IMPLEMENTATION OF SQL QUERY CONSTRUCTION TO IMPROVE DATABASE CONCEPT UNDERSTANDING WITH CLOSE-ENDED APPROACH

UNDERGRADUATE THESIS

Digunakan Sebagai Syarat Maju Ujian Diploma IV
Politeknik Negeri Malang

By:

MUHAMMAD ILHAM ADHIM NIM. 1841720076

INFORMATICS ENGINEERING STUDY PROGRAM
DEPARTMENT OF INFORMATION TECHNOLOGY
POLITEKNIK NEGERI MALANG
2022

LETTER OF APPROVAL

IMPLEMENTATION OF SQL QUERY CONSTRUCTION TO IMPROVE DATABASE CONCEPT UNDERSTANDING WITH CLOSE-ENDED APPROACH

Arranged By:

MUHAMMAD ILHAM ADHIM NIM. 1841720076

This undergraduate thesis has been tested in

21 Juni 2021

Approved by:

1.	Supervisor	:	Putra Prima Arhandi, S.T. M.Kom. NIP. 19840610 200812 1 004	....
2.	Supervisor	:	Muhammad Shulhan Khairy S.Kom., M.Kom NIP. 19920517 201903 1 020	....
3.	Penguji Utama	:	Usman Nurhasan, S.Kom., MT NIP. 198609232015041001	....
4.	Penguji Pendamping	:	Mustika Mentari, S.Kom., M.Kom NIP. 198806072019032016	....

Ascertain,

Head of Information Technology Department	Head of Informatics Engineering Study program
Rudy Ariyanto, S.T., M.Cs. NIP. 19711110 199903 1 002	Imam Fahrur Rozi, S.T., M.T. NIP. 19840610 200812 1 004

STATEMENTS

I hereby declare that in this thesis there is no work, either in whole or in part, that has been submitted for an academic degree at any university, and to the best of my knowledge there is no work or opinion that has been written or published by another person, except which are cited in writing in this manuscript and mentioned in the citation list/bibliography.

Malang, 21 June 2022

Muhammad Ilham Adhim.

ABSTRAK

Adhim, Muhammad Ilham. "Penerapan Konstruksi SQL Query dalam Peningkatan Pemahaman Konsep Database dengan Pendekatan Close-Ended". **Pembimbing: (1) Putra Prima A., ST., M.Kom., (2) Muhammad Shulhan Khairy S.Kom., M.Kom.**

Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2022.

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 (75.2000) lebih tinggi daripada rata-rata nilai pre-test (64.0000). 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, Wilcoxon Signed Ranks Test*

ABSTRACT

Adhim, Muhammad Ilham. “Implementation of SQL Query Construction to Improve Database Concept Understanding with Close-Ended Approach”. **Advisor: (1) Putra Prima A., ST., M. Kom., (2) Muhammad Shulhan Khairy S.Kom., M.Kom.**

Thesis, Informatics Management Study Program, Department of Information Technology, State Polytechnic of Malang, 2022.

Structured Query Language (SQL) is being used dominantly both in industry and academic fields, especially in Information Technology area, SQL uses declarative syntax to fetch data from database. To make an effective query when learning SQL Query, students are required to visualize various elements within the database concept and understand how to extract such data, which may lead to burdening cognitive load of the learner. This research purposes a provided close-ended SQL hints and using drag-and-drop SQL parts to help students practice SQL Query exercises regarding required tables, columns, and SQL syntax to construct valid SQL Query. Such method is being used in SQLearn, a web platform for students to exercise SQL Query, To observe the impact of such implementation to students understanding, 28 students were asked to participate in pre-test, SQLearn exercise, and post-test. The data then will be processed using Kolmogorov-Smirnov for normality test. The result for normality test with mentioned participants signifies that the data is not normally distributed. Therefore, the data is being processed with non-parametric test named Wilcoxon Signed Ranks Test to determine if there was an significant average difference in pre-test and post-test. The result shows that post-test average score (75.2000) is higher than pre-test average score (64.0000). In addition to that, the Asymp. value Sig. (2-tailed) is 0.00 (< 0.05). Therefore, it can be concluded that the such average improvement is proved being significant statistically. Hence, the usage of SQL Query Construction with drag-and-drop method alongside with close-ended approach towards students understanding of SQL Query SELECT has positive impact significantly.

Keywords: *SQL Query, MySQL, Drag-and-drop, Data Manipulation Language, Wilcoxon Signed Ranks Test*

KATA PENGANTAR

Puji Syukur kami panjatkan kehadirat Allah SWT/Tuhan YME atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul “IMPLEMENTATION OF SQL QUERY CONSTRUCTION TO IMPROVE DATABASE CONCEPT UNDERSTANDING WITH CLOSE-ENDED APPROACH”. Skripsi ini penulis susun sebagai persyaratan untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Kami menyadari bahwasannya dengan tanpa adanya dukungan dan kerja sama dari berbagai pihak, kegiatan laporan akhir ini tidak akan dapat berjalan baik. Untuk itu, kami ingin menyampaikan rasa terima kasih kepada:

1. Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
2. Kedua orang tua saya yang selalu memberikan dukungan moril dan materiil, motivasi dan juga doa agar saya dimudahkan dalam menyelesaikan skripsi ini dengan tepat waktu serta nilai yang baik.
3. Bapak Putra Prima Arhandi, S.T., M.Kom., selaku pembimbing I yang telah memberi bimbingan, arahan, dan meluangkan waktu kepada saya dalam menyusun skripsi ini dengan baik.
4. Bapak Muhammad Shulhan Khairy S.Kom., M.Kom**.** selaku pembimbing II yang telah memberi bimbingan, arahan, dan meluangkan waktu kepada saya dalam menyusun skripsi ini dengan baik.
5. Bapak Rudy Ariyanto, ST., M.Cs. selaku Ketua Jurusan Teknologi Informasi.
6. Bapak Imam Fahrur Rozi, ST., MT. selaku Ketua Prodi Teknik Informatika.
7. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan Laporan Akhir dari awal hingga akhir yang tidak dapat kami sebutkan satu persatu.

Penulis menyadari bahwa dalam penyusunan laporan akhir ini, masih banyak terdapat kekurangan dan kelemahan yang dimiliki penulis baik itu sistematika penulisan maupun penggunaan bahasa. Untuk itu penulis mengharapkan saran dan kritik dari berbagai pihak yang bersifat membangun demi penyempurnaan laporan ini. Semoga laporan ini berguna bagi pembaca secara umum dan penulis secara khusus. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, 21 June 2021

Researcher

TABLE OF CONTENTS

Page

FIGURE LIST

Page

TABLE LIST

ATTACHMENTS

CHAPTER I. INTRODUCTION

1.1. Background

Programming is a fundamental skill for science and technology (Ali et al., 2017). The awareness of programming implementation in multiple areas of expertise has made it a mandatory course in academic institutions (Fedorenko et al., 2019). By learning computer science, students are expected to increase their creativity and improve critical thinking (Jamil & Isiaq, 2019). Despite its benefits, learning to program has various challenges, such as a lack of motivation to comprehend and write the codes. In addition, the high dropout rates of studying computer science, especially in programming courses, has indicated the subject as one of the hardest courses to learn (Bosse & Gerosa, 2017)

Despite a high dropout rate in programming subjects, the implementation of databases is one of the valued skills in the informatics and computer science area (Puspitasari et al., 2019). Therefore, Database is a mandatory subject and becomes a core curriculums that computer science students required to take at the beginning of the semester (Jiandong et al., 2018; Shang, 2017). Learning and implementing databases is not limited to academic areas, as it affects industry operation in the process. The relevancies of databases in the industry lies in organization management, data entry, and effective data processing (Luthfi & Ayu, 2019). To increase understanding of programming courses, especially in databases, the approach that needs to be used should not only be limited to code generation, but needs to reflect the thinking process of the student, case study description, and problem-solving (Villamor, 2020). Database course may be given to 2nd year students (Mason et al., 2016) but the time may varies according to curriculum implemented in respective universities.

SQL Query Formulation can be a challenging task due to various factors, such as its declarative form of SQL syntax (Taipalus, 2019), the query complexity that is limited to short- term memory of code writer when retrieving data, this happens since formulating complex query requires higher working memory load (Sweller, 2020), and potential of ambiguity in retrieving data (Borthick et al., 2001).

Structured Query Language (SQL) is still being used dominantly both in industry and academic fields (Taipalus, 2019). Although it is widely implemented, the SQL learning process is quite rough for novices (Leinonen et al., 2020). To make an effective query, students are required to visualize various elements within the database concept and understand how to extract such data, which may lead to burdening cognitive load of the learner (Shin, 2020). Due to this reason, the SELECT statements becomes the main concept in learning SQL.

The common practice of teaching databases is mostly concerned with learning normalization techniques, SQL syntax, operating database management, and designing ERD (Sastry, 2015). Various researches to gain a better student's understanding of database courses has done before, such as using computational thinking as core of database-teaching model by (Huang & Leng, 2019), usage of hybrid learning theory and MOOC by (Shang, 2017), and implementation of iterative teaching on Database course by (Jiandong et al., 2018). All of mentioned research has proved that they impacted positively towards both student’s enthusiasm for learning database and the teaching method.

SQL code construction has been implemented in the medical system, with more focus on using database concepts to make a valid SQL Query (Gorskis, 2018). The approach done on Gorskis' research was class concepts, object property concepts and data property concepts. Similar research has been done by (Phewkum et al., 2019) in the E-Learning platform with a drag-and-drop concept. By having 30 participants contribute to it, they mentioned that the application gives new experience in learning SQL. Yet, based on the score, it is indicated that the application is not the easiest to use, and there is no further explanation on the impact of the application on students’ understanding of learning SQL. In addition, the code construction used is using an open-ended question format. As a result, even the syntax error could be minimized, students also need to consider the order of code blocks to get the correct answer.

As research develops, a close-ended approach has been widely used in data gathering. This is because of its ease of use and configuration for participants (Baburajan et al., 2021). In a learning platform, a close-ended approach can be implemented in the teaching process or examination. This can be seen from one close-ended approach that has predefined answers. It allows students to think logically about solving case studies within the appropriate context and scope of the problem (Lin & Lien, 2013).

Based on the mentioned background, the writer proposes research "Implementation of SQL Query Construction to Improve Database Course Understanding with Close-Ended Approach". As for the participants of this research, they will be from Politeknik Negeri Malang, and from its curriculum, the database course is being given to students in their 1st year. The application proposed will be used by these students who take the database course. With this research, it is expected that using a close-ended approach in the database subject may reflect students' thinking process and problem-solving skills by using SQL Code reconstruction may improve students' understanding of database concepts significantly.

1.2. Research Problem

Based on the mentioned background, the problem statement in this research relies on How is the effect of SQL Code construction with drag-and-drop and close-ended approach on students' understanding of SQL SELECT Statements concepts?

1.3. Research Scope

Based on the mentioned background, problem constraints in this research according to the scope of this research are:

- The application runs on a web platform
- Using MySQL database.
- SQL queries used for SELECT statements
- Grading and back-end process using NodeJs.
- The application will be tested by students in database classes and lecturers in Information Technology Major in State Polytechnic of Malang.

1.4. Objectives

This research aims to observe the effect of SQL code construction implementation with a drag-*and-drop* method and *close-ended* approach to students' understanding of the SQL SELECT Statements concepts.

1.5. Benefits

The benefits of this research are as follows:

- Researcher, to identify whether students who are involved in learning SQL using the *drag-and-drop* and *close-ended* approach have an increased understanding of the SQL SELECT Statements concepts.
- State Polytechnic of Malang, to make reference for further research. Especially for students that has interest on developing database learning.
- Students, can be used to enhance understanding and experience in learning database concepts, especially in SQL queries.

CHAPTER II. LITERATURE STUDY

2.1. Literature Study

There are several references to support this research, one of which is research on the role of database learning and its effect on student understanding in the field of Information Science, Drag-and-Drop concepts in SQL learning, and the use of SQL Construction in non-educational and educational fields.

Research conducted by (Sastry, 2015) entitled "An Effective Approach for Teaching Database" explained that the process of learning databases should be focused on learning normalization techniques, SQL queries understanding, operating database products, and designing a comprehensive database as a one-semester curriculum. The blocker is, the learning pattern is only limited to normalizations and designing the Entity-Relationship Diagram (ERD) from time to time. Based on the conclusion of this research, one of many factors that define students' understanding is using Problem-Based Learning in formulating SQL queries. This is because students need not do further research regarding a specific case study and focus more on giving the solution in SQL code into the system. Furthermore, flexibility when doing the assignment to get in-depth knowledge of SQL queries allows them to gain more insights on solutions available to solve the case. This allows students to discuss and collaborate and do their research on which solution suits the best based on the given case study.

The idea of focusing more on SELECT statements can be concluded based on the research done by (Faeskorn-Woyke et al., 2020). The research wanted to classify what is the most mistakes done by the students while learning SQL Queries. They had a system to monitor the process for seven months and gathered 7533 wrong statements as an input. This input will be treated as a training set. While the database being used is ORACLE instead of MySQL, it is still a reference that could reflect the idea which part of novice's mistake while learning SQL.

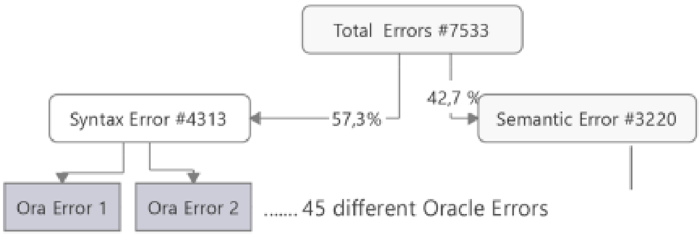
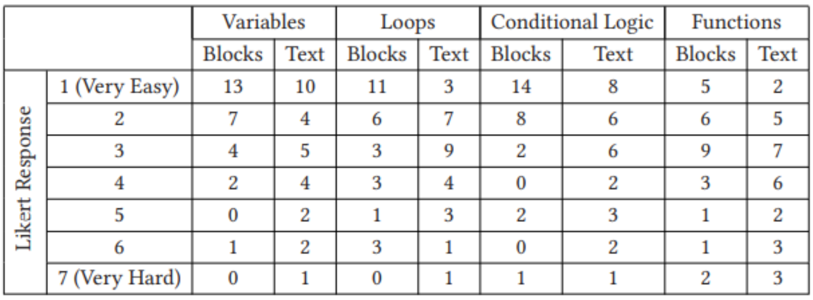


Figure 2. 1 Errors Distribution from (Woyke et al., 2020)

As stated from its journal "Most of the errors were misunderstandings of the data model or the correct structure of a simple SQL SELECT statement. Some of the mistakes result from the misunderstanding of the character of the relational database when a set is, for example, compared with a single value". And from its own research result in Q2 part that signifies the most errors were due to insufficient knowledge or understanding of the data model that includes table names, columns related, and fundamental SELECT Statement. Coming from this fact, we decided to conduct more on students' SELECT statements understanding in our research.

While lecturers creating case studies, it is recommended to make a problem and approach that uses industry standards. To deal with this, a research conducted by (Broatch et al., 2019) concludes that it can be done in some ways. First, the database case study is using widely implemented language in database field. Therefore, in this proposed research, the language used will be SQL. Secondly, when it comes to a relational database – as MySQL is being used in this research – students have to learn about common set operators, filters, and joins. Lastly, it is required for the students to formulate the query while having visual representation of the query result. In this proposed research, this is solved using a feature that enables the student to see the expected result of correct SQL Query, and the case study table preview. To summarize, the idea of giving problem accordance to industry condition, the lecturer can give the operators, filters and joins for the case study, whereas the other criterion mentioned has been solved by the chosen SQL type and feature provided in the system.

Implementation of drag and drop in learning programming context has done by (Weintrop & Wilensky, 2017) in their research. The research conducted for five weeks and use two classes at the same school that are categorized into two groups. The goal is to compare the outcomes of using text-based interface with block-based interface that can be interacted by dragging and dropping by the participants. Both groups are using same programming environment and curriculum. The assessment method to determine the difference was done by using pre- and post-content assessments along with attitudinal surveys and classroom observations.



		Variables		Loops		Conditional Logic		Functions	
		Blocks	Text	Blocks	Text	Blocks	Text	Blocks	Text
Likert Response	1 (Very Easy)	13	10	11	3	14	8	5	2
	2	7	4	6	7	8	6	6	5
	3	4	5	3	9	2	6	9	7
	4	2	4	3	4	0	2	3	6
	5	0	2	1	3	2	3	1	2
	6	1	2	3	1	0	2	1	3
	7 (Very Hard)	0	1	0	1	1	1	2	3

Table 2. 1 Distribution of Ease-of-Use Responses

As shown in the above table, while learning the algorithm of variables, loops, conditional logic, and functions, the group that performs drag and dropping the blocks stated that using very easy to use while answering across those 4 concepts.

