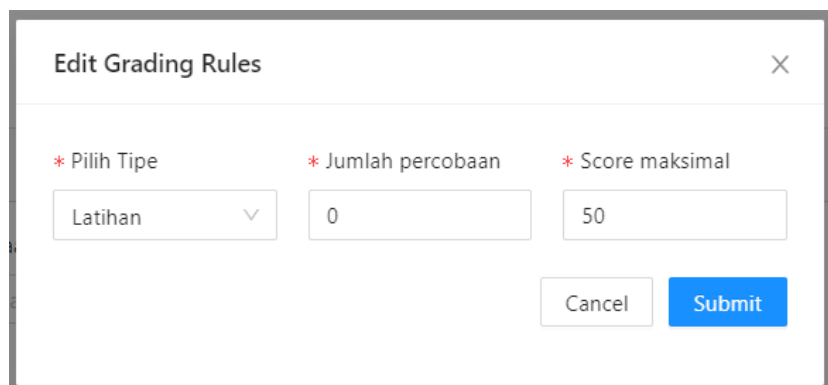
a) Daftar grading rules

b) Tambah grading rules

Untuk menambah grading rules dapat mengisi tipe, jumlah percobaan dan score yang didapatkan pada kotak B, dan simpan dengan klik submit

c) Edit grading rules

Untuk mengubah grading rules dapat klik tombol biru pada kotak C dari data rules yang ingin diubah, kemudian akan muncul popup modal untuk dapat mengubah data yang diinginkan, klik submit untuk menyimpan data.



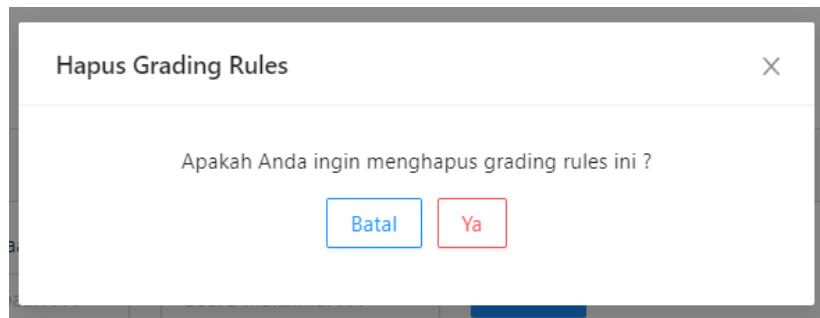
**Gambar 8 Edit Grading Rules**

d) Delete grading rules

Untuk menghapus grading rules dapat klik tombol merah pada kotak d dari data rules yang ingin dihapus, kemudian akan muncul



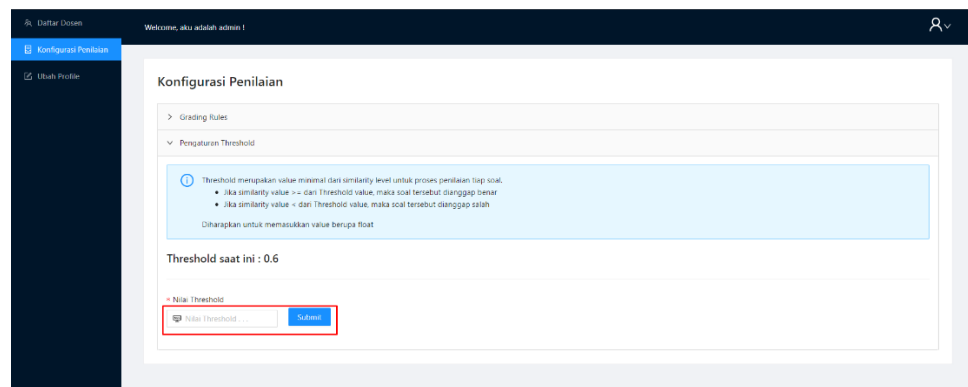
popup modal untuk konfirmasi menghapus data, klik Ya untuk konfirmasi menghapus data.



*Gambar 9 Delete Grading Rules*

### 2.1.2.2. Threshold

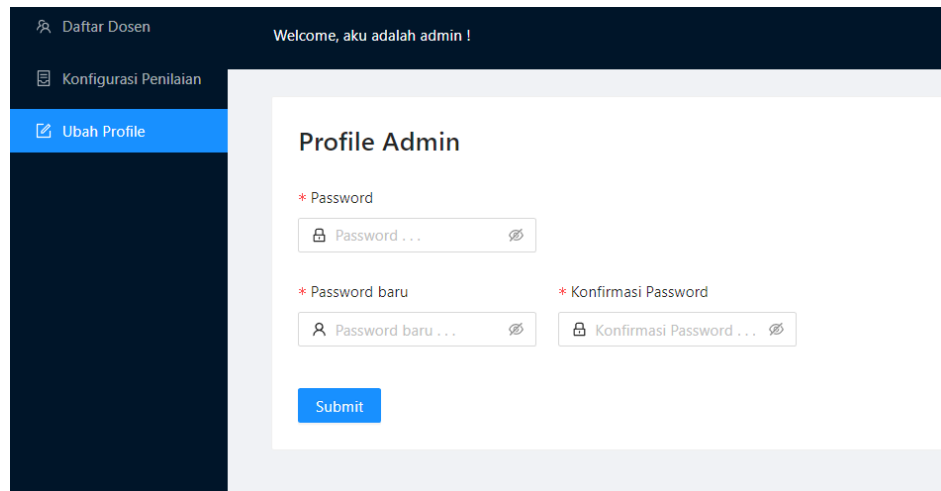
Admin dapat mengubah threshold similarity yang digunakan untuk menjadi batas bawah nilai similarity sebelum dilakukan pengecekan hasil query, untuk mengubah nilai threshold dapat dilakukan dengan cara mengisi nilai threshold dengan nilai antara 0 – 1 dengan koma, seperti 0,3, jika sudah klik submit untuk menyimpan.



*Gambar 10 Threshold Setting*

### 2.1.3. Ubah Profile

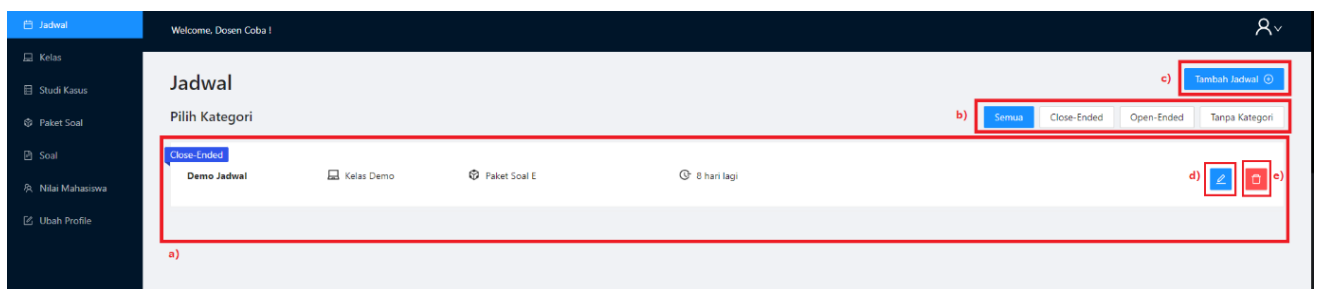
Admin dapat merubah password sesuai dengan keinginan pada halaman ubah profile, dimana untuk mengubah password harus memasukkan password sekarang, kemudian masukkan password baru dan konfirmasi password baru, jika sudah klik submit.



*Gambar 11 Ubah Profile Admin*

## 2.2. Role Dosen

### 2.2.1. Jadwal



*Gambar 12 Halaman Jadwal Dosen*

Setelah berhasil login sebagai dosen maka akan diarahkan ke halaman jadwal, dosen dapat melakukan aksi sebagai berikut

- a) Melihat daftar jadwal yang dibuat
- b) Melakukan filter jadwal berdasarkan tanggal mulai dan tanggal berakhir
- c) Menambahkan jadwal.

Untuk menambahkan jadwal dapat menekan tombol Tambah Jadwal pada point C maka akan muncul modal untuk menambahkan jadwal kedalam sistem dengan mengisi nama jadwal, tipe jadwal, tanggal mulai, tanggal berakhir, kelas, dan paket soal kemudian klik submit.

**Tambah Jadwal** [X]

\* Nama Jadwal:

\* Tipe:

\* Waktu Mulai:

\* Waktu Selesai:

\* Kelas:

\* Paket Soal:

[Cancel] [Submit]

*Gambar 13 Add Jadwal Modal*

d) Mengubah jadwal.

Untuk mengubah jadwal dapat menekan tombol D yang ada pada setiap jadwal kemudian akan muncul modal beserta data jadwal, dapat mengubah sesuai yang dibutuhkan kemudian klik submit

**Edit Jadwal** [X]

\* Nama Jadwal:

\* Tipe:

\* Waktu Mulai:

\* Waktu Selesai:

\* Kelas:

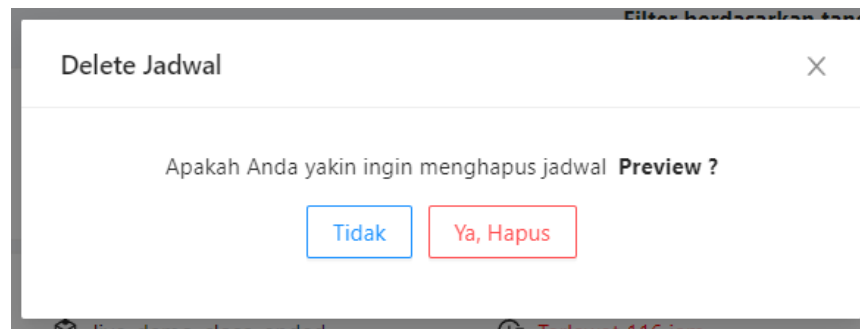
\* Paket Soal:

[Cancel] [Submit]

*Gambar 14 Edit Jadwal Modal*

e) Menghapus jadwal.

Untuk menghapus jadwal dapat menekan tombol E yang ada pada setiap jadwal kemudian akan muncul modal beserta konfirmasi apakah ingin menghapus, tekan ya, hapus untuk menghapus jadwal.

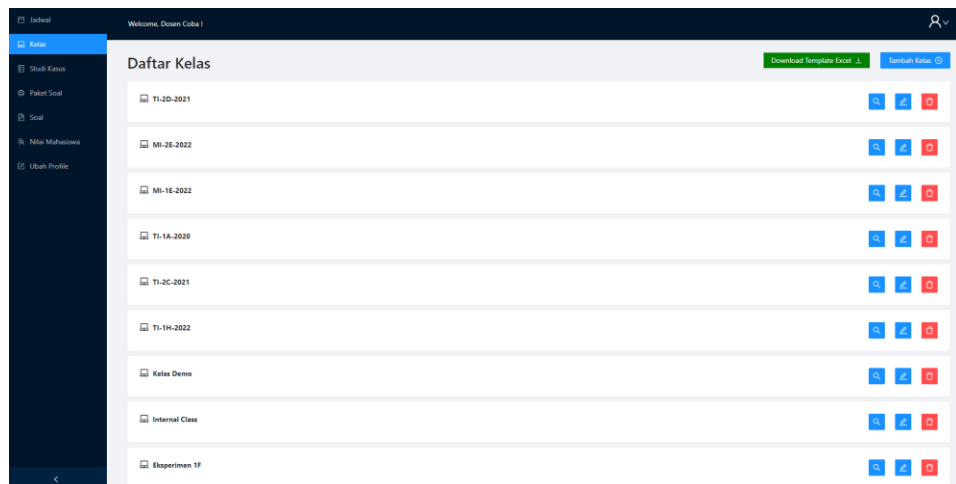


*Gambar 15 Delete Jadwal Modal*

### 2.2.2. Kelas

Setelah berhasil login sebagai dosen maka halaman yang dapat diakses oleh dosen selanjutnya adalah halaman kelas, pada halaman ini terdapat fitur fitur sebagai berikut:

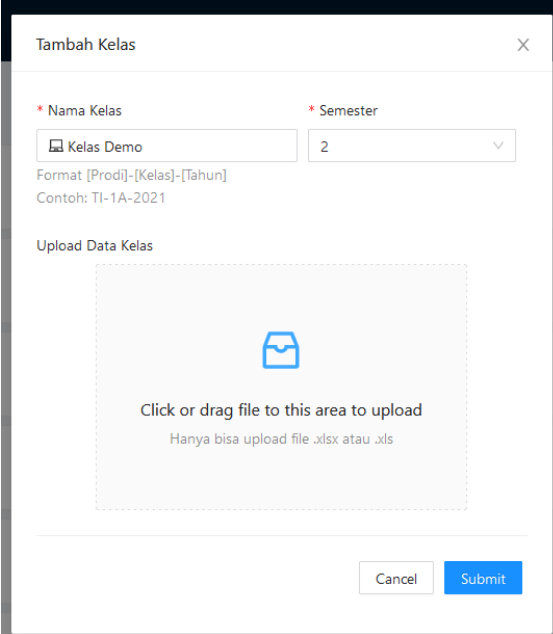
- a) Melihat daftar kelas



*Gambar 16 Daftar Kelas*

b) Menambahkan kelas baru


Jika ingin menambahkan data kelas baru, dosen perlu memasukkan informasi nama kelas, semester, dan Upload data kelas dalam bentuk file excel. Telah disediakan juga untuk download template file excel yang bisa digunakan untuk menambahkan daftar mahasiswa dalam kelas tersebut dengan cepat.

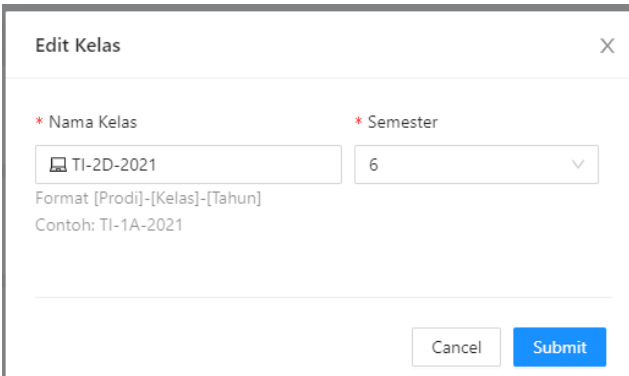


The screenshot shows a modal window titled "Tambah Kelas" with a close button (X) in the top right corner. It contains two required fields: "Nama Kelas" (Class Name) with a text input containing "Kelas Demo" and "Semester" with a dropdown menu set to "2". Below these fields, there is a format instruction: "Format [Prodi]-[Kelas]-[Tahun]" and an example: "Contoh: TI-1A-2021". A section titled "Upload Data Kelas" features a dashed box with a blue folder icon and the text "Click or drag file to this area to upload" and "Hanya bisa upload file .xlsx atau .xls". At the bottom, there are "Cancel" and "Submit" buttons.

*Gambar 17 Menambahkan kelas baru*

c) Mengubah kelas


Untuk mengubah sebuah kelas, dosen perlu menekan tombol  dari item kelas yang ingin dipilih, maka akan muncul sebuah modal. Dosen dapat memasukkan data nama kelas dan semester yang baru untuk kelas yang ingin diubah

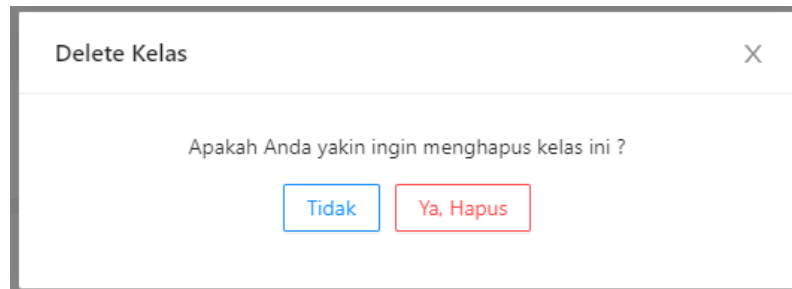


The screenshot shows a modal window titled "Edit Kelas" with a close button (X) in the top right corner. It contains two required fields: "Nama Kelas" (Class Name) with a text input containing "TI-2D-2021" and "Semester" with a dropdown menu set to "6". Below these fields, there is a format instruction: "Format [Prodi]-[Kelas]-[Tahun]" and an example: "Contoh: TI-1A-2021". At the bottom, there are "Cancel" and "Submit" buttons.

*Gambar 18 Mengubah data kelas*

d) Menghapus kelas

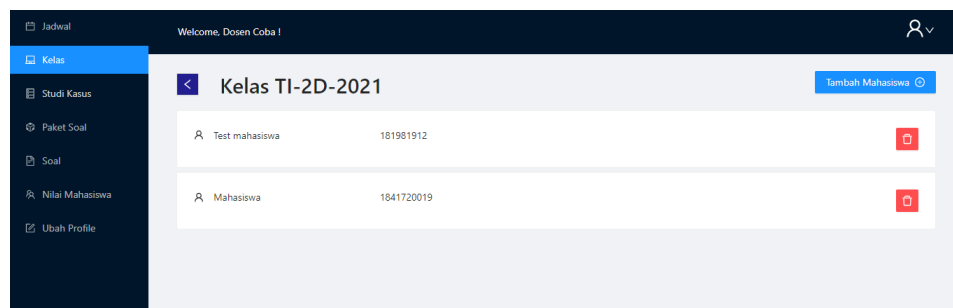
Untuk menghapus kelas dapat menekan tombol  yang ada pada setiap kelas. Kemudian akan muncul modal beserta konfirmasi apakah ingin menghapus, tekan ya, hapus untuk menghapus jadwal.



*Gambar 19 Menghapus data kelas*

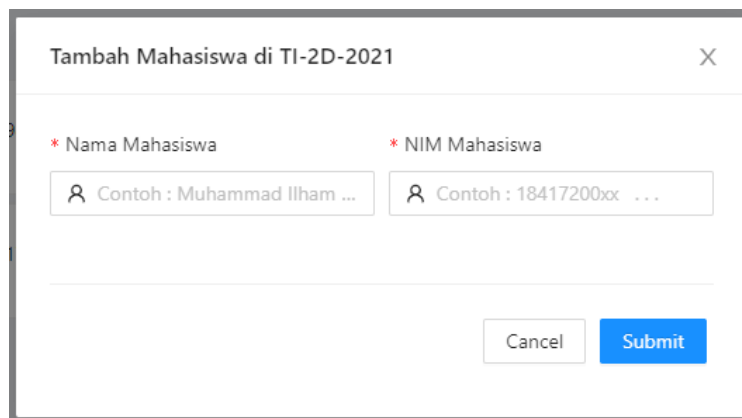
e) Preview detail kelas

Dalam detail kelas, dosen dapat melihat, menambahkan dan menghapus mahasiswa dari kelas tersebut



*Gambar 20 Preview Detail Kelas*

Untuk menambahkan mahasiswa dalam kelas tertentu, dosen perlu memasukkan nama dan nim dari mahasiswa tersebut.