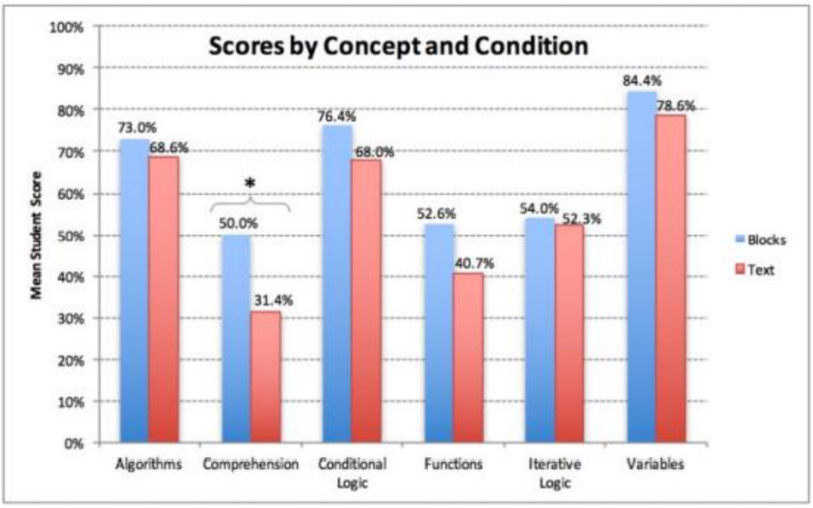
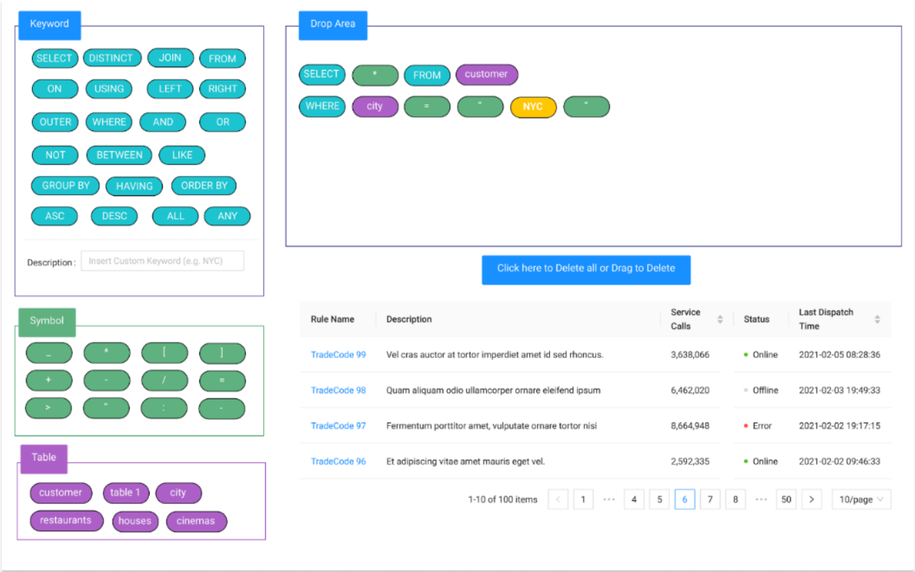


Figure 2. 2 Average Score Comparison of Block and Text by (Weintrop & Wilensky, 2017)

In addition to that, the average of result in pretest and post-test comparison for the groups which use blocks and text while answering the problem are illustrated as above. From that, we can safely conclude from this research that the group which used block outperforms the blocks condition in every category provided. To summarize, the usage of drag and drop in programming environment gives students ease of use while using the application as well as having a higher score.

SQL code reconstruction has been implemented by (Gorskis, 2018) in his journal entitled “SQL query construction from database concepts”. It is explained that there are 3 databases concepts, namely 'class type' – which is more based on tables and views –, 'Object Property' – which focus on the relationship among tables –, and ‘data property’ – a concept that is more focused on attributes that exist on each table –. The effective implementation of database concepts in SQL code reconstruction highly depends on the experts that understand the existing dataset structure. Therefore, from a learning perspective, the teacher and lecturer on this subject are expected to have in-depth knowledge of database concepts that is suitable for each case study so that the students can make valid SQL queries and finish the task more effectively.

In another research (Phewkum et al., 2019) entitled “Scramble SQL: A Novel, Drag-and- drop SQL Learning Tool” indicates that drag-and-drop practices in learning databases – especially SQL – has given new experience for students to solve case studies. By involving 30 participants that have done database subjects, it is concluded that they tend to have good knowledge of computer commands and enjoy the learning process with an average score of 3.90 by 5. By having an open-ended case in mind, research done by (Phewkum et al., 2019) gives options to the students to formulate the code reconstruction based solely on the students’ knowledge and creativity, which is prone to logical error even if the result is correct.



Rule Name	Description	Service Calls	Status	Last Dispatch Time
TradeCode 99	Vel cras auctor at tortor imperdiet amet id sed rhoncus.	3,638,066	● Online	2021-02-05 08:28:36
TradeCode 98	Quam aliquam odio ullamcorper ornare eleifend ipsum	6,462,020	● Offline	2021-02-03 19:49:33
TradeCode 97	Fermentum porttitor amet, vulputate ornare tortor nisi	8,664,948	● Error	2021-02-02 19:17:15
TradeCode 96	Et adipiscing vitae amet mauris eget vel.	2,592,335	● Online	2021-02-02 09:46:33

Figure 2. 3 Drag-and-Drop Model by Phewkum C.

(Reference: (Phewkum et al., 2019))

As illustrated in mockup above, the research done by (Phewkum et al., 2019) uses open-ended approach since most of the SQL Query keywords, symbols, and available tables are provided directly to the students to answer each question.

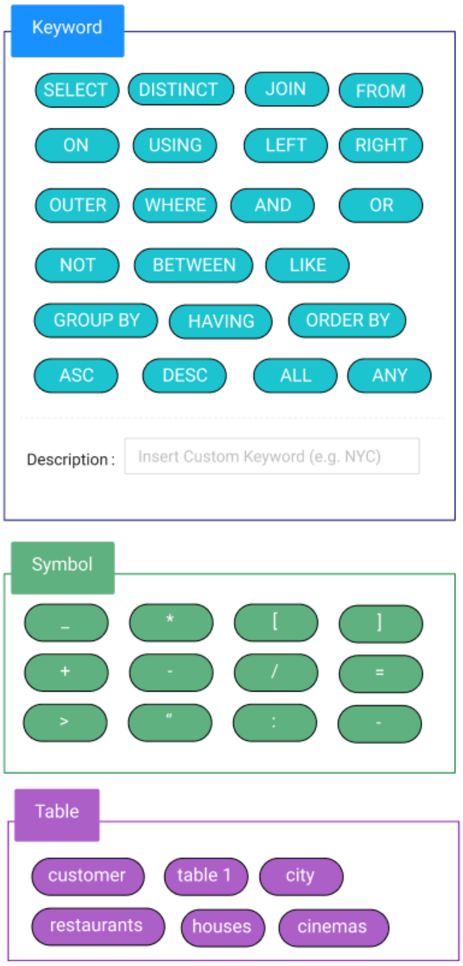
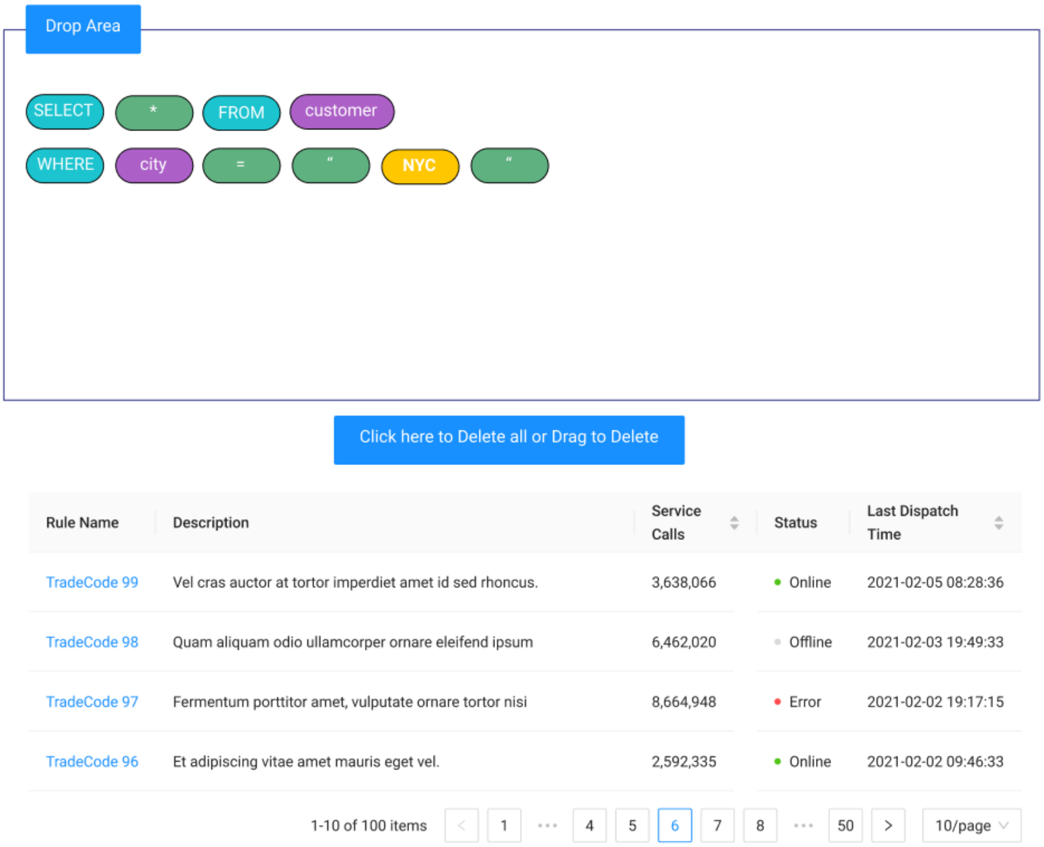


Figure 2. 4 Drag and Drop components

While doing the practices as well as the assignments from the lecturer in each question, students are required to solve the problem by dragging and dropping the SQL components that are categorized into 3 parts. Which are keywords, symbols, and tables available for its respective question. This approach uses open-ended since the SQL parts are given as the MySQL commands are available to be used. The parts that were close-ended in research done by (Phewkum et al., 2019) were the table options due to its matching of available database in respective case study.



Rule Name	Description	Service Calls	Status	Last Dispatch Time
TradeCode 99	Vel cras auctor at tortor imperdiet amet id sed rhoncus.	3,638,066	● Online	2021-02-05 08:28:36
TradeCode 98	Quam aliquam odio ullamcorper ornare eleifend ipsum	6,462,020	● Offline	2021-02-03 19:49:33
TradeCode 97	Fermentum porttitor amet, vulputate ornare tortor nisi	8,664,948	● Error	2021-02-02 19:17:15
TradeCode 96	Et adipiscing vitae amet mauris eget vel.	2,592,335	● Online	2021-02-02 09:46:33

Figure 2. 5 Mockup of Drop Area box and Table preview

The Illustration above shown us the drop box as the target of Dragged SQL parts on the left. SQL parts constructed in the drop box will be validated with the valid answer of its respective question. Just below that, there is a button provided to reset the drop area to empty state. The table below the button is used to preview the table being used in order to answer the question.

Whereas the ease-of-use aspect while answering the practices get feedbacks from the user that the system is easy to use and scored 3.56 out of 5. As stated from the journal by (Phewkum et al., 2019) that this may be due to participants’ who could touch type quickly, did not prefer a drag-and-drop settings.

The research entitled “The Different Role of Working Memory in Open-Ended Versus Closed-Ended Creative Problem Solving: A Dual-Process Theory Account” by (Lin & Lien, 2013) has concluded that when solving a scientific task, the working memory of a person is dominated by the thought of ‘how to solve the problem’ in technical approach and highly depends on the complexity of the problem itself. 40 participants in that research stated that they stuck on creating correct hypotheses and needed to revise their idea to solve the problem. In other words, if there is any case that requires some predefined syntax or formulation or rules, the thinking ability and memory capacity of the person will be fully concerned with finishing the task while considering the rules at the same time. By having a close-ended approach, the complexity level can be reduced and since the rules are predefined, participants are more resistant to syntax errors.

Previous research on literature review has indicated that a combination of drag-and-drop and learning SQL queries gives a new experience to students in database courses. There are some parts of practical implementation of past research that have the potential to be improved. Such as observing the understanding difference of database concepts based on specified testing parameters as well as the implementation of the close-ended approach in learning database courses.

2.2. Basic Theory

2.2.1. Database

A database is a collection of structured and organized data that are stored in a computer system. Database configuration can be done by using Database Management System (Surbakti, 2018). Databases play an important role in information systems, such as storing and processing data (Luthfi & Ayu, 2019). In its development, the database has 2 categories, namely relational database and non-relational database. The consideration of choosing one of these is the data type, data size, system capability, and maintainer skills for managing the existing database.

2.2.2. SQL Query

SQL Query is the most common operation to manipulate databases. This is because SQL Query is useful for analysing, storing, and retrieving data that has been processed from a database (Borthick et al., 2001). A description of the process of retrieving or manipulating the database by the user can be illustrated as follows:

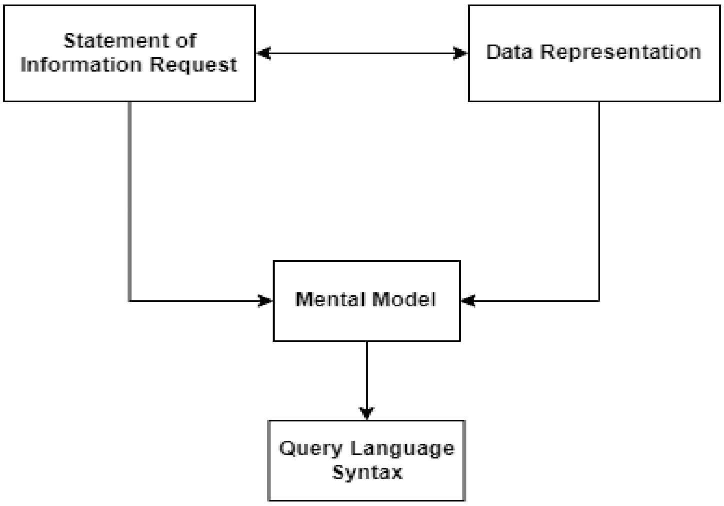


Figure 2. 6 Model of end users' query formulation processes

(Reference: (Borthick et al., 2001))

As the complexity of the query grows, a higher cognitive level of database administrators is required to make a query formula that has good performance. A commonly used approach is to use a subquery. By the top-down method, a complex problem can be analyzed and divided into some simpler problems to solve. (Hoque et al., 2014)

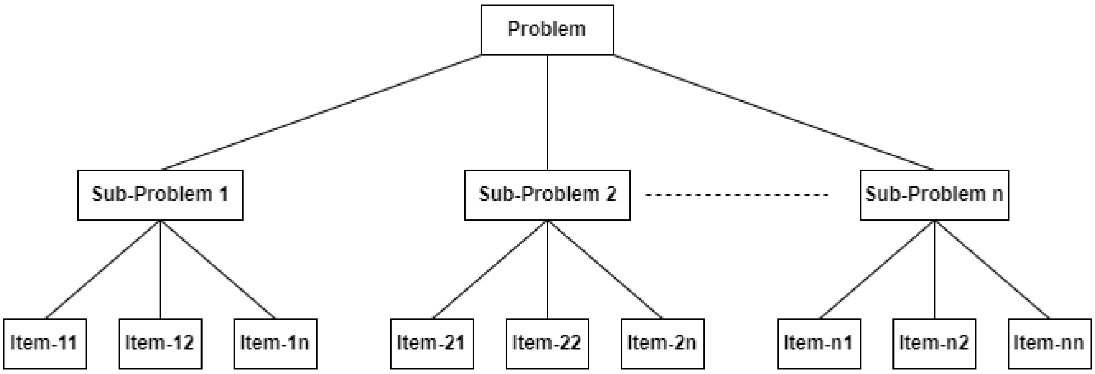


Figure 2. 2 Top-down Analysis of PBL Problem (Reference: (Hoque et al., 2014))

2.2.3. Drag-and-Drop

The *drag-and-drop* operation consists of object selection, object transforming, and placing it in another place. A drag-and-drop implementation may improve user experience on certain programming activity, and a faster problem-solving process since it can reduce typing error and gives flexibility for the users to adjust code in order (Phewkum et al., 2019). Moreover, the implementation of *drag-and-drop* showed greater learning gains and higher level of interest in solving problems (Weintrop & Wilensky, 2017).

2.2.4. Close-Ended Approach

A close-ended approach is a way for respondents or participants to choose one from multiple predefined answers based on the context of an existing problem. That way, the data retrieval process can be done quickly and easily (Hyman, 2016). When respondents are faced with a problem using a close-ended approach, their thinking process steps start from interpreting the problem, understanding the context of the measured score, and determining the correct answer based on their respective perceptions and understandings (Baburajan et al., 2021).

2.2.5. SQLearn

SQLearn is a learning SQL platform for students to understand database concepts, specializing in query SQL. To use the application, the registered lecturer of the database subject should create questions for a class. For each question, the lecturer provides the SQL queries part required to solve the quiz or case study. These SQL parts components need to be constructed by students by dragging and dropping them to a specified place. By close-ended approach, while it is possible to make the wrong answer, the SQL parts given by the lecturer should be a huge lead for students to answer it correctly.

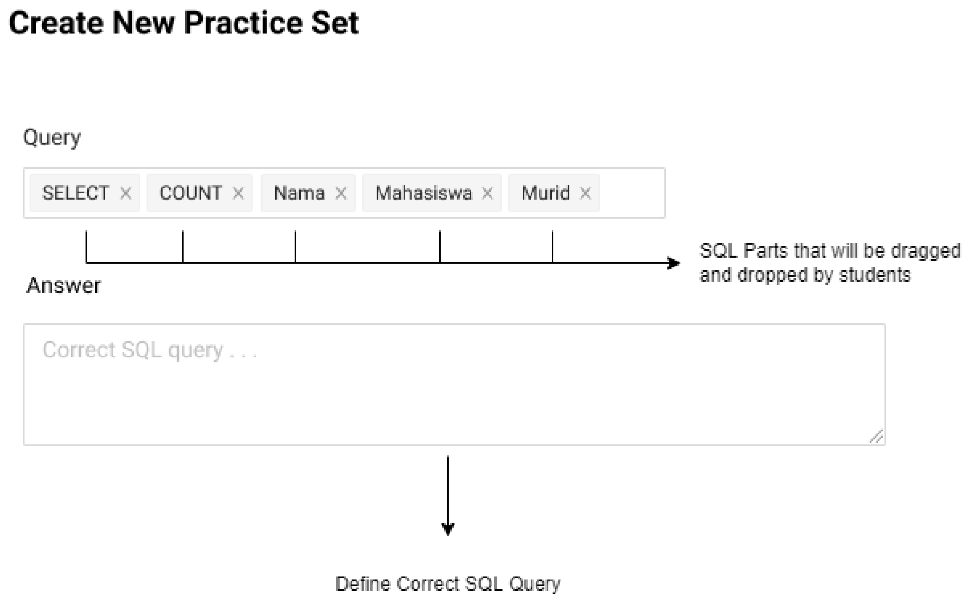


Figure 2. 7 Lecturer - Application Concept

From the students' perspective, firstly the students need to choose the class and quiz created by their lecturer, Then, they are required to construct a valid SQL query to answer the question based on options available by dragging and dropping it. In its process, students are allowed to check and validate their constructed SQL queries by clicking 'Check Query'. If they are confident enough, they can just click 'Submit' directly. Based on the previous explanation, the application concept of SQLearn is illustrated as follows:

Write an SQL query to fetch “FIRST_NAME” from Worker table in upper case.

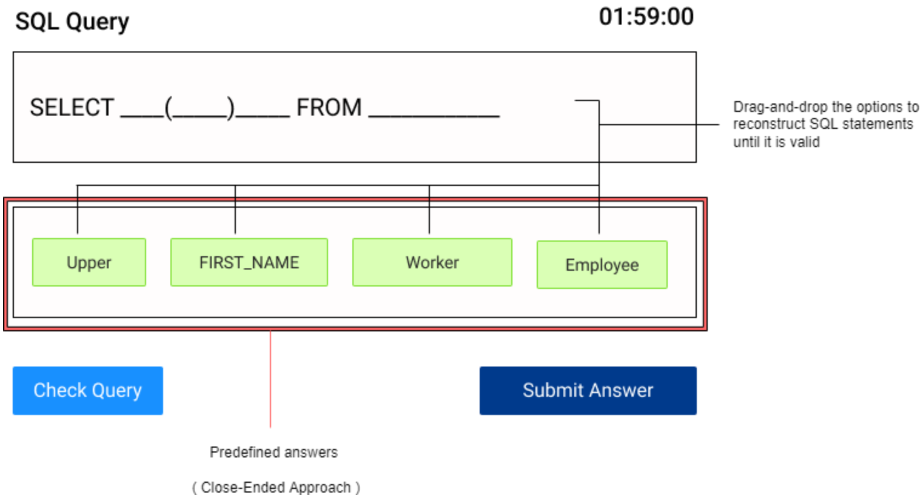


Figure 2. 8 Students - Application Concept

This application complements the idea of past researches regarding the SQL Learning Platform that uses conventional approach, not overrides nor replacing it. This application usage can become an alternative for the lecturers to give variation on form of question which will be given to the students for practices. While for the student, they can get new experience while answering a specific problem with predefined answers (close-ended approach).

As for the case study in Politeknik Negeri Malang, the database learning process is done on 2nd semester of Informatics Technology Major. Therefore, the main role of the application will be the lecturer of Database course and early-level students.

CHAPTER III. RESEARCH METHODOLOGY

3.1. Time and Place of Research

The research will be conducted in State Polytechnic of Malang and will be held for 5 months from January 2022 until May 2022.

3.2. Data Collection

Data that will be used in this research is students' pre-test and post-test score results. Moreover, while the students are dragging and dropping, the system will collect their activity log. This activity log includes:

- Time spent on solving a case study
- Number of attempts while checking query
- Constructed SQL Record with timestamps

But for this research purpose, the activity log is collected and will not proceed to processing the activity log any further.

3.2.1. Field Research

Interview with one of the lecturers that teaches database subjects in Information Technology Major State Polytechnic of Malang to get accurate information. The author will provide some questions regarding the database teaching style for the SQL Query chapter, especially in the SELECT statements. The result of the interview will be analyzed and used to formulate a better close-ended approach and suit the learning flow of database subjects with the student's capability.

3.2.2. Library Research

The author has performed literature reviews from journals and articles regarding the key problem that will be discussed to get a theoretical basis and gather some insights on approaches being used in previous research.

3.3. Data Processing

In this research, data will be processed using Non-Parametric Wilcoxon Signed Ranks Test or Paired T-Test to determine whether there is any difference on pre-test average score and post-test average score. Wilcoxon Signed Ranks Test will be proceeded if both pre-test and post-test is not normally distributed. Whereas the Paired T-Test, will be used if pre-test and post-test data is normally distributed. To identify the such condition, firstly, collected data will need to be processed using Kolmogorov-Smirnov Test.

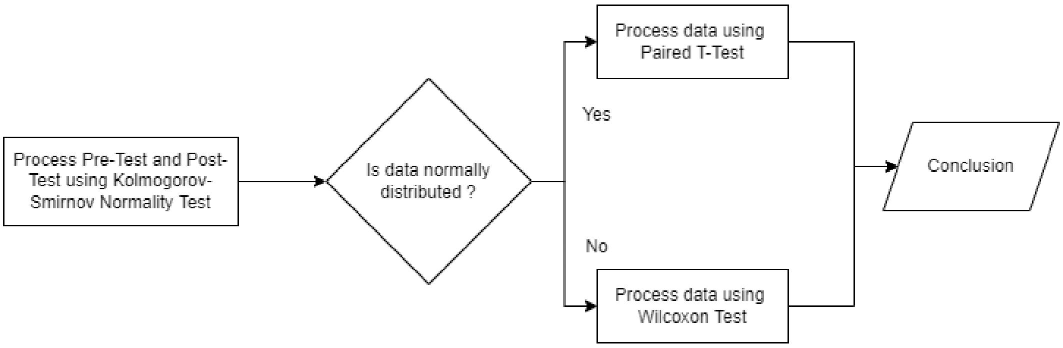

Process Pre-Test and Post-Test using Kolmogorov-Smirnov Normality Test
Is data normally distributed ?
Yes
No
Process data using Paired T-Test
Process data using Wilcoxon Test
Conclusion

Figure 3. 1 Data Processing Flowchart

This research uses *SPSS* (Statistical Package for the Social Sciences) software to ease researcher process data collected from students that have participated. Below are the walkthrough of data processing using *SPSS* software starting from data preparation, Kolmogorov-Smirnov Normality Test, Wilcoxon Signed Ranks Test, and the result.

3.3.1. Kolmogorov-Smirnov Test

Kolmogorov-Smirnov test is a method to determine the distribution normality of paired data. In this research, this method gives insight to process data whether using Paired T-Test or Wilcoxon Signed Rank Test. The decision-making basis of normality are listed as follows:

a. If the value of *Asymp. Sig. (2-tailed)* > 0,05. Data is normally distributed.
b. If the value of *Asymp. Sig. (2-tailed)* < 0,05. Data is not normally distributed.

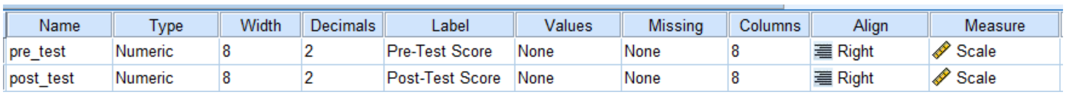
After taking a note on how the distribution normality is set, the steps to do Kolmogorov-Smirnov test in *SPSS* are as follows:

a. Prepare pre-test and post-test sample that will be used in *Kolmogorov-Smirnov* test.

Pre-Test	Post-Test
50	60
30	50
60	60
70	70
60	70
70	80
60	60
60	80
60	60
30	60

Table 3. 1 Sample Data *Kolmogorov-Smirnov* Test

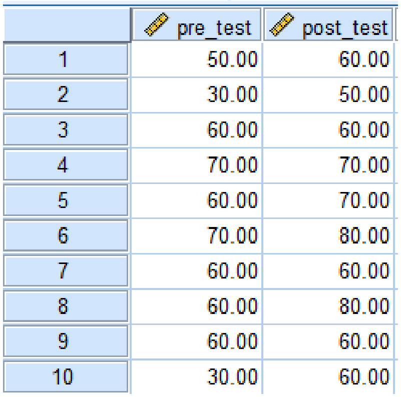
b. Create new project in SPSS. In "Variable View" fill the variable properties as follows



Name	Type	Width	Decimals	Label	Values	Missing	Columns	Align	Measure
pre_test	Numeric	8	2	Pre-Test Score	None	None	8	Right	Scale
post_test	Numeric	8	2	Post-Test Score	None	None	8	Right	Scale

Figure 3. 2 Variable View *Kolmogorov-Smirnov* Test

c. In "Data View" fill the data according to the sample data that is shown in Table 3.1



	pre_test	post_test
1	50.00	60.00
2	30.00	50.00
3	60.00	60.00
4	70.00	70.00
5	60.00	70.00
6	70.00	80.00
7	60.00	60.00
8	60.00	80.00
9	60.00	60.00
10	30.00	60.00

Figure 3. 3 Data View Kolmogorov-Smirnov Test

d. In menu, click “*Analyze* “> “*Nonparametric Tests*” > “*Legacy Dialogs*” > “*1-Sample K-S*”.

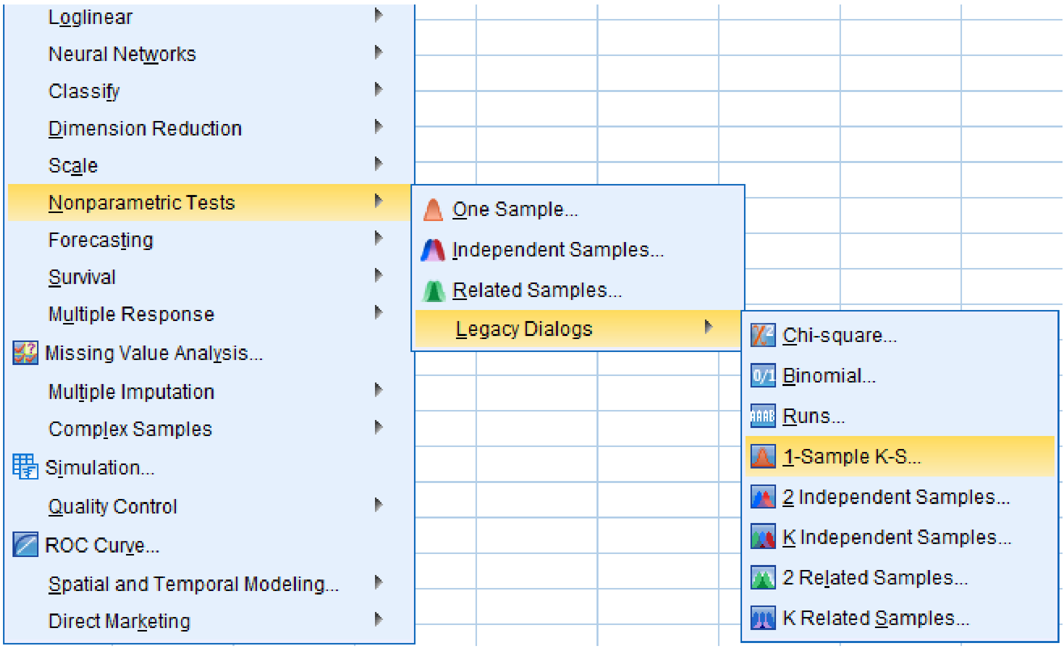
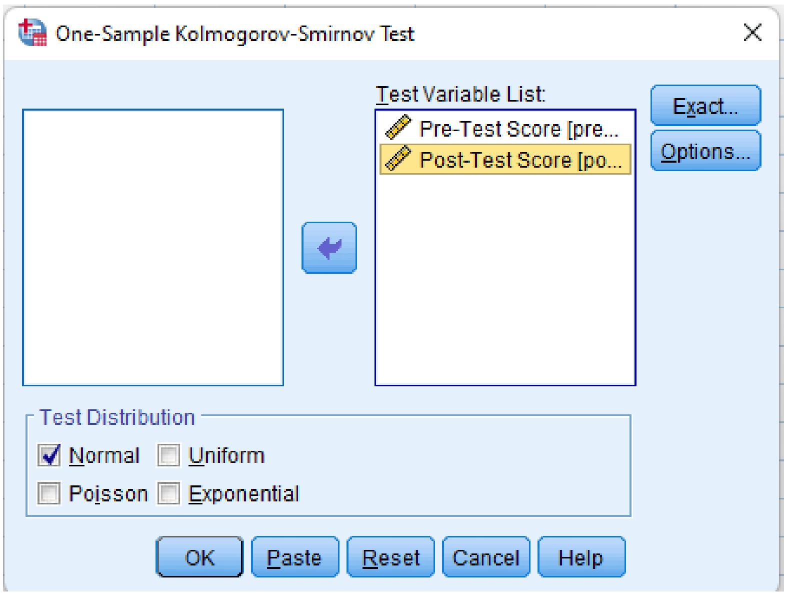


Figure 3. 4 Normality Test Menu *1-Sample K-S* in *SPSS*

e. In dialog *One-Sample Kolmogorov-Smirnov* Test, input pre_test and post_test variable as *Test Variable List*. Then, check the box in *Normal*.



Test Variable List:
Pre-Test Score [pre...
Post-Test Score [po...

Test Distribution
☑ Normal ☐ Uniform
☐ Poisson ☐ Exponential

Figure 3. 5 Dialog Kolmogorov-Smirnov Test

f. Then, SPSS will display the result of *One-Sample Kolmogorov Smirnov Test.* Below are the result with sample data listed

One-Sample Kolmogorov-Smirnov Test			
		Pre-Test	Post-Test
N		10	10
Normal Parameters	Mean	55.000	65.000
	Std. Deviation	14.337	9.718
Most Extreme Differences	Absolute	0.336	0.297
	Positive	0.164	0.297
	Negative	-0.336	-0.203
Test Statistic		0.336	0.297
Asymp. Sig. (2-tailed)		0.002	0.013

Table 3. 2 Normality Test Result with Sample Data in SPSS

g. Interpretation of *Kolmogorov-Smirnov* Test Result

Based on *Kolmogorov-Smirnov* Test Result in Figure 3.6, it is known that the *Asymp. Sig. (2-tailed)* of Pre-Test score is 0.002 (< 0.05) and Post-Test score is 0.013 (< 0.05). According to decision-making basis, it can be concluded that pre-test and post-test score from sample data is not normally distributed.

3.3.2. Wilcoxon Signed Rank Test

Wilcoxon test is used for determining whether there is any average difference between two paired samples that are not normally distributed. The decision-making basis are as follows:

a. If the value of *Asymp. Sig. (2-tailed)* < 0,05. Null hypotheses (H0) are rejected and H1 is accepted. Within context of this research, it means there is average difference between pre-test score and post-test score.

b. If the value of *Asymp. Sig. (2-tailed)* > 0,05. Null hypotheses (H0) are accepted and H1 is accepted. Within context of this research, it means there is no average difference between pre-test score and post-test score.