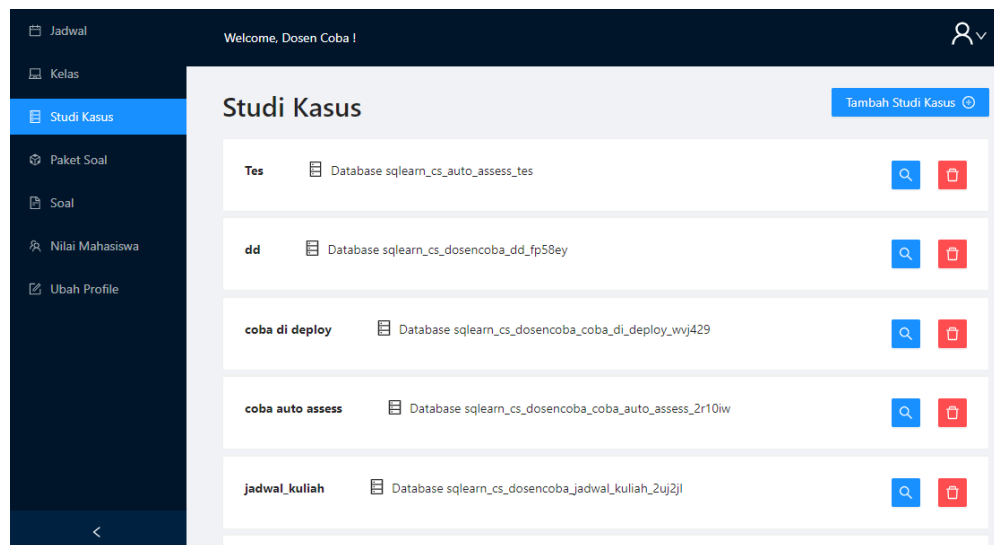


*Gambar 21 Tambah Mahasiswa baru*

### 2.2.3. Studi Kasus

Studi kasus adalah database yang dapat digunakan untuk Latihan soal siswa. Pada halaman ini terdapat fitur fitur sebagai berikut:

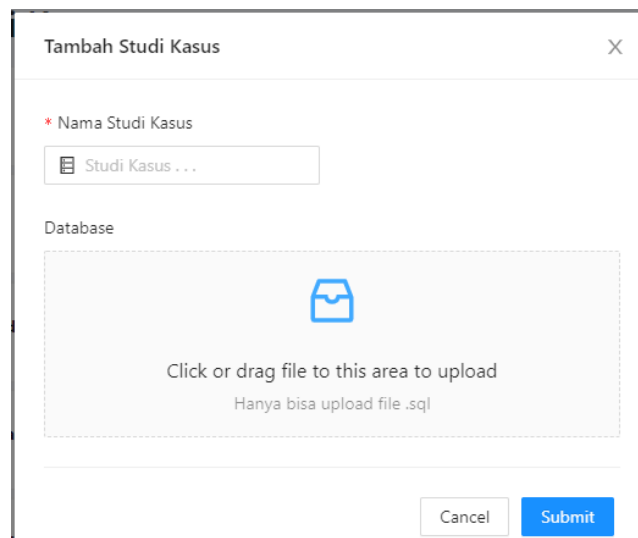
a) Melihat daftar studi kasus



*Gambar 22 Daftar Studi Kasus*


b) Membuat Studi Kasus baru

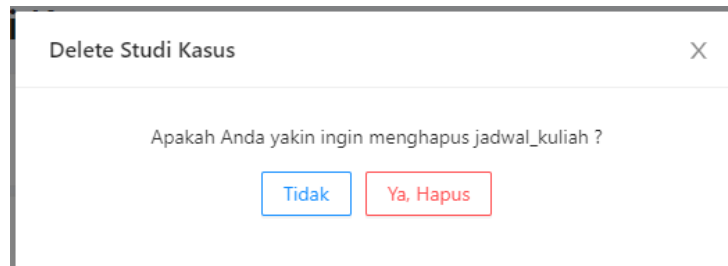
Jika ingin membuat studi kasus baru, dosen perlu klik tombol **Tambah Studi Kasus**, kemudian perlu mengisi informasi nama studi kasus dan database yang akan digunakan dalam format file .sql



*Gambar 23 Tambah Studi Kasus*


c) Menghapus Studi Kasus

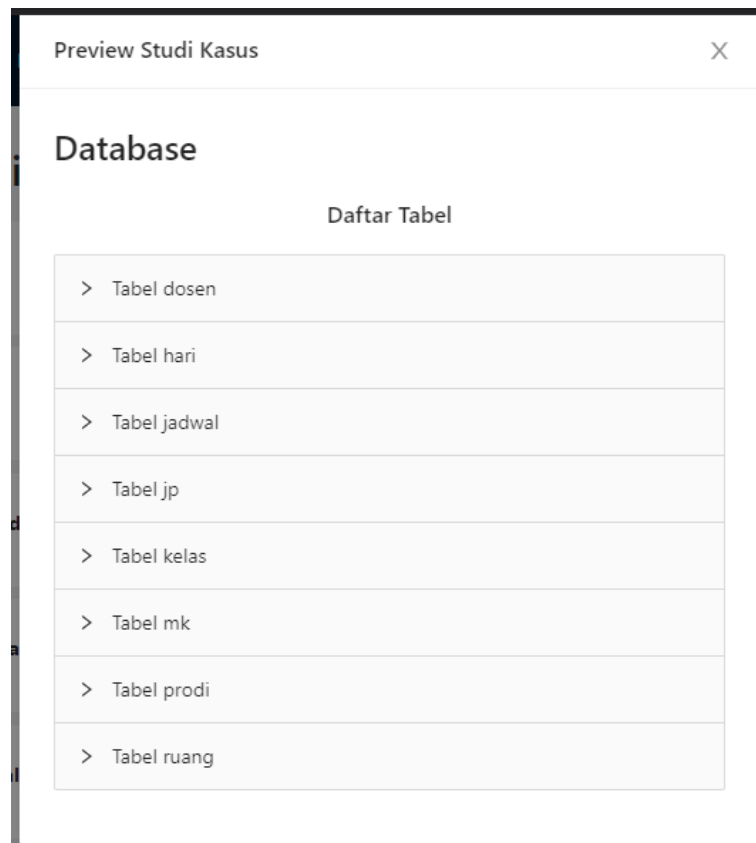
Untuk menghapus studi kelas dapat menekan tombol  yang ada pada setiap kelas. Kemudian akan muncul modal beserta konfirmasi apakah ingin menghapus, tekan ya, hapus untuk menghapus jadwal.



*Gambar 24 Menghapus Studi Kasus*

d) Preview Studi Kasus

Setelah membuat studi kasus, dosen dapat melihat detail dari studi kasus tersebut melalui fitur preview. Modal ini dapat diakses dengan klik tombol  di halaman studi kasus. Informasi yang dapat diberikan adalah detail tabel, kolom yang tersedia, serta isi dari tabel tersebut.



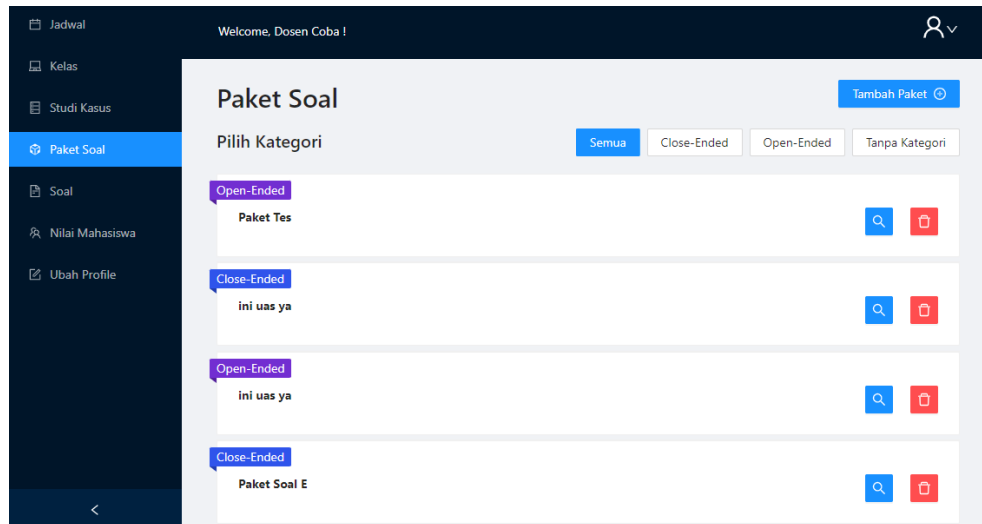
*Gambar 25 Preview Studi Kasus*



#### 2.2.4. Paket Soal

Fitur paket soal digunakan untuk memudahkan dosen mengumpulkan soal yang akan digunakan dalam sebuah jadwal latihan. Pada halaman ini, terdapat fitur sebagai berikut:


a) Melihat daftar paket soal



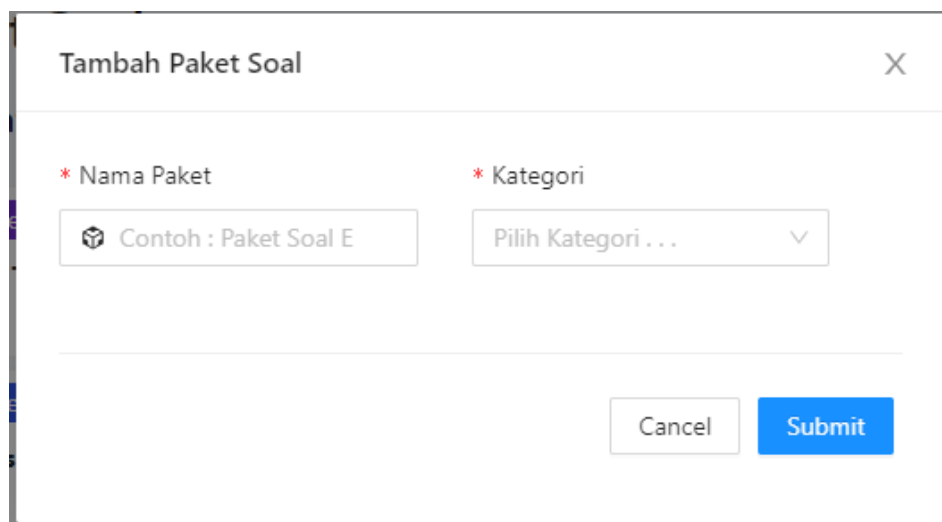
*Gambar 26 Daftar Paket Soal*

b) Menambahkan paket soal baru


Untuk menambahkan paket soal baru, dosen perlu klik tombol

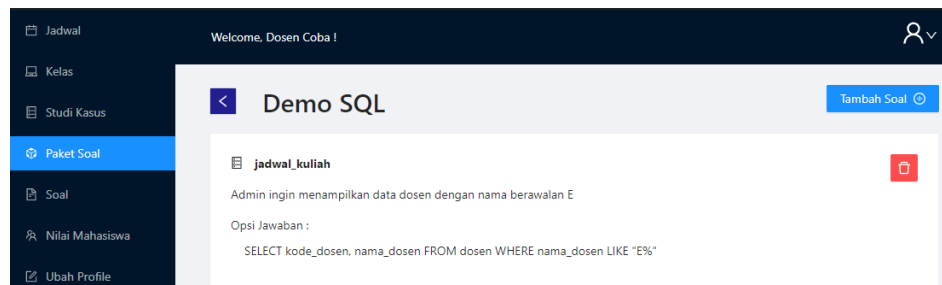
**Tambah Paket** 

Setelah itu, dosen perlu mengisi informasi nama paket, dan memilih kategori baik itu close-ended maupun open-ended.