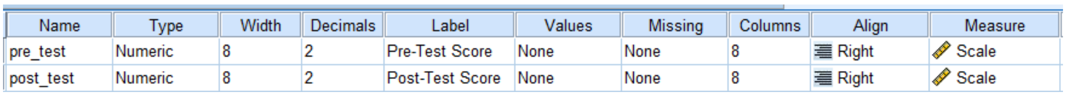
After taking note on how the decision-making and the define null hypotheses. Steps to do Wilcoxon Signed Rank test in *SPSS* are as follows:

a. Prepare sample data pre-test and post-test scores for Wilcoxon Signed Rank Test

Pre-Test	Post-Test
50	60
30	50
60	60
70	70
60	70
70	80
60	60
60	80
60	60
30	60

Table 3. 3 Sample Data for Wilcoxon Test

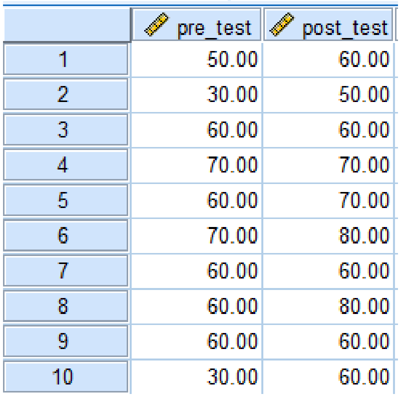
b. Create new page in SPSS. In “Variable View” fill the variable properties as follows



Name	Type	Width	Decimals	Label	Values	Missing	Columns	Align	Measure
pre_test	Numeric	8	2	Pre-Test Score	None	None	8	Right	Scale
post_test	Numeric	8	2	Post-Test Score	None	None	8	Right	Scale

Figure 3. 6 Variable View *Wilcoxon Signed Rank* Test

c. In “Data View” fill the data according to the sample data that is shown in Table 3.2



	pre_test	post_test
1	50.00	60.00
2	30.00	50.00
3	60.00	60.00
4	70.00	70.00
5	60.00	70.00
6	70.00	80.00
7	60.00	60.00
8	60.00	80.00
9	60.00	60.00
10	30.00	60.00

Figure 3. 7 Data View *Wilcoxon Signed Rank* Test

d. In menu, click “*Analyze*” > “*Nonparametric Tests*” > “*Legacy Dialogs*” > “*2 Related Samples*”.

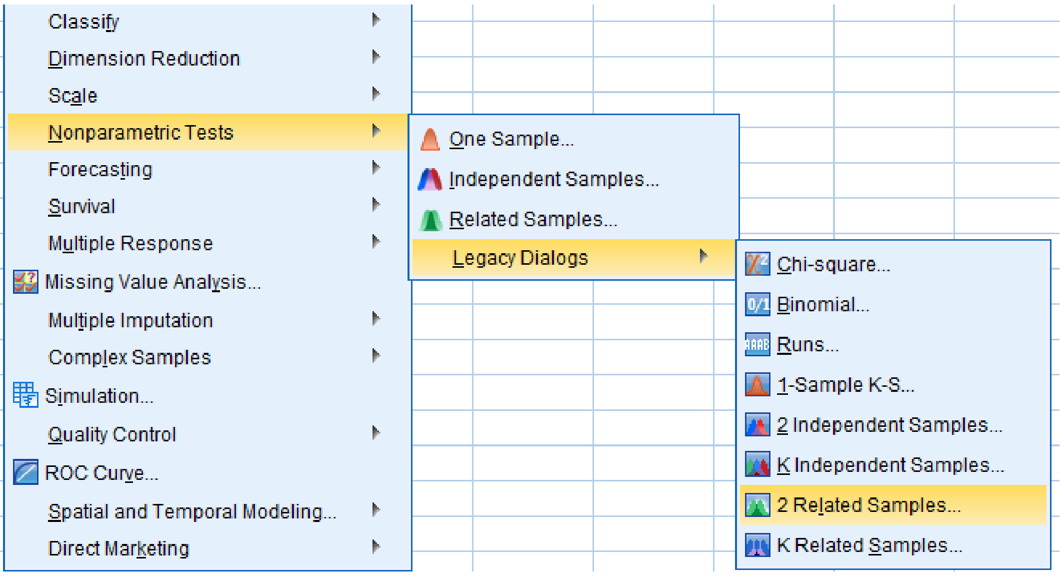
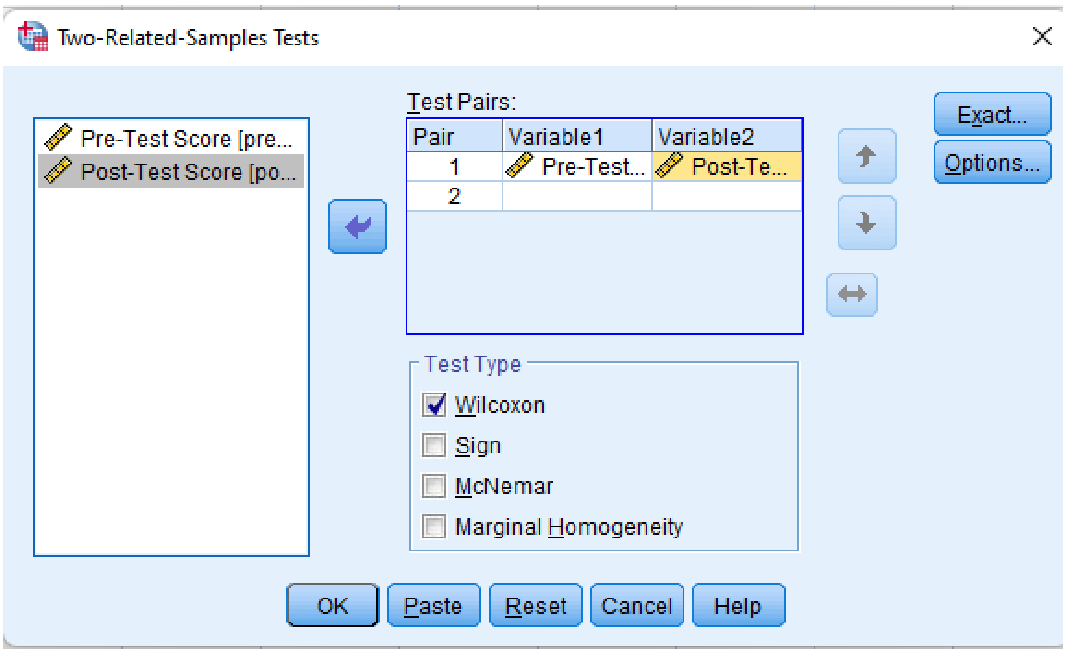


Figure 3. 8 Menu *Wilcoxon Signed Rank* Test

e. In dialog “*Two Related Samples Tests*”. Input pre_test and post_test variable as *Test Variable List*. Then, check the box *Wilcoxon*.



Test Pairs:

Pair	Variable1	Variable2
1	Pre-Test...	Post-Te...
2		

Test Type

☑ Wilcoxon
□ Sign
□ McNemar
□ Marginal Homogeneity

Figure 3. 9 Dialog Wilcoxon Test

f. Wilcoxon Test Result

Ranks				
		N	Mean Rank	Sum of Ranks
Pre-Test & Post-Test	Negative Ranks	0	0.00	0.00
	Positive Ranks	6	3.50	21.00
	Ties	4		
	Total	10		

Table 3. 4 Wilcoxon Ranks Test Result

Test Statistics	
	PostTest - PreTest
Z	-2.232
Asymp. Sig. (2-tailed)	0.026

Table 3. 5 Wilcoxon Test Statistics Result

g. Interpretation of *Wilcoxon Signed Ranks* Test Result

Based on Table 3.4, it is specified that *Negative Ranks* has 0 value, and sum of negative rank value is 0 as well. This means that there is no decreasing value from pre-test and post-test sample data. For *positive ranks,* there is 6 data that has increase of value from pre-test to post-test sample data. The average rank is 3.50 and the sum of ranks is 21.00. And lastly, there are 4 sample data that has same pre-test and post-test value.

Based on Table 3.5, it is determined that *Asymp. Sig. (2-tailed)* is 0,026 (< 0,05). Therefore, H0 (null hypotheses) is rejected and H1 is accepted. In conclusion, from sample data, there is significant average difference between pre-test and post-test score.

CHAPTER IV. SYSTEM DESIGN AND ANALYSIS

4.1.Application Overview

SQLearn is a web-based application to help students practicing their understanding of MySQL syntaxes, especially in SELECT statements. Students will do the practice scheduled by the lecturer in Database course. The idea of SQLearn is to give student new method to do the practices with *drag-and-drop* concept and close-ended approach. To respond to that, lecturers need to provide SQL parts alongside with SQL hints to finish questions. SQLearn also has its automated assessment to handle the grading by considering the SQL constructed parts made by the students. In addition to that, when students finished their session, the system records their logs movement and timer as well.

4.2. User Analysis

This application has 3 different roles, such as lecturers, students, and admins. Lecturers are the one who manage case studies (databases that will be the reference to run query in grading process), classes, schedules, questions, and see the students' score, as well as registering students. Whereas for student role, they are capable of doing the practices only after the lecturer create a schedule for them. Admins are allowed to manage lecturers, adjusting grading rules, and setting threshold.

4.3. Functional Requirement Analysis

Functional Requirement is the features that are the application is capable of. Such processes are as follows:

- Application can manage the case studies, classes, questions, question sets, schedules, scores, as well the user its respective roles.
- Application can receive user input in various type. Such as logs when *drag-and-drop* process and filling forms,
- Application can do automated assessment for students' practices
- Application can fetch data correctly for SQL Parts, SQL Hints and others and shown in its respective feature.

- Application can store students log in database and display it for lecturer role

4.4. Non-Functional Requirement Analysis

There are some requirement analyses for non-functional parts, such as its tools in order to run this application properly which can be categorized into software and hardware specifications. The details are as follows:

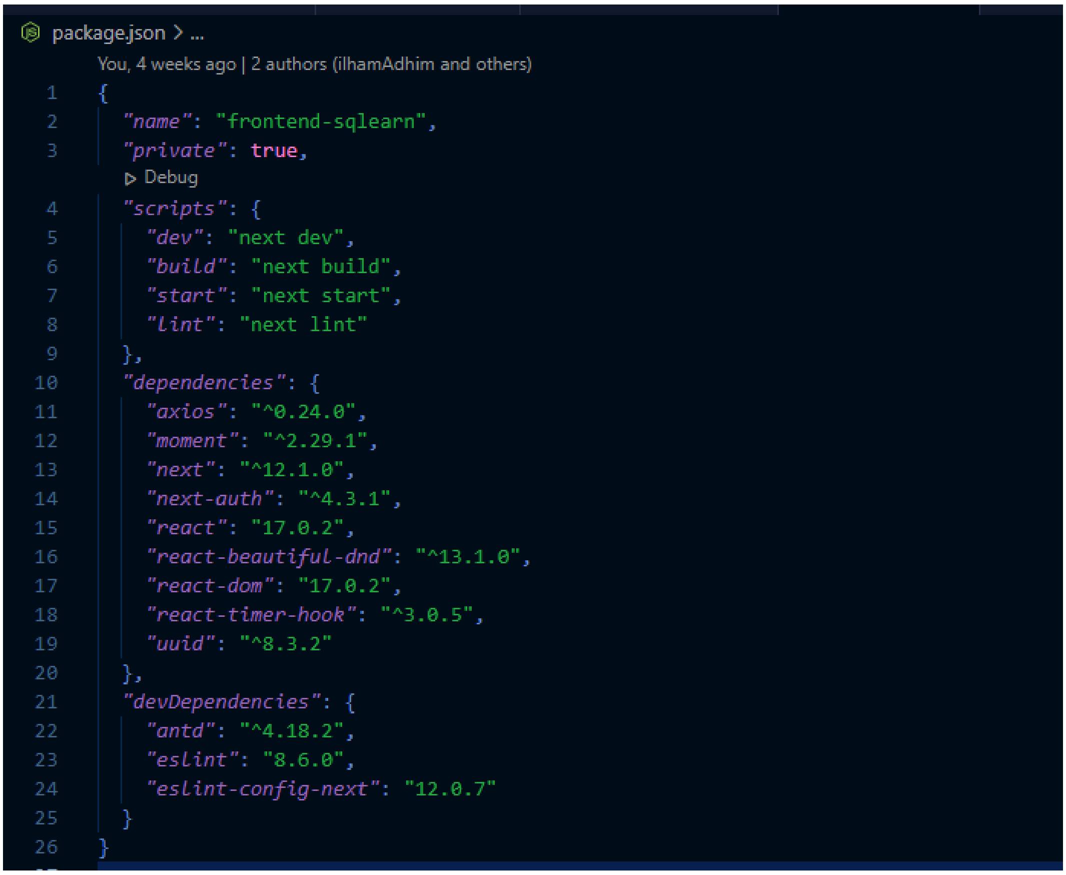
4.4.1. Software

These are general software specification for building the application and displayed as below:

No	Software
1	Operating System Windows 7/8/10/11
2	VS Code Editor
3	Node JS v.16.13.1
4	MySQL
5	Microsoft Edge (Chromium Based), Google Chrome, Mozilla Firefox

Table 4. 1 Software Specification for SQLearn

In addition to that, these are the dependencies used while building the application:



```json
{
  "name": "frontend-sqlearn",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "axios": "^0.24.0",
    "moment": "^2.29.1",
    "next": "^12.1.0",
    "next-auth": "^4.3.1",
    "react": "17.0.2",
    "react-beautiful-dnd": "^13.1.0",
    "react-dom": "17.0.2",
    "react-timer-hook": "^3.0.5",
    "uuid": "^8.3.2"
  },
  "devDependencies": {
    "antd": "^4.18.2",
    "eslint": "8.6.0",
    "eslint-config-next": "12.0.7"
  }
}
```

4.4.2. Hardware

These are general software specification for building the application and displayed as below:

No	Hardware
1	Processor Intel Core i3 1115
2	RAM 8GB
3	SSD 512 GB

Table 4. 2 Hardware Specification for SQLearn

4.5. System Design

4.5.1. Use Case Diagram

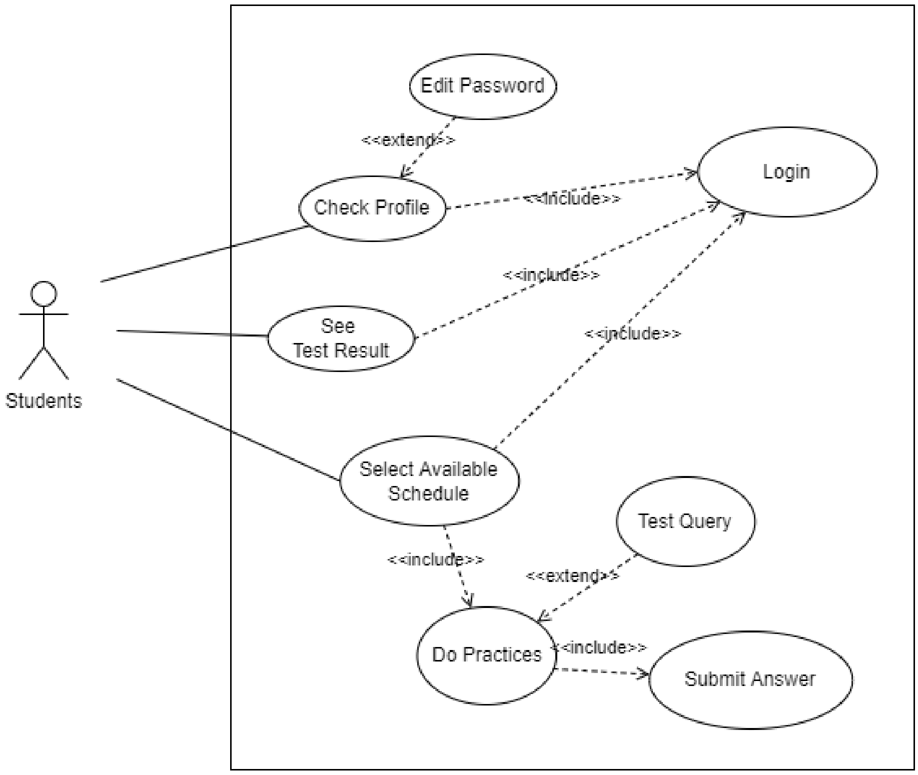


Figure 4. 1 Use Case Diagram - Student

As illustrated in the above diagram, students are able to check their profile, see how's their performance in each practice by accessing test result page, and select available schedule to do the practices. As for the practice itself, students have option to check their query, this action will be recorded and being used in assessment by counting how many attempts they checked their query. After done, they need to submit the answer.