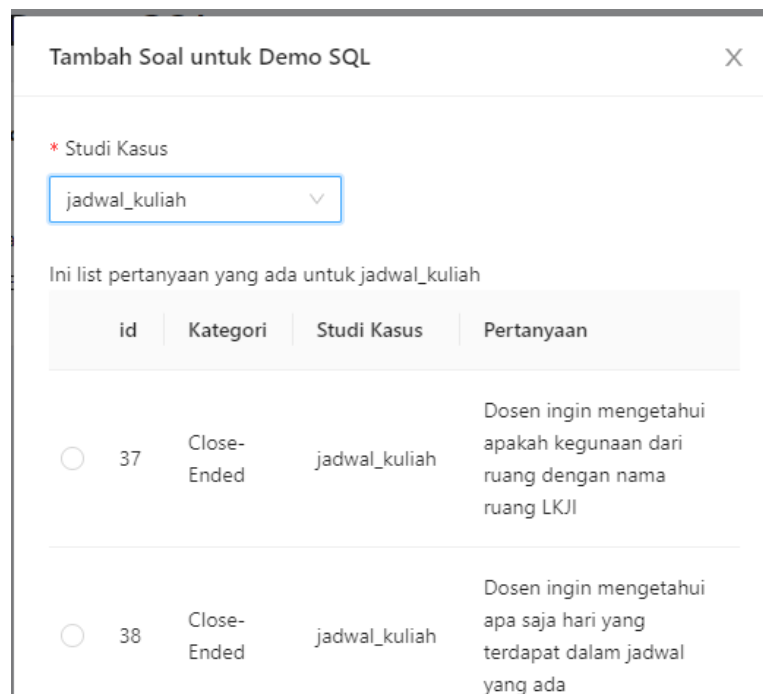


c) Preview paket soal

Setelah membuat paket, dosen dapat melihat detail dari studi kasus tersebut melalui fitur preview. Modal ini dapat diakses dengan klik tombol  di halaman paket. Di halaman ini, dosen diperlihatkan soal apa saja yang terdaftar dalam paket soal tersebut.




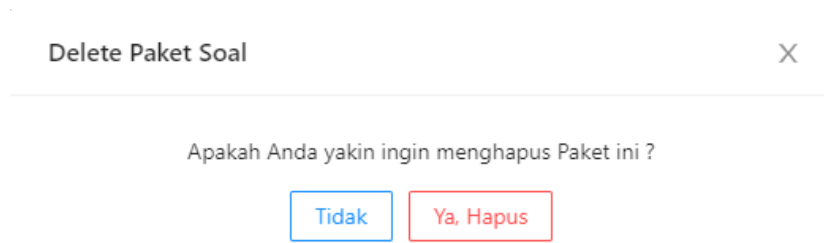
Gambar 27 Preview Paket Soal Close-Ended



Gambar 28 Tambah Soal ke Paket Soal Close-Ended

d) Menghapus paket soal

Untuk menghapus paket soal dapat menekan tombol  yang ada pada setiap paket. Kemudian akan muncul modal beserta konfirmasi apakah ingin menghapus, tekan ya, hapus untuk menghapus jadwal.

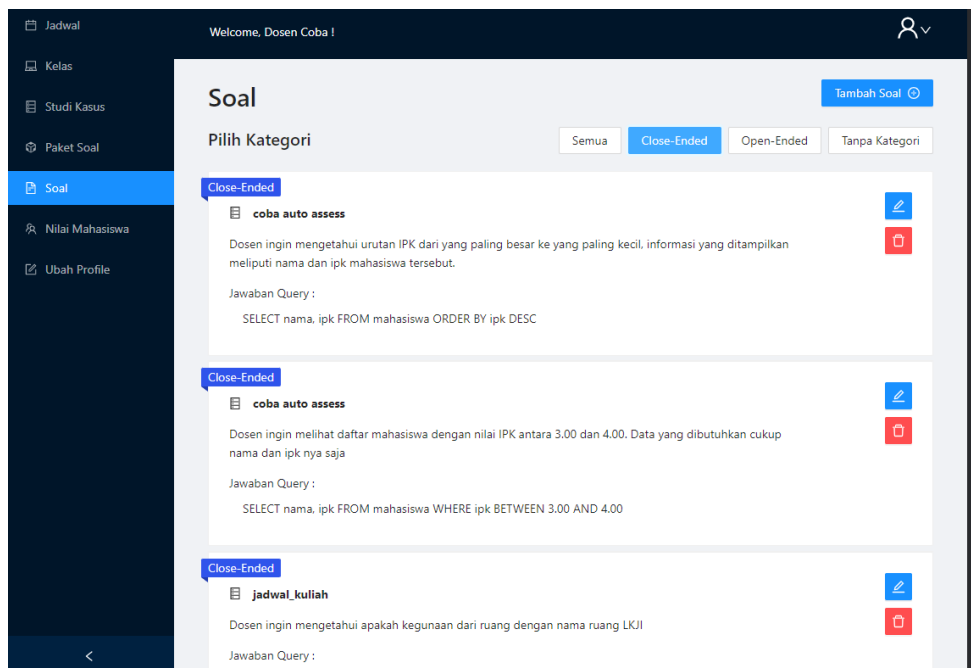


*Gambar 29 Modal Hapus Paket Soal*

### 2.2.5. Soal

Dosen dapat membuat soal sebagai latihan siswa. Aplikasi SQLearn memberikan opsi bagi dosen untuk membuat soal dengan cara open-ended maupun close-ended. Untuk memberikan latihan SQL dengan drag-and-drop, dosen dapat memilih untuk menambahkan soal dengan kategori close-ended.

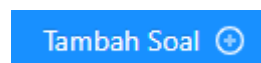
#### a) Melihat Daftar Soal yang ada



*Gambar 30 Daftar Soal Close-Ended*

#### b) Membuat Soal baru

Untuk membuat soal baru, dosen perlu klik tombol



Setelah modal terbuka, maka dosen perlu untuk mengisi informasi sebagai berikut:

##### 1. Teks soal


2. Kategori soal -> Pilih close-ended
3. Komponen SQL. Inilah komponen yang akan di drag-and-drop oleh mahasiswa ketika pengerjaan soal
4. Petunjuk SQL. Komponen yang menjadi petunjuk bagi siswa ketika menjawab soal ini
5. Jawaban Benar. Dosen perlu menambahkan opsi jawaban benar pula, karena dalam bagian komponen SQL dosen dapat memberikan opsi jawaban yang salah
6. Preview Hasil. Dosen perlu memasukkan hasil tabel yang diharapkan dalam querying. Ini akan menjadi petunjuk bagi siswa dalam mengerjakan soal.
7. Studi Kasus. Menghubungkan pertanyaan ini ke satu studi kasus yang telah dibuat oleh dosen
8. Tabel. Ada banyak tabel yang digunakan dalam 1 studi kasus, oleh karena itu perlu menambahkan tabel yang akan digunakan dalam soal ini.

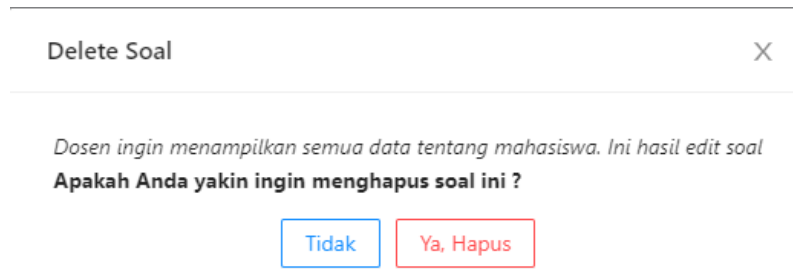
*Gambar 31 Pembuatan Soal Baru Close-Ended*

c) Mengubah Soal

Memiliki tampilan yang sama dengan tambah soal, namun digunakan untuk mengubah soal, komponen drag-and-drop, mengubah gambar, maupun mengubah studi kasus dan tabel yang digunakan dalam soal yang sudah ada.

d) Menghapus soal

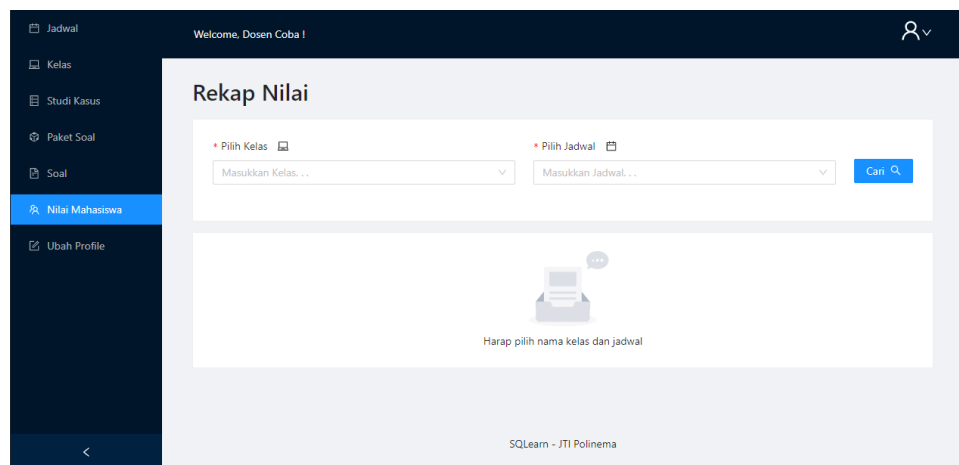
Untuk menghapus soal dapat menekan tombol  yang ada pada setiap soal. Kemudian akan muncul modal beserta konfirmasi apakah ingin menghapus, tekan ya, hapus untuk menghapus jadwal.