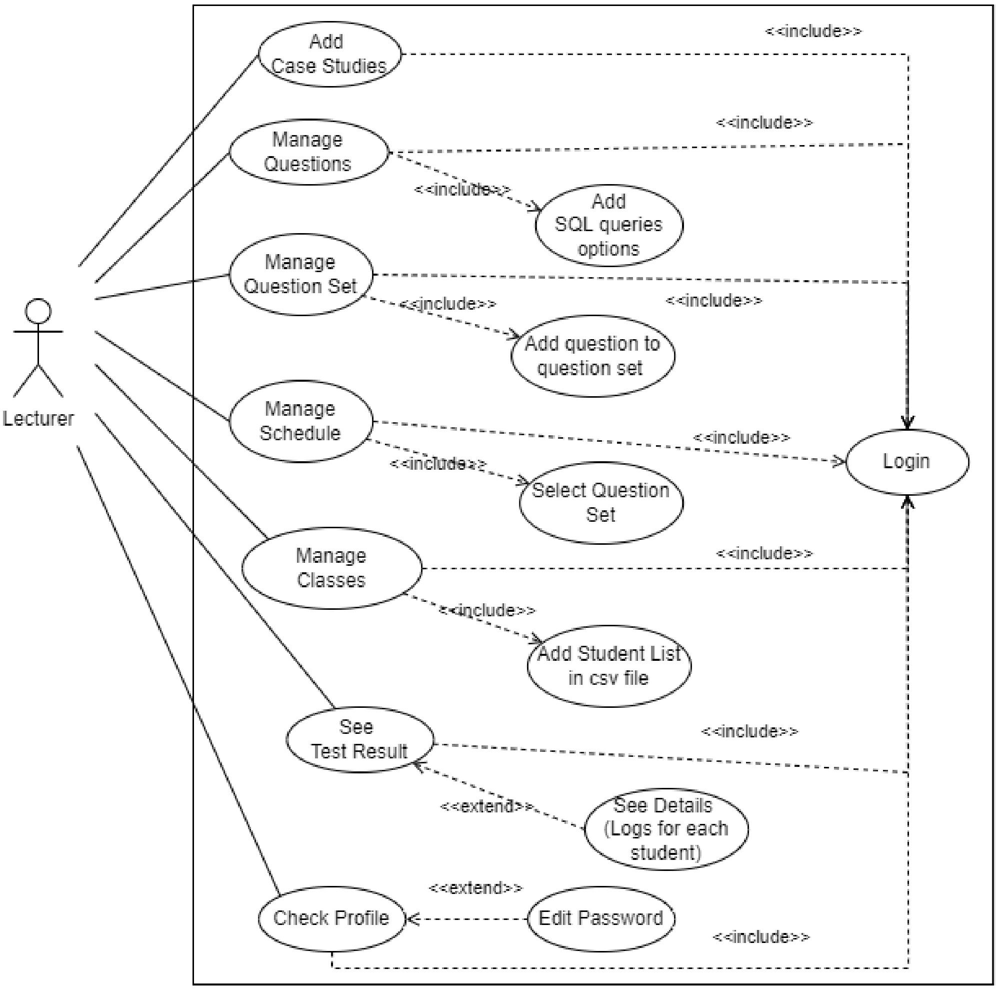


Figure 4. 2 Use Case Diagram - Lecturer

As illustrated in the above diagram, the lecturers are able to do things that are related to creating questions and more. After logged in to the system, lecturer can create a case study, manage questions, manage question set, manage schedule, manage classes, see test result, and check profile. Each has its respective action. In the process of creating a question, lecturer need to add SQL Queries options to define the category to close-ended.

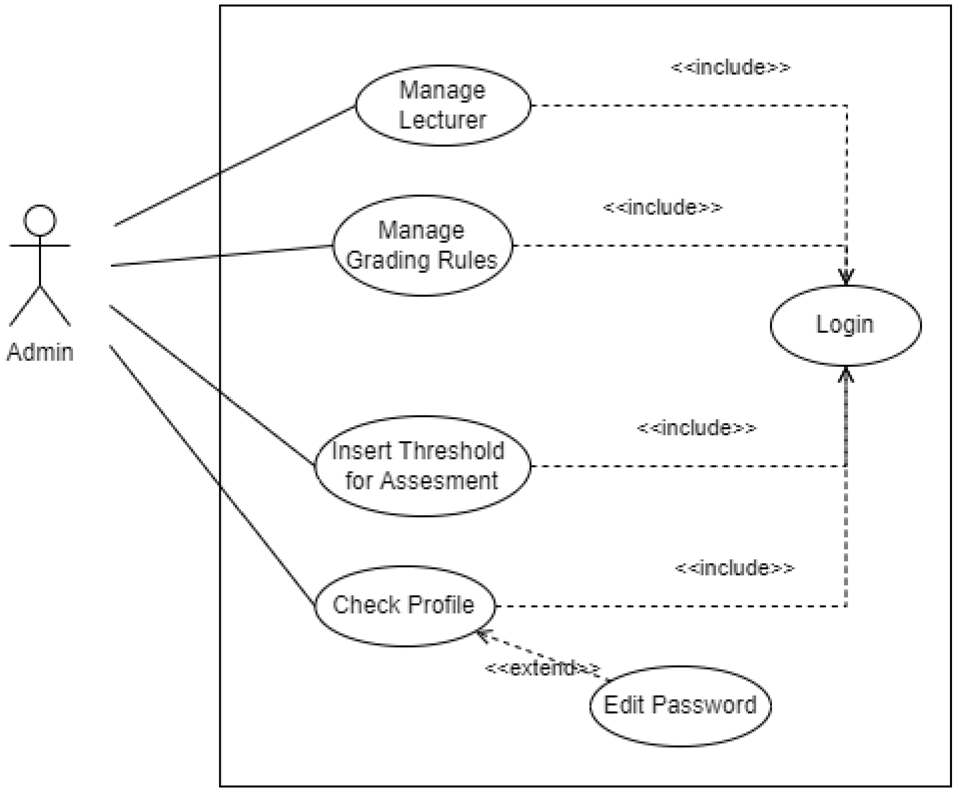


Figure 4. 3 Use Case Diagram - Admin

As illustrated in the above diagram, admins are responsible of managing the lecturer, the grading rules, threshold assessment, and check their own profile. The grading rules are used for validating how many attempts student has performed while checking their query. Whereas the threshold is used for automation assessment process.

4.5.2. System Architecture

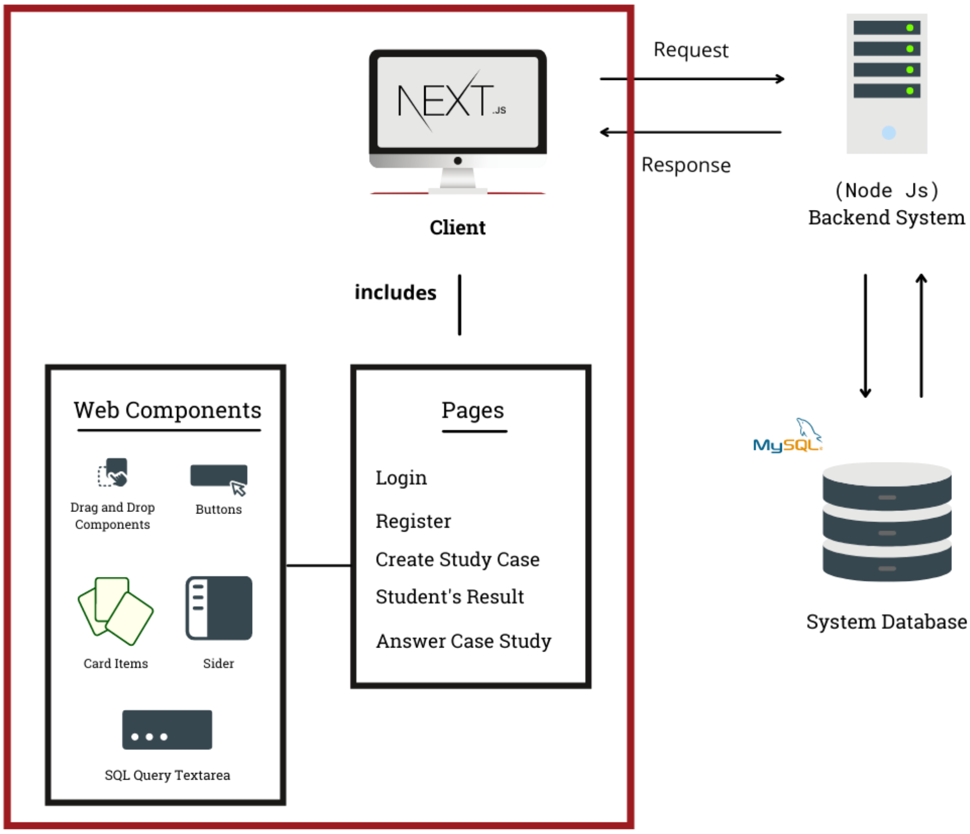


Figure 4. 4 System Architecture of SQLearn

As an effect of having multiple case studies in one system, the usage of multiple databases included in SQLearn is inevitable. Such databases are also divided into case-study databases and the system database. With Node JS serving the backend process, it is expected to have multiple ORMs as well. The backend system will handle authentication, SQL query validation, and processing other data.

As for the frontend, Next JS is an open-source framework that utilizes React JS which is a JavaScript library that allows the developer to create web components to build the user interface of the SQLearn web application. Each component has its roles and is independent to each other. With this component approach in building a website, it is also possible to have a component that can be dragged and dropped.

4.5.3. Activity Diagram

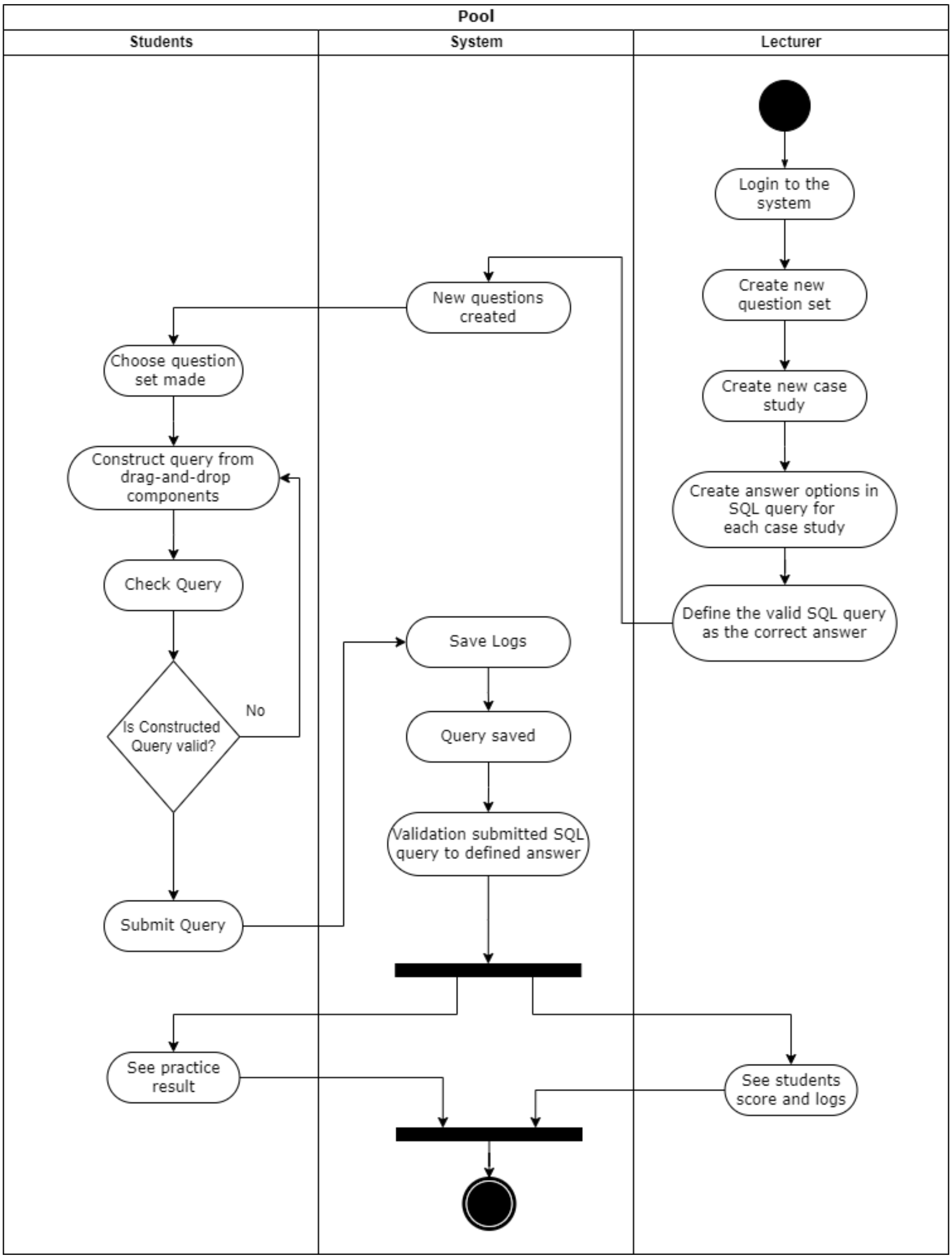

Pool
Students
System
Lecturer
Login to the system
Create new question set
Create new case study
Create answer options in SQL query for each case study
Define the valid SQL query as the correct answer
New questions created
Choose question set made
Construct query from drag-and-drop components
Check Query
Is Constructed Query valid?
No
Submit Query
Save Logs
Query saved
Validation submitted SQL query to defined answer
See practice result
See students score and logs

Figure 4. 5 Activity Diagram of SQLearn

To have access to SQLearn, both lecturers and students need to be registered and logged in with their credentials. As for this research, and to make the testing phase more seamless, it is recommended for lecturers to set their credentials with their NIDN. Whereas students are expected to set their credentials with NIM.

To make this application usable, lecturers should make the question sets first. Each question set has a one-to-many case study. For each case study, the lecturer should define the SQL query options and their valid answer. While the valid answer is the constructed SQL query, These SQL query options will then be represented as drag-and- drop components in the students' menu.

For students, SQLearn is the playground to practice their SQL query for SELECT statements. After their respective lecturer in the database subject has added question sets, students are allowed to do the practice. In their dashboard, they will have a menu that redirects them to the available question sets. To finish the practice, they should solve the case study by drag-and-drop from the provided SQL queries by their lecturer. In the process, they can either test their constructed query first to check its validity, or just directly submit the SQL query to the system and proceed to the next case study.

Once the question sets are all answered by the students, the system will check and validate submitted SQL queries to the defined answers in the lecturer's role. When it is done, both roles are expected to get the result in their dashboard. For the students, the result will be the final score. Whereas for the lecturers, the result is the final score of students that have finished a specific question set.

4.5.4. Application Features

a. Login

Students and lecturers can log in to the system. Each role has specified features. Lecturers' login with their NIDN as their username and password. For students, they login with their NIM.

b. Create Questions

This feature is limited to the lecturer's role only. In this feature, the lecturer will need to create some questions for students' practice. In addition, the lecturer needs to specify SQL Statements as the drag-and-drop component in students' roles.

c. Practice SQL Query

This feature is limited to the students' roles only. Students are allowed to have practice on questions provided by their respective lecturer in the database subject. To finish the practice, students will need to construct the query based on the SQL components given.

d. Practice Result

Students and their respective lecturers in database subjects will have access to view the score of the practice and how is the student's performance on constructing SQL queries.

4.5.5. Database

1. Database Relation Scheme

The relationship amongst the tables is defined by using database relation scheme. Such relation scheme that will builds the way data being connected behind the application is attached in Lampiran 1. Database Relation Scheme.