Gambar 32 Hapus Soal

## 2.2.6. Nilai Mahasiswa

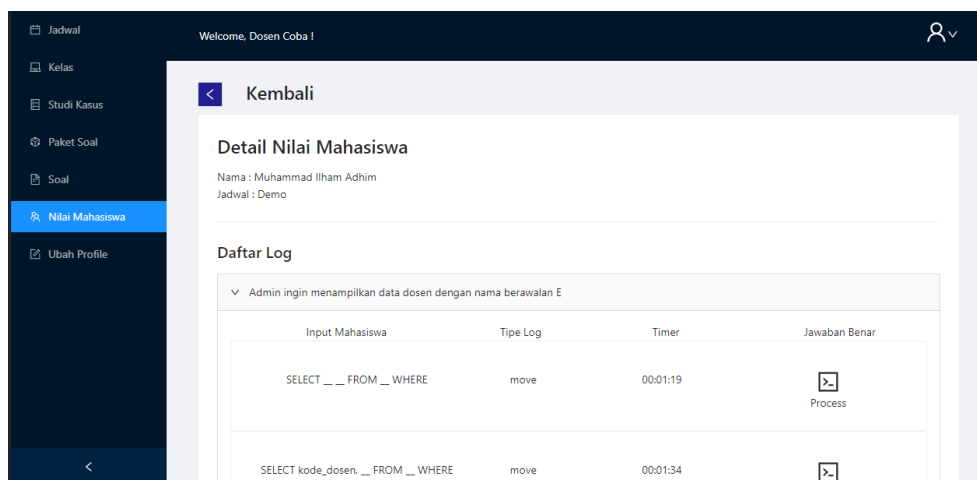
Dosen dapat melihat daftar nilai dari jadwal yang ada. Informasi yang perlu dimasukkan adalah nama kelas dan nama jadwal.



Gambar 33 Tampilan Daftar Nilai

Di halaman ini, dosen dapat melakukan fitur sebagai berikut:

- Melihat daftar nilai tiap mahasiswa untuk kelas dan jadwal yang dipilih
- Melihat detail nilai dan log aktivitas mahasiswa dari tiap soal yang telah dikerjakan.



Gambar 34 Log Aktivitas Mahasiswa

### 2.2.7. Ubah Profile

Dosen dapat merubah password sesuai dengan keinginan pada halaman ubah profile, dimana untuk mengubah password harus memasukkan password sekarang, kemudian masukkan password baru dan konfirmasi password baru, jika sudah klik submit.

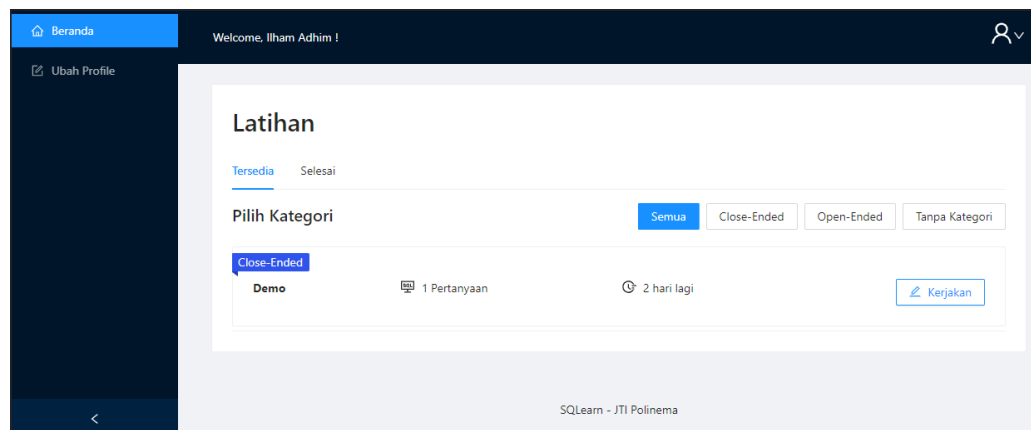
## 2.3. Role Mahasiswa

### 2.3.1. Jadwal

Setelah berhasil login sebagai mahasiswa maka halaman yang pertama kali dimuat adalah halaman jadwal, pada halaman ini terdapat 2-tab page yaitu:

#### a. Tersedia

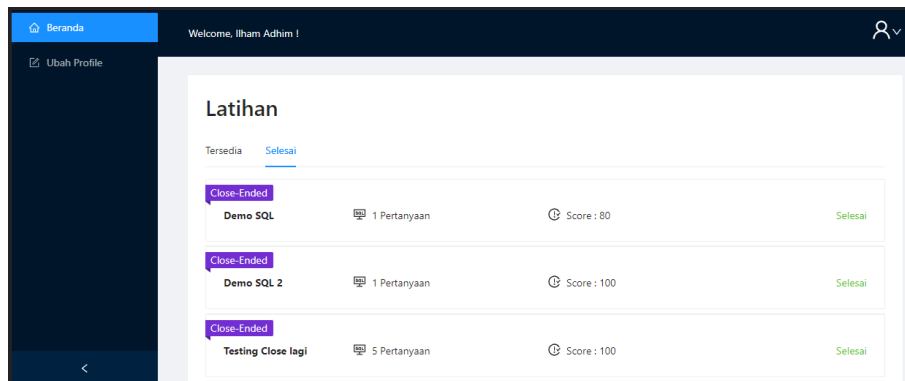
mahasiswa dapat melihat daftar jadwal yang dapat dikerjakan dapat melakukan filter jadwal yang tersedia,



*Gambar 35 Jadwal Tersedia*

#### b. Selesai

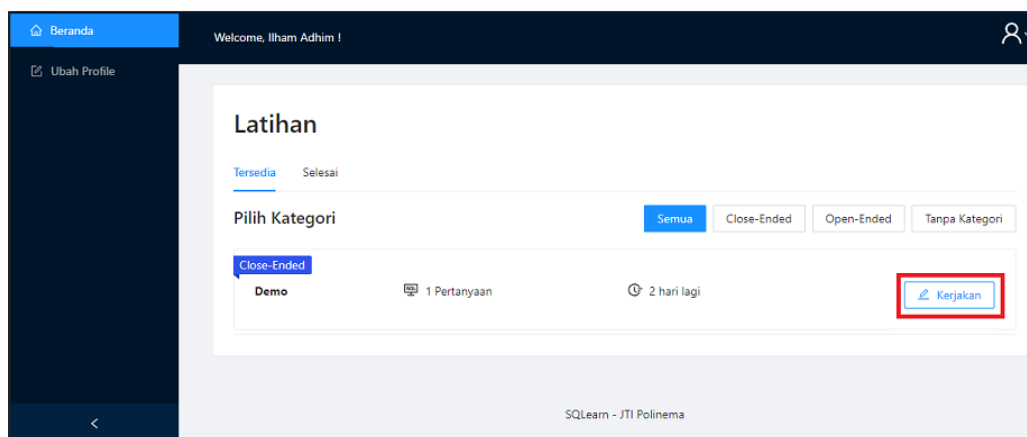
Melihat daftar jadwal yang telah dikerjakan serta melihat nilai yang didapat



*Gambar 36 Jadwal Selesai*

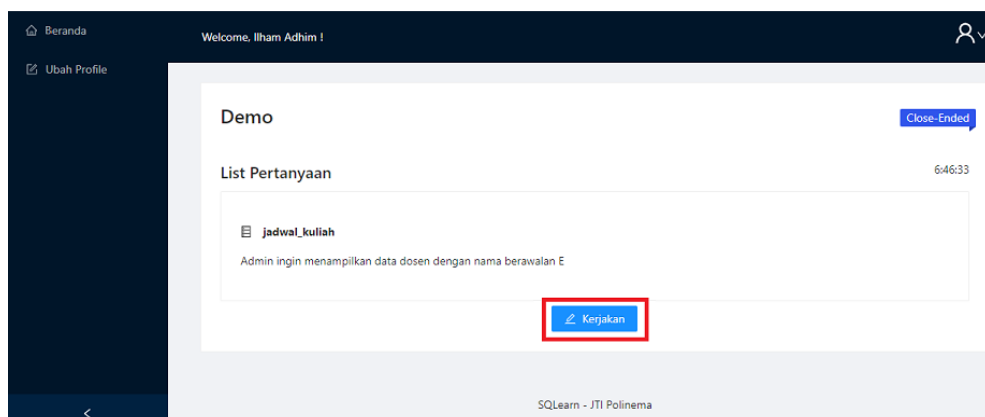
### 2.3.2. Pengerjaan Soal

Untuk dapat melakukan pengerjaan soal dapat menekan tombol kerjakan



*Gambar 37 Kerjakan Jadwal*

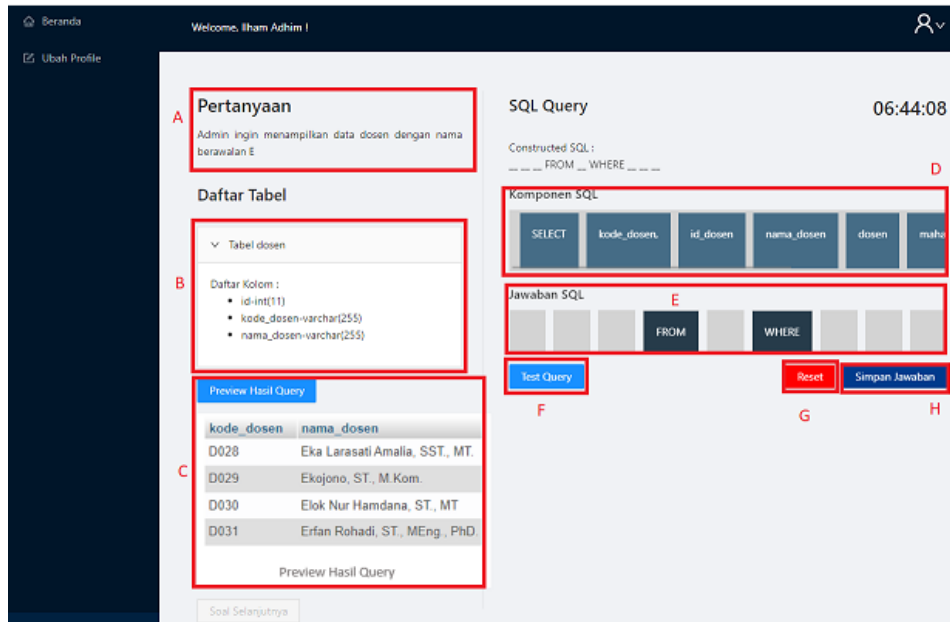
Kemudian akan muncul halaman daftar soal, untuk memulai pengerjaan dapat menekan tombol kerjakan.