2. Database Tables Structure

The table structure is way of configuring database to store the data involved for each respective table that has its own data type. These data types make it more convenient to process the data. Below is database table structure of SQLearn:

Column	Data Type	Description
id	*INT(11)*	*PK, AI*
name	*VARCHAR(50)*	-
db_list_id	*INT(11)*	*FK, Ref. db_list (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 3 Table case_studies

Column	Data Type	Description
class_id	*INT(11)*	*FK, Ref. classes (id)*
schedule_id	*INT(11)*	*FK, Ref. schedules (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 4 Table class_schedules

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
name	*VARCHAR(20)*	*UNIQUE*
semester	*INT(11)*	-
user_id	*INT(11)*	*FK, Ref. users (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 5 Table classes

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
description	*TEXT*	-
user_id	*INT(11)*	*FK, Ref. users (id)*
label_id	*INT(11)*	*FK. Ref. questions_label (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 6 Table containers

Column	**Data Type**	**Description**
id	*INT(11)*	*PK, AUTO INCREMENT*
db_name	*VARCHAR(50)*	-
db_filename	*VARCHAR(100)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 7 Table db_list

Column	**Data Type**	**Description**
id	*INT(11)*	*PK, AUTO INCREMENT*
session_id	*INT(11)*	*FK, Ref. sessions (id)*
question_id	*INT(11)*	*FK, Ref. questions (id)*
answer	*Text*	-
answer_json	*Longtext*	-
type	*enum('start','test','submit','move','reset')*	-
similarity	*Decimal(10,2)*	-
is_equal	*Tinyint(1)*	-
timer	*Time*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 8 Table log_ session_student

Column	Data Type	Description
Question_id	INT(11)	FK, Ref. questions (id)
Container_id	INT(11)	FK, Ref. containers (id)
createdAt	DATETIME	-
updatedAt	DATETIME	-

Table 4. 9 Table question_containers

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
text	*TEXT*	-
sql_parts	*LONGTEXT*	-
sql_hints	*LONGTEXT*	-
answer	*LONGTEXT*	-
answer_pic	*TEXT*	-
tables	*TEXT*	-
case_study_id	*INT(11)*	*FK, Ref. case_studies (id)*
user_id	*INT(11)*	*FK, Ref. users (id)*
label_id	*INT(11)*	*FK, Ref. questions_label (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 10 Table questions

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
name	*VARCHAR(100)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 11 Table questions _label

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
start	*DATETIME*	-
finish	*DATETIME*	-
container_id	*INT(11)*	*FK, Ref. containers (id)*
description	*VARCHAR(360)*	-
type	*ENUM(‘latihan’,’ujian’)*	-
user_id	*INT(11)*	*FK, Ref. users (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 12 Table schedules

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
student_id	*INT(11)*	*FK, Ref. students (id)*
schedule_id	*INT(11)*	*FK, Ref. schedules (id)*
score	*INT(11)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 13 Table scores

Column	Data Type	Description
session_id	INT(11)	FK, Ref. sessions (id)
db_list_id	INT(11)	FK, Ref. db_list (id)
createdAt	DATETIME	-
updatedAt	DATETIME	-

Table 4. 14 Table session_db

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
student_id	*INT(11)*	*FK, Ref. students (id)*
schedule_id	*INT(11)*	*FK, Ref. schedules (id)*
session_started	*DATETIME*	*Current_timestamp()*
is_finished	*Tinyint(1)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 15 Table sessions

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
attemps	*TINYINT(4)*	-
value	*TEXT*	-
type	*ENUM('threshold','latihan','ujian')*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 16 Table settings

Column	Data Type	Description
student_id	*INT(11)*	*FK, Ref. students (id)*
class_id	*INT(11)*	*FK, Ref. classes (id)*
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 17 Table student_classes

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
username	*VARCHAR(50)*	-
password	*TEXT*	-
nim	*VARCHAR(50)*	*UNIQUE*
name	*VARCHAR(100)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 18 Table students

Column	Data Type	Description
id	*INT(11)*	*PK, AUTO INCREMENT*
username	*VARCHAR(50)*	-
password	*TEXT*	-
level	*ENUM(‘DOSEN’,’ADMIN’)*	-
no_induk	*VARCHAR(30)*	-
name	*VARCHAR(100)*	-
createdAt	*DATETIME*	-
updatedAt	*DATETIME*	-

Table 4. 19 Table users

4.5.6. Design Interface

Design Interface shown in here is the initial mockup that is prepared for the UI application to be built. Generally, design interface can be implemented in various viewports in multiple gadgets, such as design for website and mobile application. But for SQLearn, the design interface will only cover website as for its usage is focused in website for desktop browser viewport. The mockup design interfaces of SQLearn are shown as follows:

1. Available Schedule Page Design

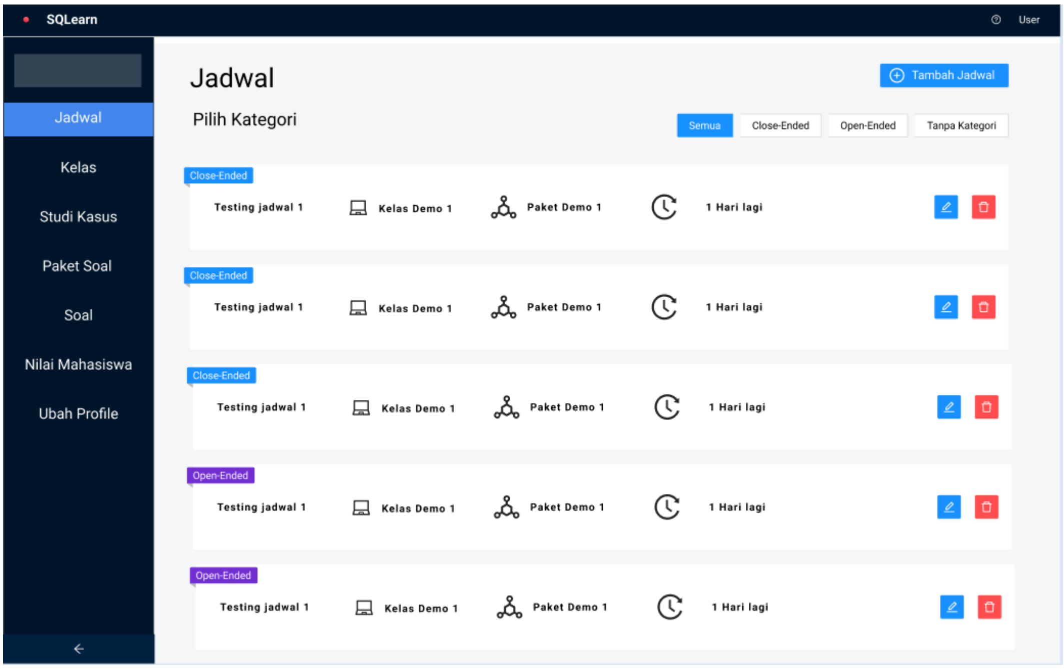


Figure 4. 6 Mockup Available Schedule page

After lecturer is logged in, they will be redirected to schedule page which displays all schedules that has been created in the system.

2. Add Schedule Page Design

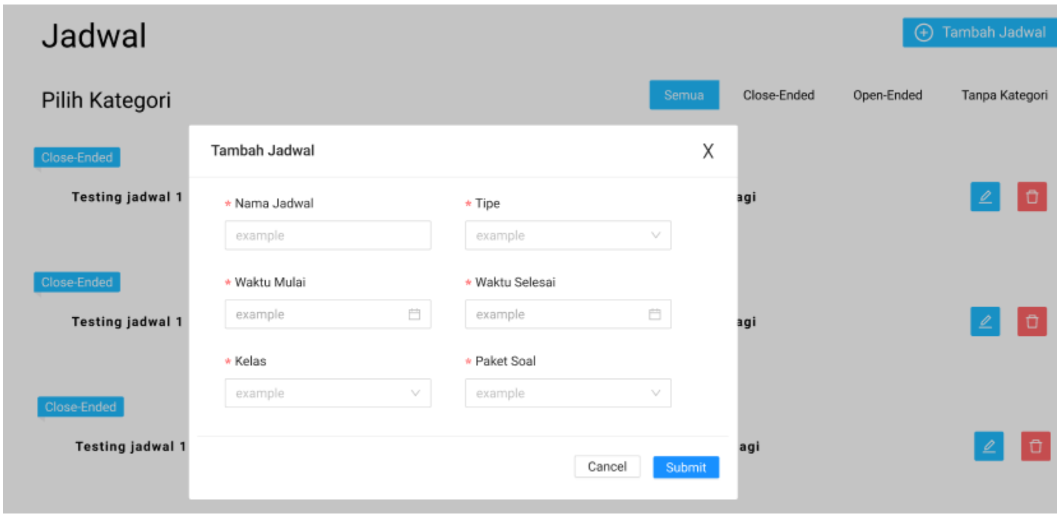


Figure 4. 7 Mockup Add Schedule page

To create a new schedule, lecturer is expected to fill schedule name, type, start-time, end-time, schedule for which class, and choose the question set.

3. Class List Page Design

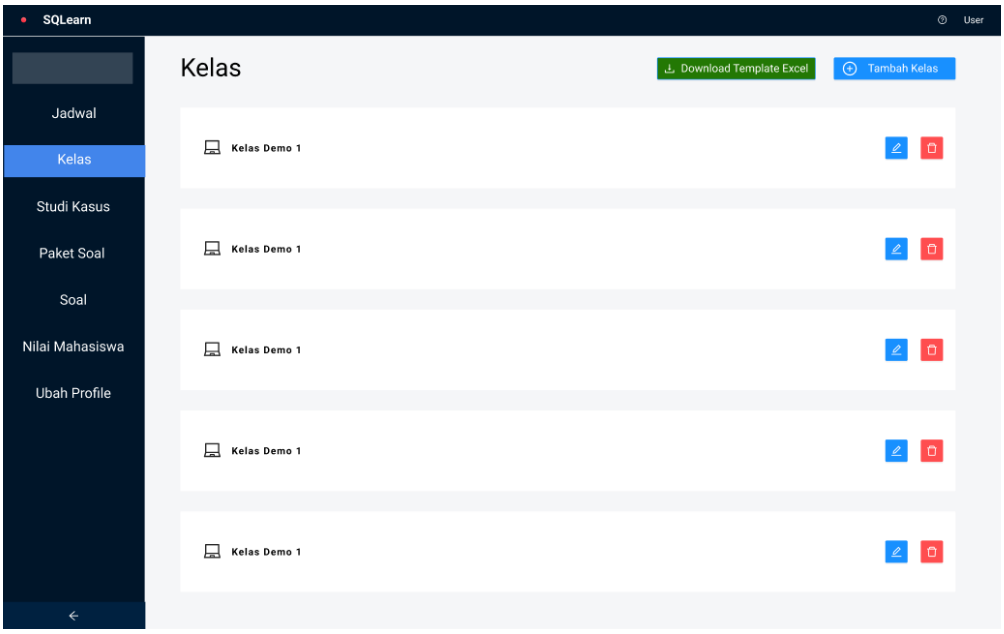


Figure 4. 8 Mockup Class List Page

Next feature available for lecturer is managing the classes. The page above displays all the classes that has been created by the lecturer. They can see the class member by clicking the detail button (blue).

4. Add Class Page Design

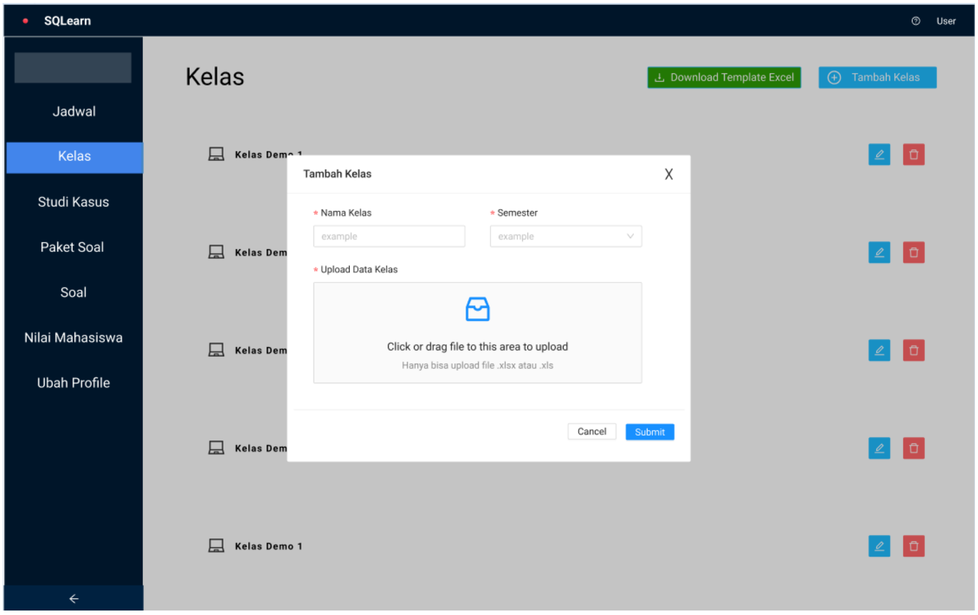


Figure 4. 9 Mockup Add Class Page

To add new class, lecturer will need to input class name, the semester, and excel file of students list of respective class.

5. Case Study List Page Design

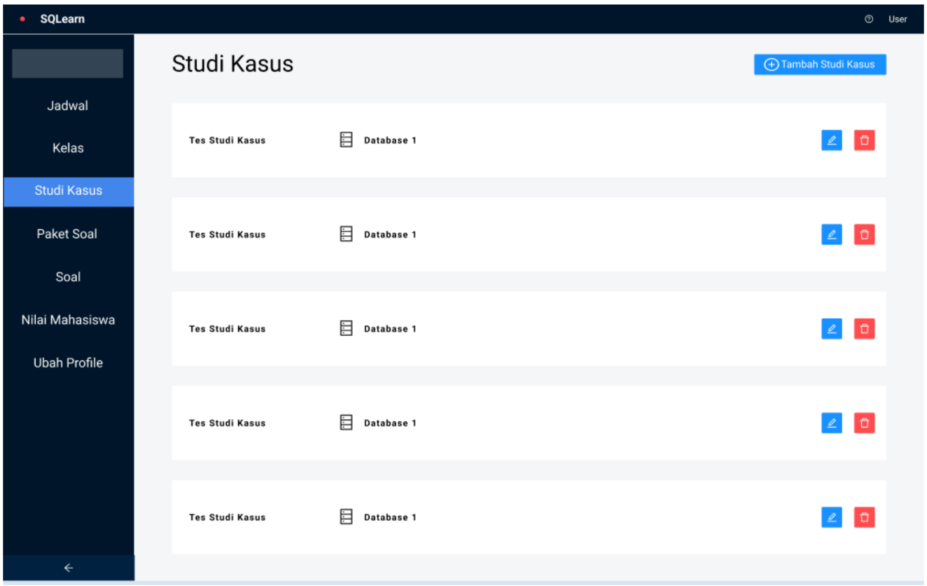


Figure 4. 10 Mockup Case Study List Page

Case study is the database that will be used in questions for students to solve. This is the mockup design of displaying all the database that has been registered.

6. Add Case Study Page Design

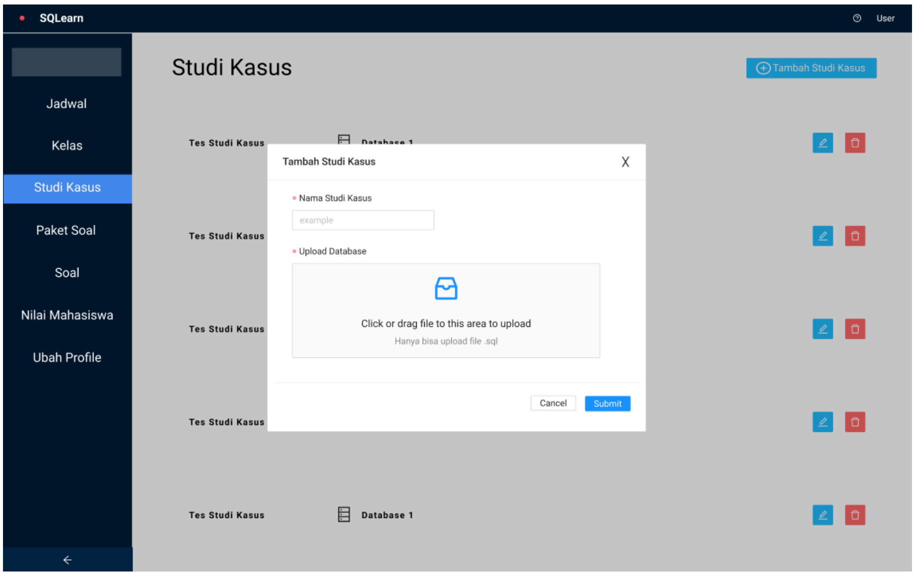


Figure 4. 11 Mockup Add Case Study Page

To add case study, lecturer will need to fill case study name and the SQL file. After it is created, lecturers can create questions that connects to this database.

7. Question Set Page Design

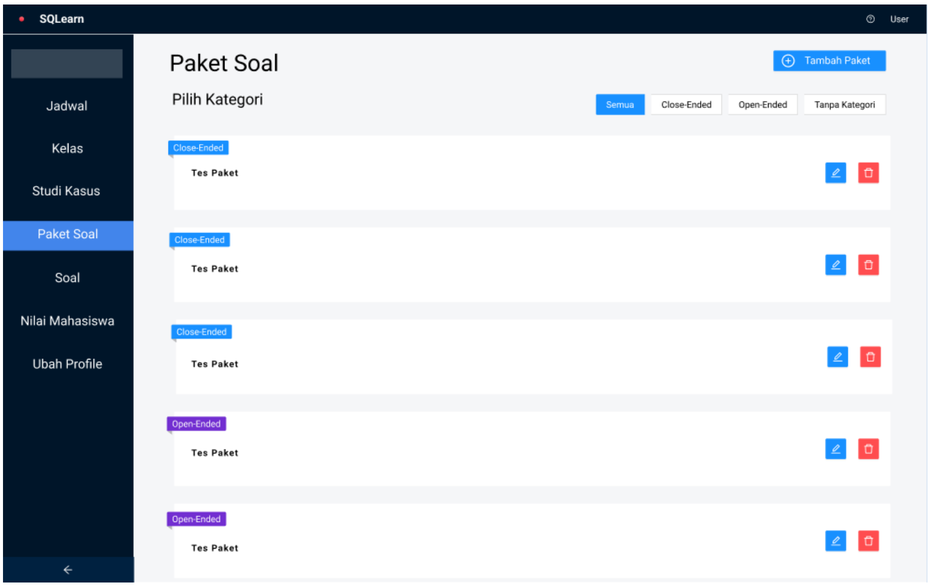


Figure 4. 12 Mockup Question Set Page

Question set is a way to organize questions easily and will be connected to available schedule. This question set defines what are the questions that should be solved by the students in a session.

8. Add Question Set Page Design

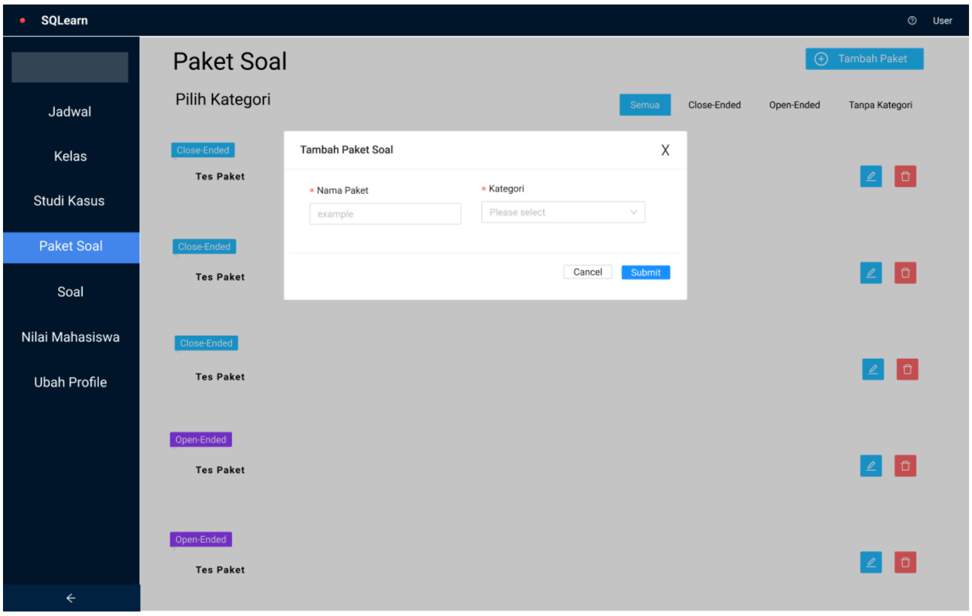
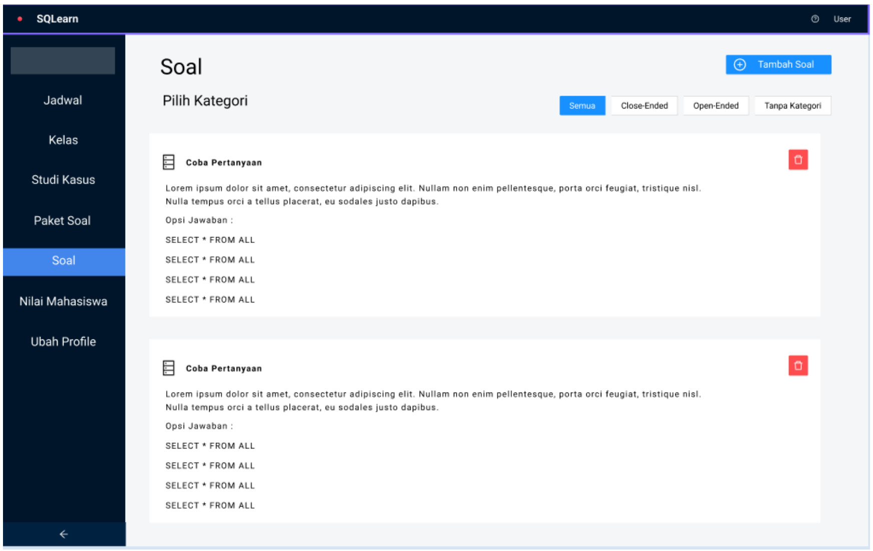


Figure 4. 13 Mockup Add Question Set Page

To create a new question set, lecture just need to input question set name, and choose the category. After that, lecturer can add the question by clicking the detail button

9. Question List Page Design



Coba Pertanyaan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam non enim pellentesque, porta orci feugiat, tristique nisl. Nulla tempus orci a tellus placerat, eu sodales justo dapibus.

Opsi Jawaban :

SELECT * FROM ALL

SELECT * FROM ALL

SELECT * FROM ALL

SELECT * FROM ALL

Coba Pertanyaan

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam non enim pellentesque, porta orci feugiat, tristique nisl. Nulla tempus orci a tellus placerat, eu sodales justo dapibus.

Opsi Jawaban :

SELECT * FROM ALL

SELECT * FROM ALL

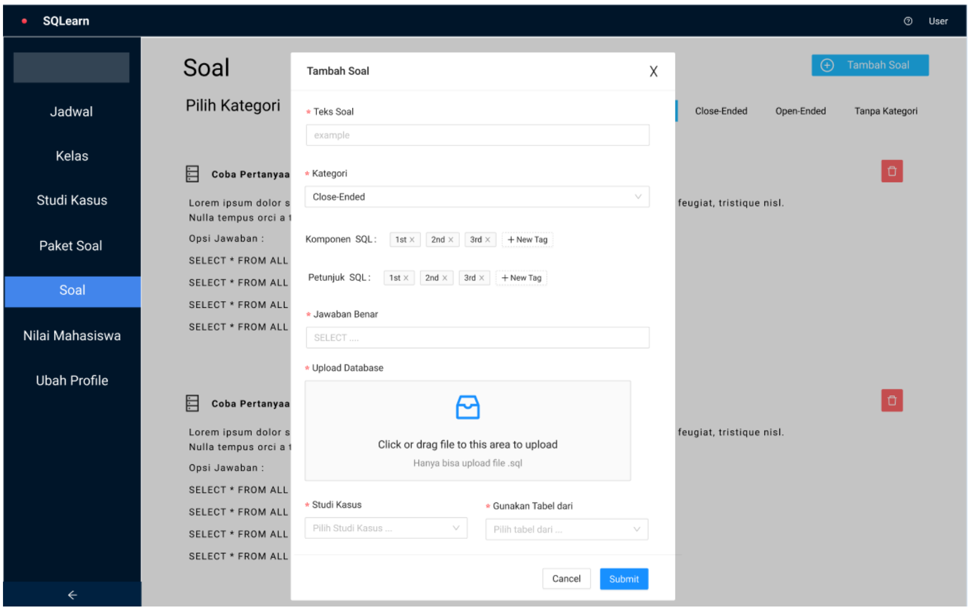
SELECT * FROM ALL

SELECT * FROM ALL

Figure 4. 14 Mockup Question List Page

This page displays all question that has been created by the lecturer. These questions should be assigned in question set first before it can be done by students.

10. Add Question Page Design



Coba Pertanyaa

Lorem ipsum dolor s ... feugiat, tristique nisl.
Nulla tempus orci a t

Opsi Jawaban :

SELECT * FROM ALL
SELECT * FROM ALL
SELECT * FROM ALL
SELECT * FROM ALL

Tambah Soal

* Teks Soal: example

* Kategori: Close-Ended

Komponen SQL : 1st × 2nd × 3rd ×

Petunjuk SQL : 1st × 2nd × 3rd ×

* Jawaban Benar: SELECT ....

* Upload Database

Click or drag file to this area to upload
Hanya bisa upload file .sql

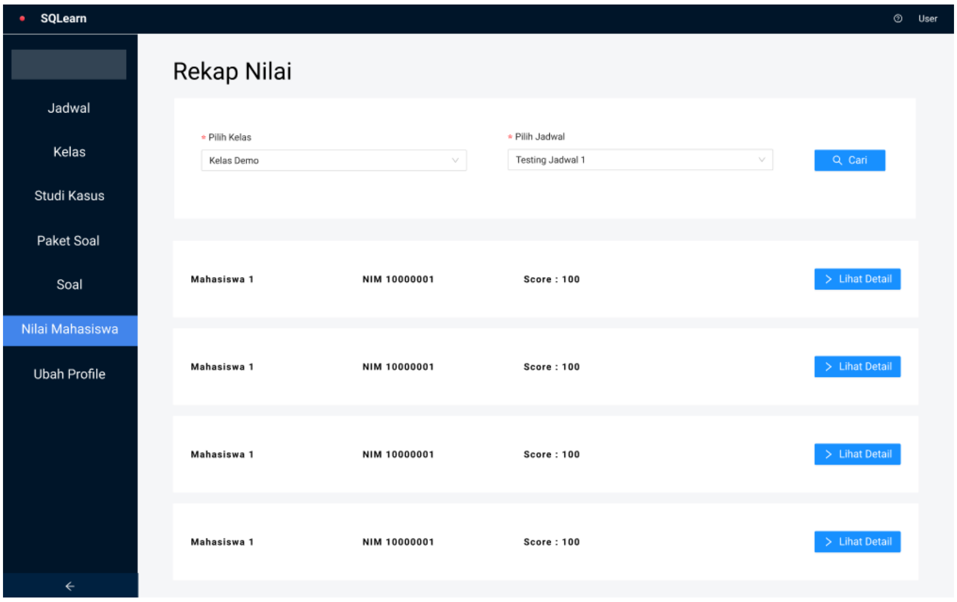
* Studi Kasus: Pilih Studi Kasus ...

* Gunakan Tabel dari: Pilih tabel dari ...

Figure 4. 15 Mockup Add Question Page

To create a new question, lecturer is required to input question text, choose category, define the SQL parts and SQL Hints, the SQL answer, image of expected result, choosing the case study and the tables

11. Score Record Page Design



* Pilih Kelas: Kelas Demo

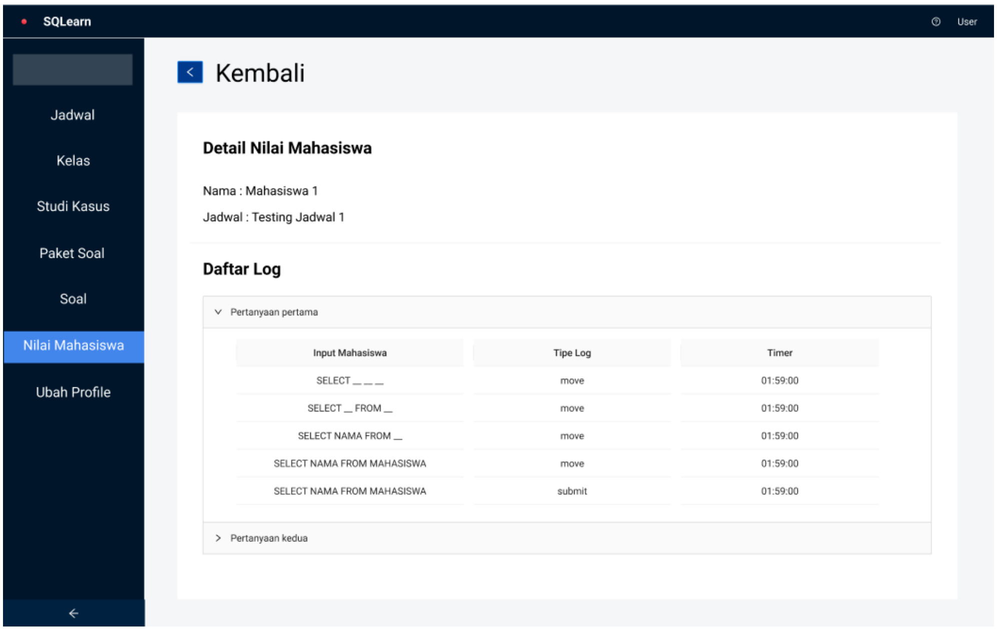
* Pilih Jadwal: Testing Jadwal 1

Mahasiswa 1	NIM 10000001	Score : 100
Mahasiswa 1	NIM 10000001	Score : 100
Mahasiswa 1	NIM 10000001	Score : 100
Mahasiswa 1	NIM 10000001	Score : 100

Figure 4. 16 Mockup Score Record Page

After students done the practice, lecturer can see their scores in this page.

12. Detail Score Record Page Design



Detail Nilai Mahasiswa

Nama : Mahasiswa 1

Jadwal : Testing Jadwal 1

Daftar Log

Pertanyaan pertama

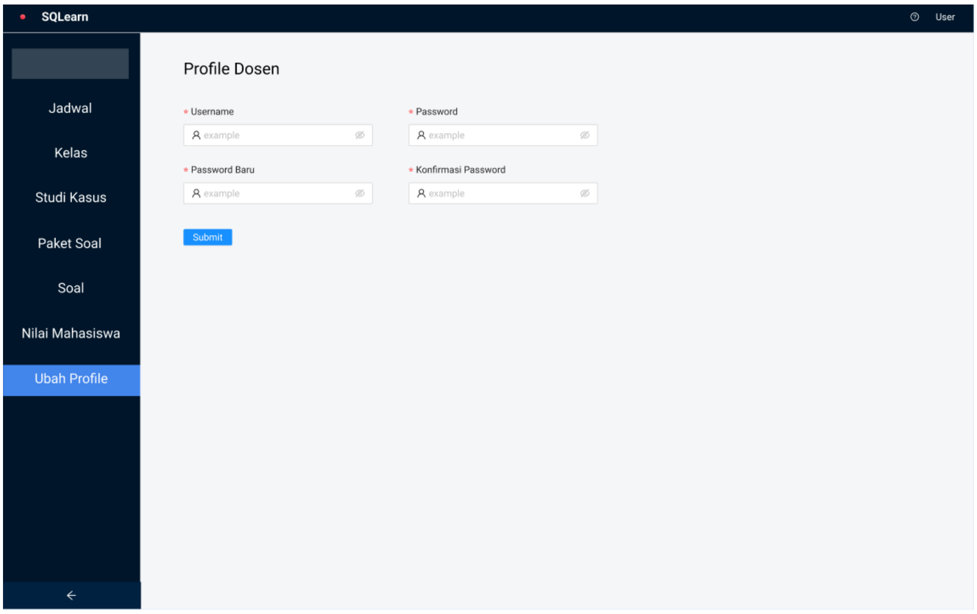
Input Mahasiswa	Tipe Log	Timer
SELECT ___ ___ ___	move	01:59:00
SELECT ___ FROM ___	move	01:59:00
SELECT NAMA FROM ___	move	01:59:00
SELECT NAMA FROM MAHASISWA	move	01:59:00
SELECT NAMA FROM MAHASISWA	submit	01:59:00

Pertanyaan kedua

Figure 4. 17 Mockup Detail Score Record Page

Furthermore, lecturers can see the logs of each students drag-and-drop process for each question that has been set in a schedule.

13. Update Profile Lecturer Page Design



Profile Dosen

* Username: example

* Password: example

* Password Baru: example

* Konfirmasi Password: example

Figure 4. 18 Mockup Update Profile Lecturer Page

Lecturer can change their username and password in this page

14. Student Available Schedule Page Design

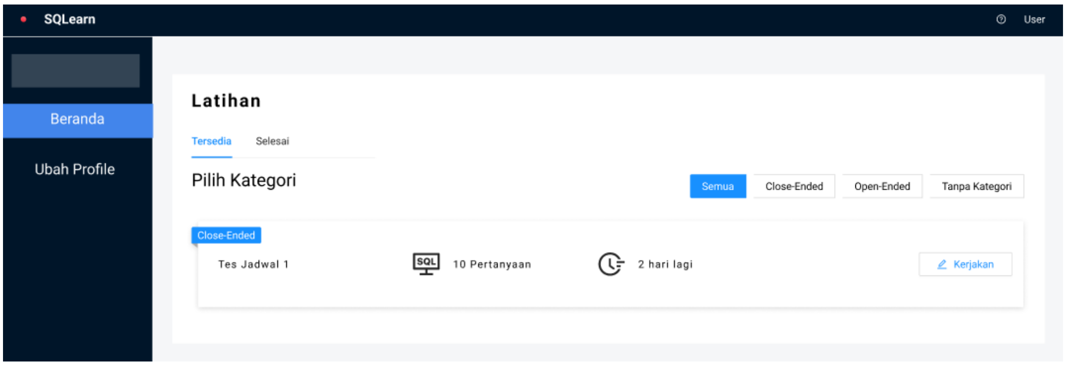
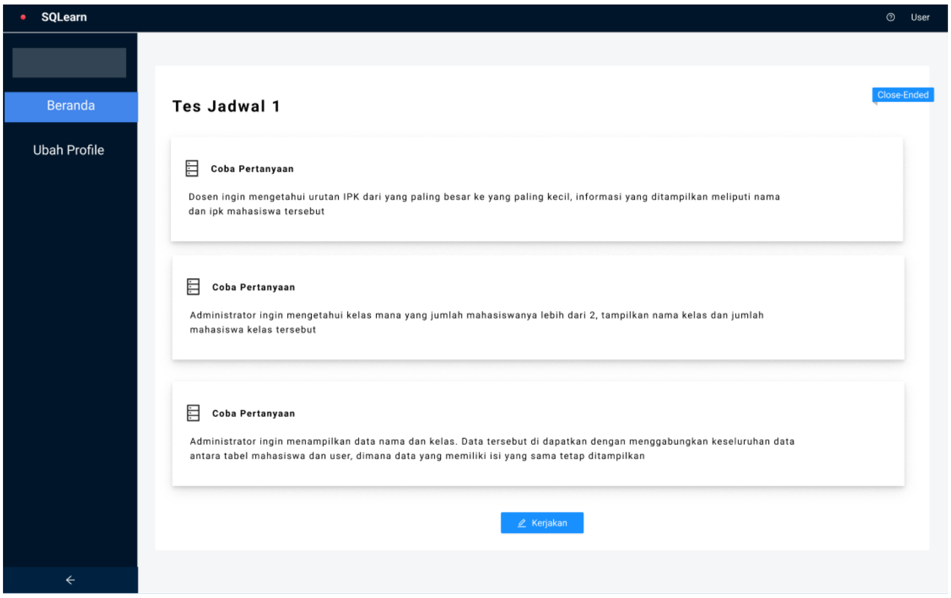


Figure 4. 19 Mockup Student Available Schedule Page Design

This page is limited to student role only. After new schedule has been created by the lecturer, students of selected class can see the exercise they need to do in this page.