*Gambar 38 Konfirmasi Kerjakan*

Akan muncul tampilan pengerjaan soal seperti berikut:

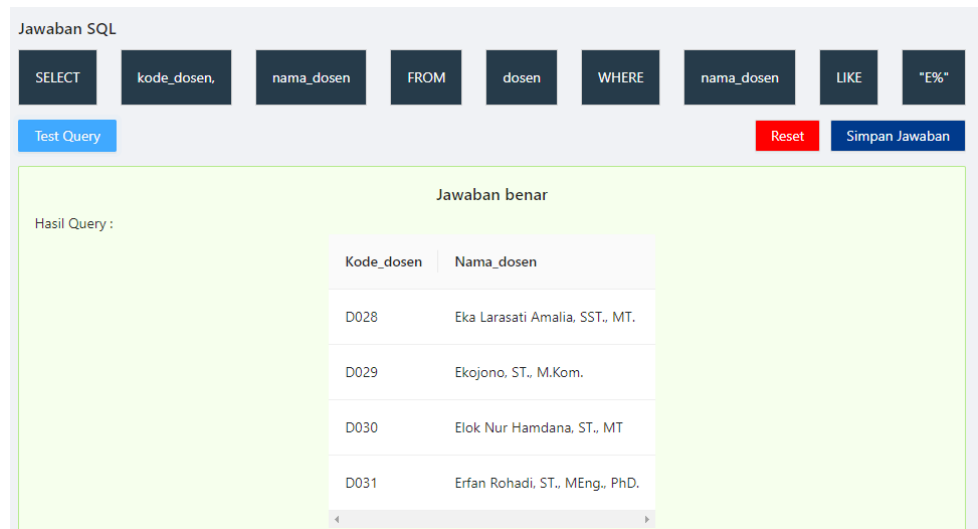




*Gambar 39 Pengerjaan Soal*

Pada gambar diatas terdapat kotak merah dengan huruf yaitu:

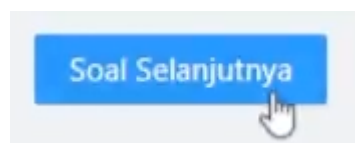
- Adalah tempat soal dari pertanyaan
- Adalah petunjuk mengenai daftar tabel dan kolom yang kemungkinan digunakan untuk menjawab soal
- Adalah foto hasil query yang diinginkan, dimana secara default belum ditampilkan, untuk ditampilkan dapat menekan tombol Preview Sub Query.
- Adalah komponen SQL yang dapat di drag oleh mahasiswa. Mahasiswa diharapkan teliti dan menggunakan petunjuk yang diberikan, karena ada kemungkinan untuk adanya komponen SQL yang salah.
- Drop area untuk Komponen SQL yang akan disusun di balik layer untuk rekonstruksi query SQL.
- Untuk melakukan test query jawaban dapat menekan tombol Test Query, dimana nanti akan muncul hasil dari query, jika benar maka akan muncul kotak hijau serta tabel hasil query yang didapatkan



*Gambar 40 Hasil Query Jawaban Benar*

- g) Jika ada kesalahan dalam drag-and-drop komponen SQL, siswa dapat menekan tombol reset. Kemudian komponen SQL dan drop area akan kembali seperti semula dan siswa diharuskan untuk menyusun komponen tersebut dari awal untuk soal tersebut.
- h) Sebelum dapat melanjutkan ke soal selanjutnya jika sudah yakin dengan jawaban harus menyimpan jawaban dengan klik tombol simpan jawaban.

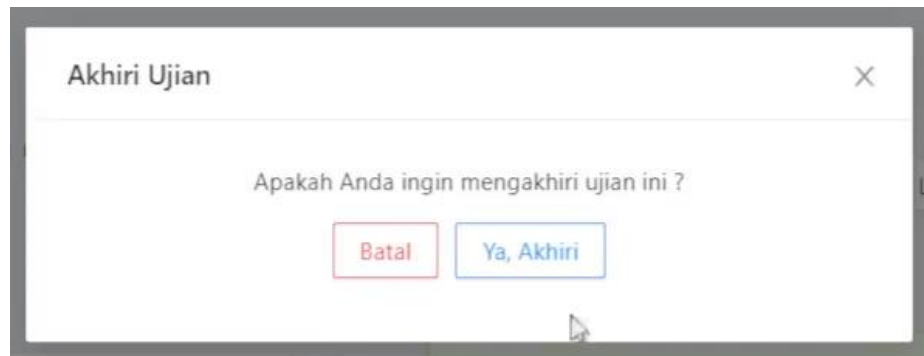
Setelah menekan tombol Simpan Jawaban maka dapat melanjutkan ke soal selanjutnya dengan menekan tombol Soal Selanjutnya yang berubah warna menjadi warna biru



Jika sudah semua soal terjawab maka tombol soal selanjutnya akan menjadi akhiri ujian,

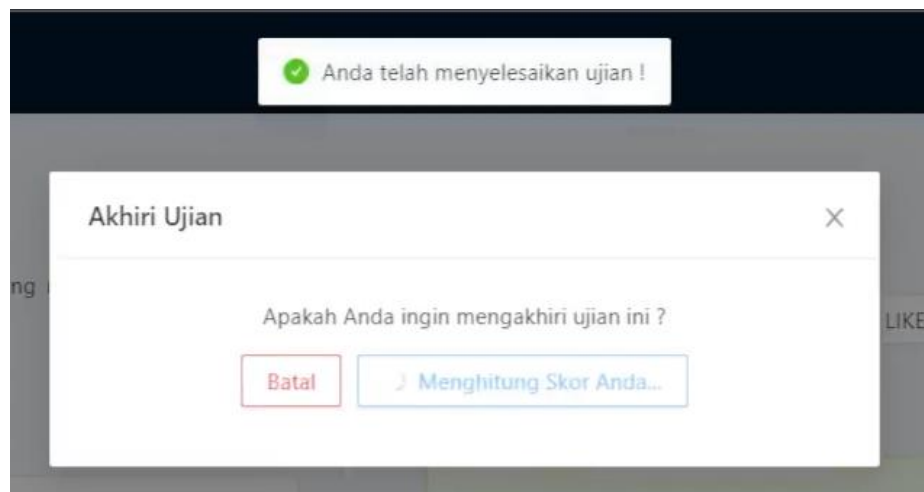


Jika diklik tombol akhiri ujian akan muncul pop up untuk akhiri ujian, klik Ya, Akhiri untuk mengakhiri ujian dan menghitung nilai yang didapatkan



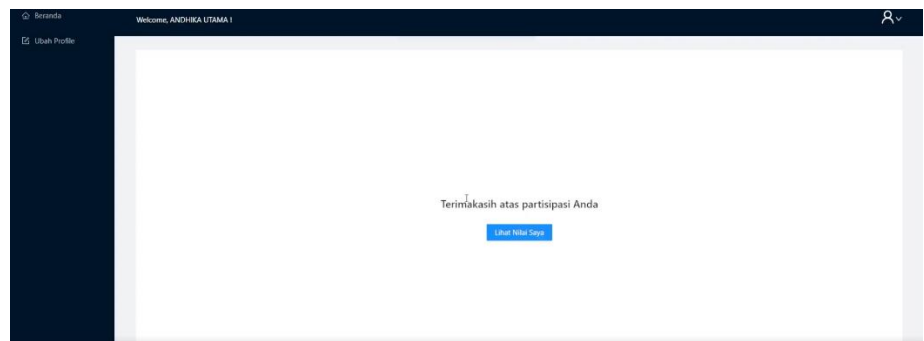
*Gambar 41 Modal Akhir Ujian*

Modal akan berubah dengan munculnya notifikasi diatas yang artinya sedang proses penilaian



*Gambar 42 Modal Loading Penilaian*

Jika penilaian telah selesai dilakukan maka akan diarahkan ke halaman akhir, untuk melihat nilai dapat menekan tombol Lihat Nilai Saya.

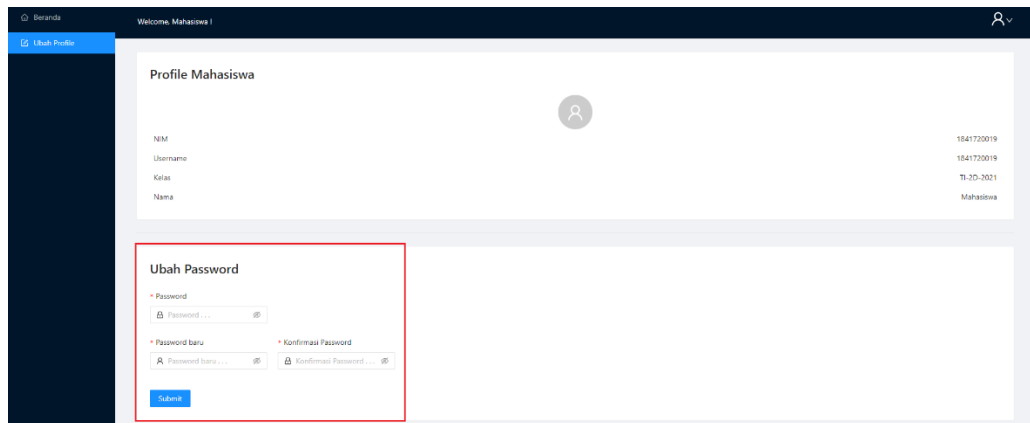


*Gambar 43 Pengerjaan selesai*

### 2.3.3. Ubah Profile

Mahasiswa dapat merubah password sesuai dengan keinginan pada halaman ubah profile, dimana untuk mengubah password harus memasukkan

password sekarang, kemudian masukkan password baru dan konfirmasi password baru, jika sudah klik submit.

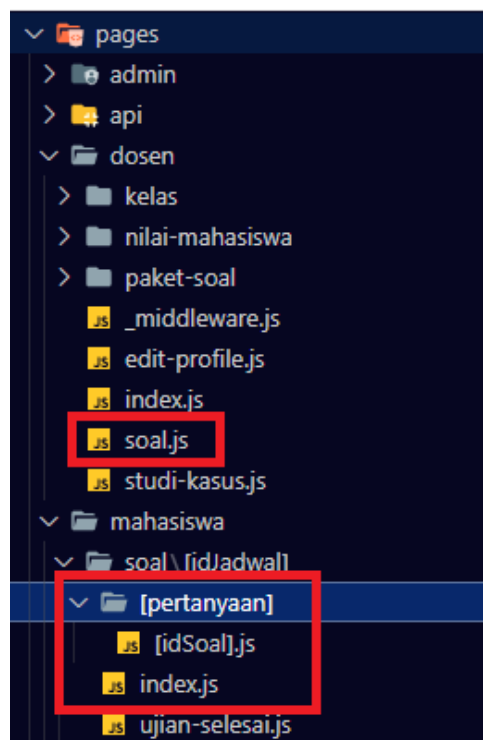


Gambar 44 Ubah Profile Mahasiswa

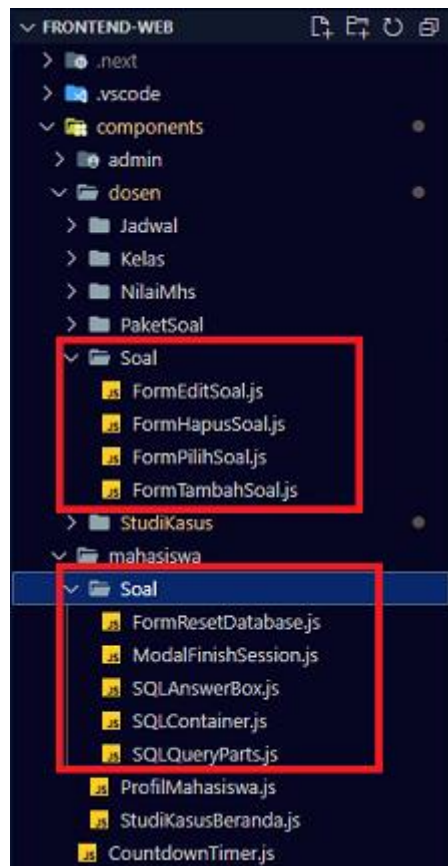
### 3. Source Code

Kode program yang digunakan pada pembuatan SQLearn ini dibuat menggunakan Visual Studio Code dengan Bahasa pemrograman Javascript. Berikut ini screenshot bagian utama kode program yang digunakan untuk menampilkan halaman pembuatan soal dengan pemberian petunjuk SQL Drag-and-Drop dari sisi dosen dan halaman pengerjaan soal bagi mahasiswa.

Berikut adalah file yang digunakan untuk membentuk halaman pembuatan soal dan pengerjaan soal secara keseluruhan:



Sementara itu, untuk komponen yang digunakan dalam membuat fungsionalitas dan bagian tampilan dari halaman tersebut adalah sebagai berikut:



```
pages\dosen\soal.js
```

```
import { React, useCallback, useEffect, useState } from "react";

import Head from "next/head";
import PageLayout from "../../components/PageLayout";

import { Typography, Row, Col, Button, message } from "antd";

import { PlusCircleOutlined } from "@ant-design/icons";

import ModalCustom from "../../components/Modal";
import FormEditSoal from "../../components/dosen/Soal/FormEditSoal";
import FormHapusSoal from "../../components/dosen/Soal/FormHapusSoal";
import {
  deleteSoal,
  getSoal,
  postSoal,
  updateSoal,
} from "../../utils/remote-data/dosen/SoalCRUD";
import ListComponent from "../../components/List";
```

```

import FormTambahSoal from "../../components/dosen/Soal/FormTambahSoal";
import RadioFilterCategory from "../../components/RadioFilterCategory";
import { useSession } from "next-auth/react";

function HalamanSoal() {
  const { data: session } = useSession();

  const [currentSoal, setCurrentSoal] = useState({});

  const [dataSoal, setDataSoal] = useState([]);
  const [soalFiltered, setSoalFiltered] = useState([]);

  const [isDataLoaded, setIsDataLoaded] = useState(false);
  const [isFilterActive, setIsFilterActive] = useState(false);

  const [isModalVisible, setIsModalVisible] = useState(false);
  const [modalRole, setModalRole] = useState("");

  useEffect(() => {
    fetchDataSoal();
  }, [session]);

  const fetchDataSoal = useCallback(() => {
    if (session !== undefined)
      getSoal(session?.user?.tokenJWT).then((res) => {
        setDataSoal(res.data);
        setIsDataLoaded(true);
        // formatToArray(res.data[0].answer);
      });
  }, [session]);

  const handleToggleModal = (state = isModalVisible) =>
    setIsModalVisible((prev) => state || !prev);

  const displayModalTambahSoal = () => {
    setModalRole("tambah");
    handleToggleModal();
  };

  const displayModalEditSoal = (soalObj) => {
    setModalRole("edit");
    setCurrentSoal(soalObj);
    handleToggleModal();
  };
};

```

```

const displayModalDeleteSoal = (soalObj) => {
  setModalRole("delete");
  setCurrentSoal(soalObj);
  handleToggleModal();
};

const aksiTambahSoal = (formSoal) => {
  // ? Close-ended udah bisa
  postSoal(session?.user?.tokenJWT, formSoal)
    .then(() => {
      handleToggleModal(false);
      message.success(`Data Pertanyaan berhasil ditambahkan`);
    })
    .then(() => fetchDataSoal())
    .catch((err) => message.error(`Data Pertanyaan gagal ditambahkan`));
};

const aksiEditSoal = (formSoal) => {
  updateSoal(session?.user?.tokenJWT, formSoal)
    .then(() => {
      handleToggleModal(false);
      message.success(`Data Pertanyaan berhasil diubah`);
    })
    .then(() => fetchDataSoal())
    .catch((err) => message.error(`Data Pertanyaan gagal diubah`));
};

const aksiDeleteSoal = (formSoal) => {
  deleteSoal(session?.user?.tokenJWT, formSoal.id)
    .then(() => {
      handleToggleModal(false);
      message.success(`Data Pertanyaan berhasil dihapus`);
    })
    .then(() => fetchDataSoal())
    .catch((err) => message.error(`Data Pertanyaan gagal dihapus`));
};

return (
  <>
    <Head>
      <title>SQLearn | Dosen - Soal </title>
    </Head>
    <PageLayout role="dosen">
      <Row justify="space-between">
        <Col>

```

```

        <Typography.Title level={2}>Soal </Typography.Title>
    </Col>
    <Col>
        <Button type="primary" onClick={displayModalTambahSoal}>
            Tambah Soal <PlusCircleOutlined />
        </Button>
    </Col>
</Row>

{isModalVisible && (
    <ModalCustom
        role={modalRole}
        entity="Soal"
        visible={isModalVisible}
        setVisible={setIsModalVisible}
        modalContent={
            modalRole === "tambah" ? (
                <FormTambahSoal
                    handleSubmit={aksiTambahSoal}
                    setVisible={setIsModalVisible}
                />
            ) : modalRole === "edit" ? (
                <FormEditSoal
                    handleSubmit={aksiEditSoal}
                    setVisible={setIsModalVisible}
                    currentSoal={currentSoal}
                />
            ) : (
                <FormHapusSoal
                    handleSubmit={aksiDeleteSoal}
                    setVisible={setIsModalVisible}
                    currentSoal={currentSoal}
                />
            )
        }
    />
)}

<RadioFilterCategory
    data={dataSoal}
    setIsFilterActive={setIsFilterActive}
    setEntityFiltered={setSoalFiltered}
/>

{/* Content asli... */}

```



```

        <ListComponent
          showDetail
          isLoading={!isLoading}
          dataSource={isFilterActive ? soalFiltered : dataSoal}
          role={"data-soal-dosen"}
          displayModalEditSoal={displayModalEditSoal}
          displayModalDeleteSoal={displayModalDeleteSoal}
        />
      </PageLayout>
    </>
  );
}

export default HalamanSoal;

```

```
pages\mahasiswa\soal\[idJadwal]\[pertanyaan]\[idSoal].js
```

```

import { React, useState, useEffect, useCallback } from "react";
import { useRouter } from "next/router";
import Head from "next/head";

import {
  Row,
  Col,
  Skeleton,
  Typography,
  Button,
  Divider,
  Form,
  Input,
  Alert,
  Table,
  Collapse,
  message,
} from "antd";

import { ConsoleSqlOutlined } from "@ant-design/icons";

```

```

import PageLayout from "../../../../../components/PageLayout";
import SQLContainer from "../../../../../components/mahasiswa/Soal/SQLContainer";
import CountdownTimer from "../../../../../components/CountdownTimer";
import { getSoalByID } from "../../../../../utils/remote-data/dosen/SoalCRUD";
import { useSession } from "next-auth/react";
import { URL_IMAGES } from "../../../../../utils/remote-data/api";
import { removeHTML } from "../../../../../utils/common";
import ModalCustom from "../../../../../components/Modal";
import
                FormResetDatabase
                from
"../../../../../components/mahasiswa/Soal/FormResetDatabase";

import useBoxes from "../../../../../utils/hooks/PengerjaanSoal/useBoxes";
import useNextQuestion from "../../../../../utils/hooks/PengerjaanSoal/useNextQuestion";
import useLogs from "../../../../../utils/hooks/PengerjaanSoal/useLogs";
import useInitializeTimer from "../../../../../utils/hooks/PengerjaanSoal/useInitializeTimer";
import
                ModalFinishSession
                from
"../../../../../components/mahasiswa/Soal/ModalFinishSession";
import useCountUpTimer from "../../../../../utils/hooks/PengerjaanSoal/useCountUpTimer";
import { finishSession } from "../../../../../utils/remote-data/mahasiswa/Session";

function LatihanSoal() {
  const router = useRouter();

  const { data: session } = useSession();
  const [timerUp] = useCountUpTimer();

  const [availableTableFromQuestion, setAvailableTableFromQuestion] = useState(
    []
  );

  // Ketika akhiri sesi
  const [isButtonLoading, setIsButtonLoading] = useState(false);

```

```
const [currentTables, setCurrentTables] = useState([]);
const [currentColumns, setCurrentColumns] = useState([]);
const [isFeedbackDisplayed, setIsFeedbackDisplayed] = useState(false);
const [feedbackContent, setFeedbackContent] = useState({ });

const [dataPertanyaan, setDataPertanyaan] = useState([]);
const [currentPart, setCurrentPart] = useState(null);
const [isDataPertanyaanLoaded, setIsDataPertanyaanLoaded] = useState(false);

const [isPreviewTable, setIsPreviewTable] = useState(false);

const [isModalVisible, setIsModalVisible] = useState(false);
const [modalRole, setModalRole] = useState("");

const handleToggleModal = () => setIsModalVisible((prev) => !prev);

const [isTimeLoaded, scheduleDate, timerLeftCounter, setTimerLeftCounter] =
  useInitializeTimer(isDataPertanyaanLoaded, dataPertanyaan);

const [boxes, setBoxes, resetBox] = useBoxes(
  isDataPertanyaanLoaded,
  dataPertanyaan
);

const [logData, saveLog, resetLog] = useLogs(
  boxes,
  dataPertanyaan,
  timerLeftCounter,
  timerUp,
  // scheduleDate,
  setCurrentPart
);
```

```
const [
  dataNextPertanyaan,
  isAnswerSaved,
  setIsAnswerSaved,
  setIsTestingQuery,
] = useNextQuestion(
  logData,
  resetLog,
  setCurrentTables,
  setCurrentColumns,
  setIsFeedbackDisplayed,
  setFeedbackContent
);

// ? Untuk simpan jawaban kalau soal ini berQuestionLabel?.name open-ended
const [form] = Form.useForm();

const fetchDataPertanyaan = useCallback(() => {
  if (session !== undefined && router?.query?.idSoal !== undefined) {
    getSoalByID(session?.user?.tokenJWT, router?.query?.idSoal).then(
      (response) => {
        setIsAnswerSaved(false);
        setBoxes([]);
        form.setFieldsValue({ jawaban_siswa: "" });
        console.clear();
        setDataPertanyaan(response.data);
        setIsDataPertanyaanLoaded(true);

        // Ini reset daftar tabel ketika ganti soal
        setAvailableTableFromQuestion([]);

        // Ini reset tabel dan kolom untuk feedback ketika test query
      }
    );
  }
});
```

```

setCurrentTables([]);
setCurrentColumns([]);
setIsFeedbackDisplayed(false);

// Reassign daftar tabel yang ada di soal
Object.keys(response.data?.tables).map((item) =>
  setAvailableTableFromQuestion((prev, id) => [
    ...prev,
    {
      id: id,
      table: item,
      columns: response.data?.tables[item] || [],
    },
  ])
);
}
);
}
}, [session, router.query.idSoal]);

useEffect(() => {
  fetchDataPertanyaan();
}, [fetchDataPertanyaan]);

useEffect(() => {
  // ? Save Log ketika ada pergerakan komponen drag-and-drop
  if (currentPart !== null) saveLog(currentPart, "move");
}, [boxes, currentPart]);

const previewTable = () => setIsPreviewTable((prev) => !prev);

const submitAnswer = (values) => {
  saveLog(values, "submit");
}

```

```

setIsTestingQuery(true);
};

const testQuery = (values) => {
  saveLog(values, "test");
  setIsTestingQuery(true);
};

const onClickButtonNext = () => {
  resetLog();

  if (dataNextPertanyaan !== null)
    router.push(
      `~/mahasiswa/soal/${parseInt(router.query.idJadwal)}/pertanyaan/${
        dataNextPertanyaan?.id
      }?session_id=${router.query?.session_id}`
    );
  else {
    setModalRole("akhiri");
    setIsModalVisible(true);
  }
};

const handleAkhiriSesi = () => {
  setIsButtonLoading(true);
  finishSession(session?.user?.tokenJWT, router.query?.session_id)
    .then((res) => {
      message.success("Anda telah menyelesaikan ujian !");
      setTimeout(() => {
        router.push(
          `~/mahasiswa/soal/${parseInt(router.query.idJadwal)}/ujian-selesai`
        );
      }, [3000]);
    });
};

```

```

    })
    .catch((err) => message.error("Mohon coba beberapa saat lagi"));
};

// ? If the timer is over, ya end the session
useEffect(() => {
  if (timerLeftCounter === "00:00:00") handleAkhiriSesi();
}, [timerLeftCounter]);

useEffect(() => {
  console.group("info LogData");
  console.log(logData);
  console.groupEnd();
}, [logData]);

return (
  <>
  <Head>
    <title>SQLearn | Mahasiswa - Soal </title>
  </Head>
  <PageLayout role="mahasiswa">
    <Row gutter={10} style={{ marginTop: "2em" }}>
      <Col lg={9} style={{ padding: "0 1em" }}>
        {isDataPertanyaanLoaded ? (
          <>
            <Typography.Title level={3}>Pertanyaan</Typography.Title>
            <Typography.Paragraph
              style={{
                marginBottom: "2em",
                textAlign: "justify",
              }}
            >
              {removeHTML(dataPertanyaan?.text) || dataPertanyaan?.text}
          </>
        ) : null}
      </Col>
    </Row>
  </PageLayout>
  </>
)

```

```

</Typography.Paragraph>
</>
): (
  <Skeleton
    active
    paragraph={false}
    title={{ width: "20vw" }}
    style={{ marginBottom: "1em" }}
  />
)}

<Row>
  { " " }
  <h2> Daftar Tabel</h2>{ " " }
</Row>
<Row
  style={{
    margin: "1em 0",
  }}
>
  <Col span="24">
    {isDataPertanyaanLoaded &&
      availableTableFromQuestion?.map((item, idPanel) => (
        <Collapse key={idPanel}>
          <Collapse.Panel
            header={`Tabel ${item?.table}`}
            key={item.id}
          >
            Daftar Kolom :
            <ul>
              {item?.columns?.map((column, id) => (
                <li key={id}> {column}</li>
              ))}
            </ul>
          )}
        )}
      )}
    )}
  </Col>
</Row>

```



```

        </ul>
        </Collapse.Panel>
    </Collapse>
    )})
</Col>
</Row>

<Button type="primary" onClick={() => previewTable()}>
    Preview Hasil Query
</Button>
<Row
    style={{
        marginTop: "1em",
        display: isPreviewTable ? "block" : "none",
    }}
>
    {isDataPertanyaanLoaded && (
        <img
            width={380}
            src={URL_IMAGES + dataPertanyaan?.answer_pic}
            alt={`${URL_IMAGES}/${dataPertanyaan?.answer_pic}`}
        />
    )}
</Row>
<Row style={{ marginTop: "1em" }}>
    <Button
        disabled={isAnswerSaved ? false : true}
        type="primary"
        onClick={onClickButtonNext}
    >
        {dataNextPertanyaan !== null
            ? "Soal Selanjutnya"
            : "Akhiri Ujian"}

```

```

</Button>
</Row>
</Col>
<Col lg={ 1 }>
  <Divider type="vertical" style={{ height: "80vh" }} />
</Col>
<Col lg={ 14 }>
  <Row justify="space-between">
    <Col>
      <h2 className="title-part">SQL Query</h2>
    </Col>
    <Col style={{ textAlign: "right" }}>
      <h2 className="title-part">
        {isTimeLoaded ? (
          <CountdownTimer
            expiryTimestamp={scheduleDate}
            // timerStartCounter={timerStartCounter}
            // setTimerStartCounter={setTimerStartCounter}
            setTimerLeft={setTimerLeftCounter}
          />
        ) : (
          "Loading. . ."
        )}
      </h2>
    </Col>
  </Row>
  {dataPertanyaan?.QuestionLabel?.name === "Open-Ended" ? (
    <>
      <Form form={form} layout="vertical">
        <Form.Item name="jawaban_siswa">
          <Input
            autoComplete="off"
            prefix={<ConsoleSqlOutlined />}

```

```

        placeholder={`Masukkan Jawaban SQL Disini . . .`}
      />
    </Form.Item>
  </Form>
</>
): (
  <SQLContainer
    boxes={boxes}
    jawaban={dataPertanyaan?.jawaban_benar}
    setBoxes={setBoxes}
    setCurrentPart={setCurrentPart}
  />
)}
<Row style={{ marginTop: "1em" }} justify="space-between">
  <Col>
    <Row gutter={10}>
      <Col>
        <Button
          type="primary"
          onClick={() =>
            testQuery(
              dataPertanyaan?.QuestionLabel?.name === "Open-Ended"
                ? form.getFieldsValue()
                : ""
            )
          }
        >
          Test Query
        </Button>
      </Col>
      {dataPertanyaan?.QuestionLabel?.name === "Open-Ended" && (
        <Col>
          <Button

```



```

    )
  }
  htmlType="submit"
>
  Simpan Jawaban
</Button>
</Col>
</Row>
</Col>
</Row>
{isFeedbackDisplayed && (
  <Row style={{ marginTop: "1em" }}>
    <Col span={24}>
      <Alert
        type={feedbackContent?.data?.is_equal ? "success" : "error"}
        description={
          <
            <h1 style={{ textAlign: "center" }}>
              {feedbackContent?.data?.message}
            </h1>
            Hasil Query :
          <Row justify="center">
            <Table
              style={{
                overflowX: "scroll",
              }}
              columns={currentColumns}
              dataSource={currentTables}
              pagination={false}
            />
          </Row>{" "}
        </>
      }

```

```

        />
    </Col>
</Row>
)}
{isModalVisible && (
    <ModalCustom
        role={modalRole}
        entity={modalRole === "delete" ? "Database" : "Ujian"}
        visible={isModalVisible}
        setVisible={setIsModalVisible}
        modalContent={
            modalRole === "delete" ? (
                <FormResetDatabase setVisible={setIsModalVisible} />
            ) : (
                <ModalFinishSession
                    isButtonLoading={isButtonLoading}
                    handleSubmit={handleAkhiriSesi}
                    setVisible={setIsModalVisible}
                />
            )
        }
    />
)}
</Col>
</Row>
</PageLayout>
</>
);
}

export default LatihanSoal;

```

#### **4. Kesimpulan**

Aplikasi SQLearn dapat digunakan sebagai aplikasi latihan SQL yang interaktif bagi siswa dengan menggunakan pendekatan drag-and-drop dalam mata kuliah Basis Data.