15. Student Practice Set Page Design



Dosen ingin mengetahui urutan IPK dari yang paling besar ke yang paling kecil, informasi yang ditampilkan meliputi nama dan ipk mahasiswa tersebut

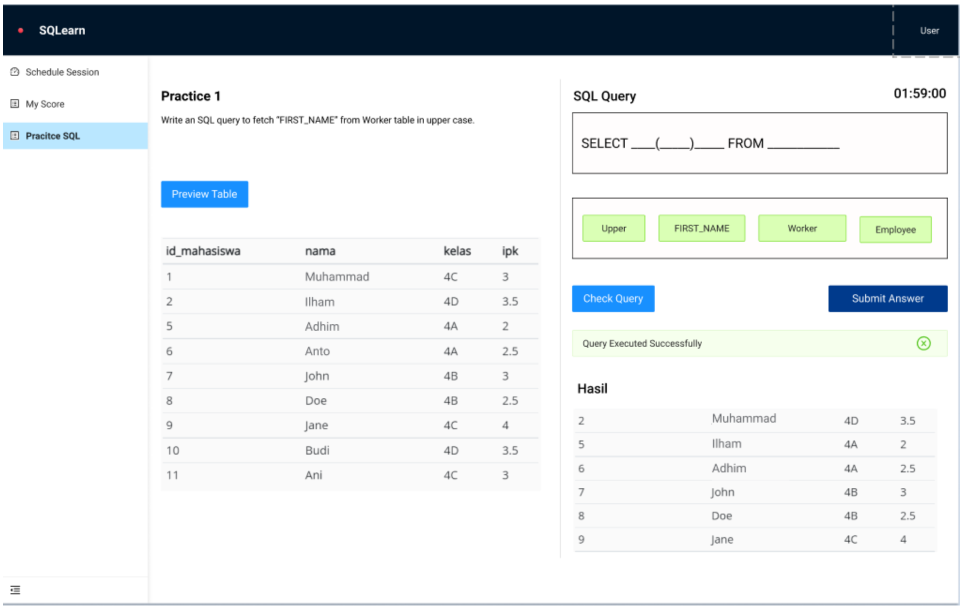
Administrator ingin mengetahui kelas mana yang jumlah mahasiswanya lebih dari 2, tampilkan nama kelas dan jumlah mahasiswa kelas tersebut

Administrator ingin menampilkan data nama dan kelas. Data tersebut di dapatkan dengan menggabungkan keseluruhan data antara tabel mahasiswa dan user, dimana data yang memiliki isi yang sama tetap ditampilkan

Figure 4. 20 Mockup Student Practice Set Page

After clicking the ‘Kerjakan’ button, students can see list of questions they need to solve in this schedule.

16. Student Practice Page Design



Practice 1

Write an SQL query to fetch "FIRST_NAME" from Worker table in upper case.

id_mahasiswa	nama	kelas	ipk
1	Muhammad	4C	3
2	Ilham	4D	3.5
5	Adhim	4A	2
6	Anto	4A	2.5
7	John	4B	3
8	Doe	4B	2.5
9	Jane	4C	4
10	Budi	4D	3.5
11	Ani	4C	3

SQL Query

SELECT ___(___)___ FROM ________

Query Executed Successfully

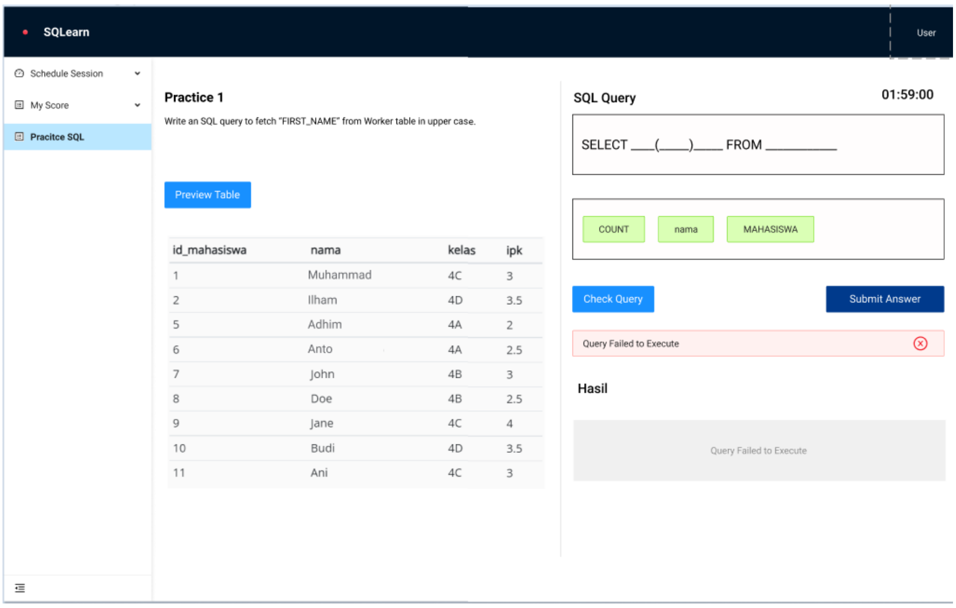
Hasil

2	Muhammad	4D	3.5
5	Ilham	4A	2
6	Adhim	4A	2.5
7	John	4B	3
8	Doe	4B	2.5
9	Jane	4C	4

Figure 4. 21 Mockup - Students Practice submit the correct query Page

The mockup of user interface here is displayed for students while doing the exercise each question. They need to drag-and-drop the SQL options until it is valid.

17. Student Practice Page Design



Practice 1

Write an SQL query to fetch "FIRST_NAME" from Worker table in upper case.

id_mahasiswa	nama	kelas	ipk
1	Muhammad	4C	3
2	Ilham	4D	3.5
5	Adhim	4A	2
6	Anto	4A	2.5
7	John	4B	3
8	Doe	4B	2.5
9	Jane	4C	4
10	Budi	4D	3.5
11	Ani	4C	3

SQL Query

SELECT ___(___)___ FROM ________

Query Failed to Execute

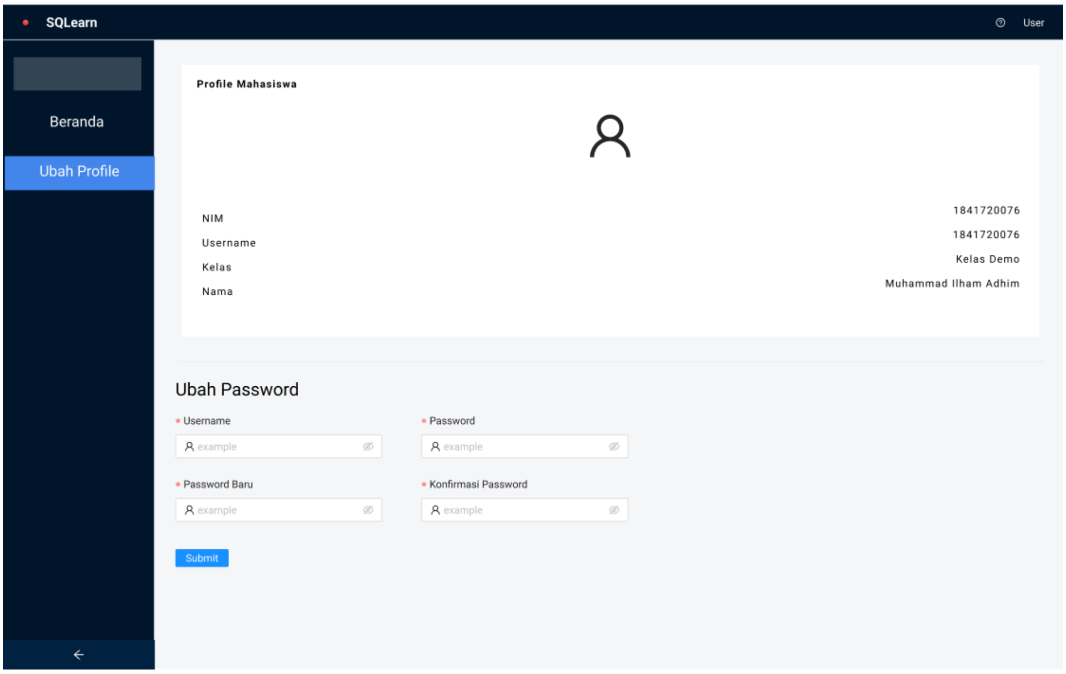
Hasil

Query Failed to Execute

Figure 4. 22 Mockup - Students Practice submit the wrong query Page

If the submitted answer or tested query is false, students cannot proceed to next question. Therefore, they need to do it until the question is answered correctly.

18. Student Change Profile Page Design



Profile Mahasiswa

NIM	1841720076
Username	1841720076
Kelas	Kelas Demo
Nama	Muhammad Ilham Adhim

Ubah Password

* Username: example

* Password: example

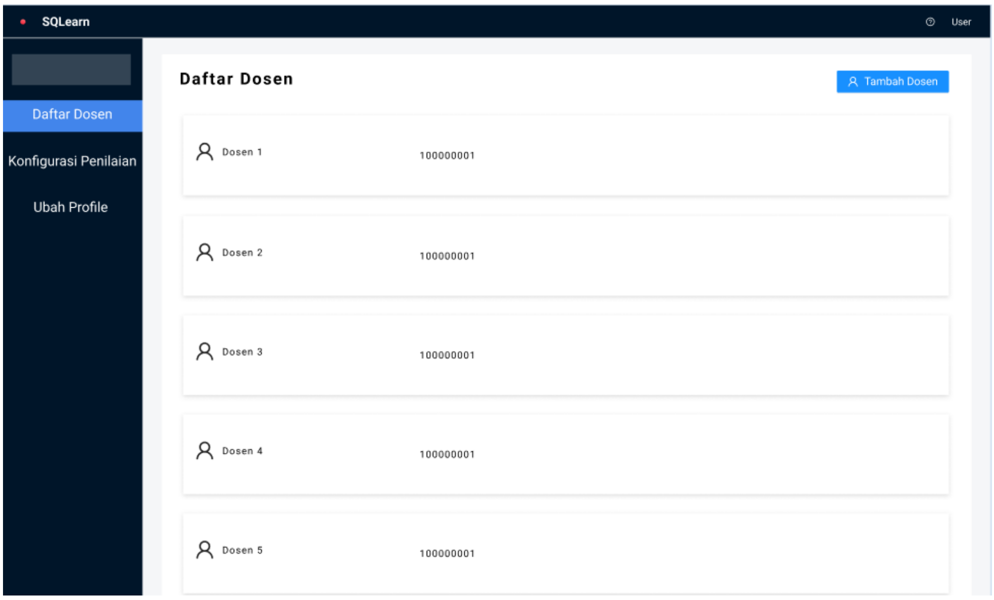
* Password Baru: example

* Konfirmasi Password: example

Figure 4. 23 Mockup Change Profile Page

Students can change their username and password in this page.

19. Admin - Lecturer List Page Design



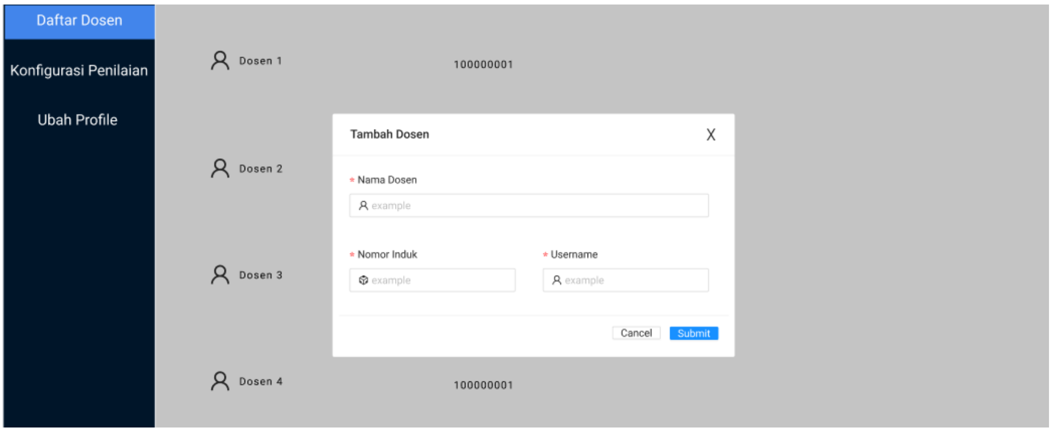
Daftar Dosen

Dosen 1	100000001
Dosen 2	100000001
Dosen 3	100000001
Dosen 4	100000001
Dosen 5	100000001

Figure 4. 24 Admin - Lecturer List Page

This page is limited to admin role only, they can see registered lecturers in the system.

20. Admin – Add Lecturer Page Design

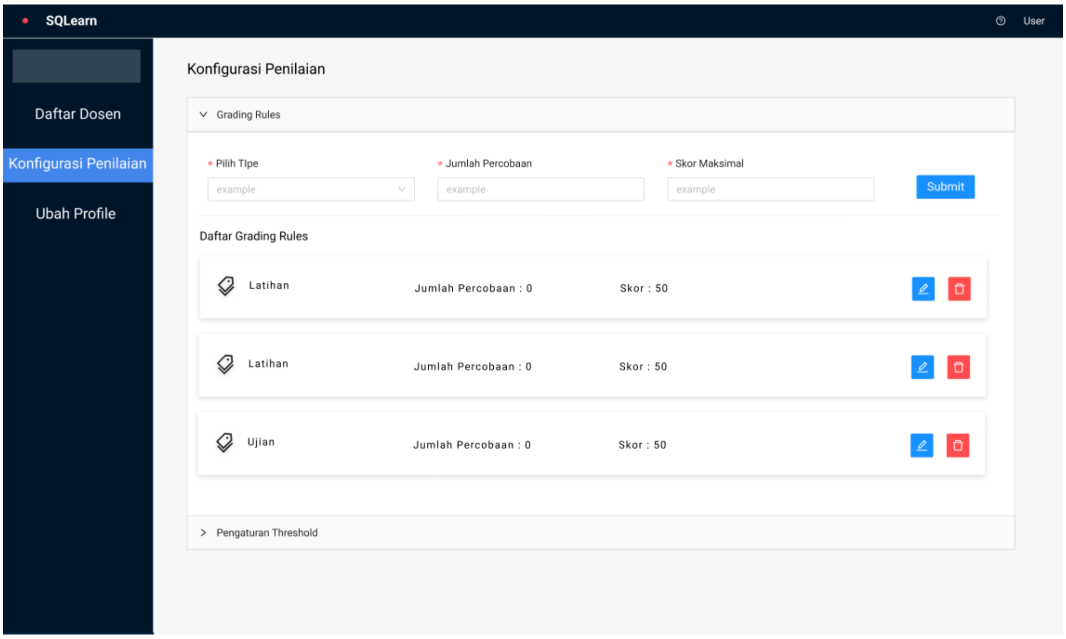


Dosen 1 100000001
Dosen 2
Dosen 3
Dosen 4 100000001

Figure 4. 25 Mockup Admin – Add Lecturer Page

Admins can create a new lecturer, new registered lecturer can login with username credentials that is filled in this form.

21. Admin – Grading Configuration Page Design

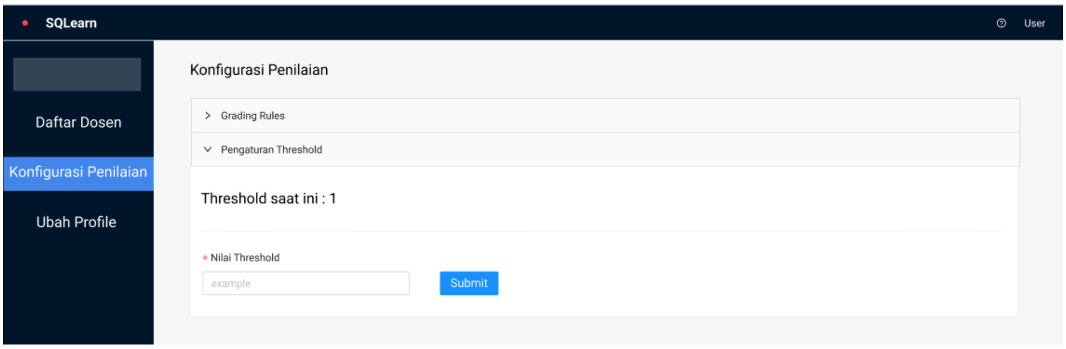


Latihan Jumlah Percobaan : 0 Skor : 50
Latihan Jumlah Percobaan : 0 Skor : 50
Ujian Jumlah Percobaan : 0 Skor : 50

Figure 4. 26 Mockup Admin – Grading Configuration Page

Admins can define the grading rules. It is the threshold for the system to calculate students score each question, which will be accumulated through all questions.

22. Threshold Configuration Page Design



Konfigurasi Penilaian

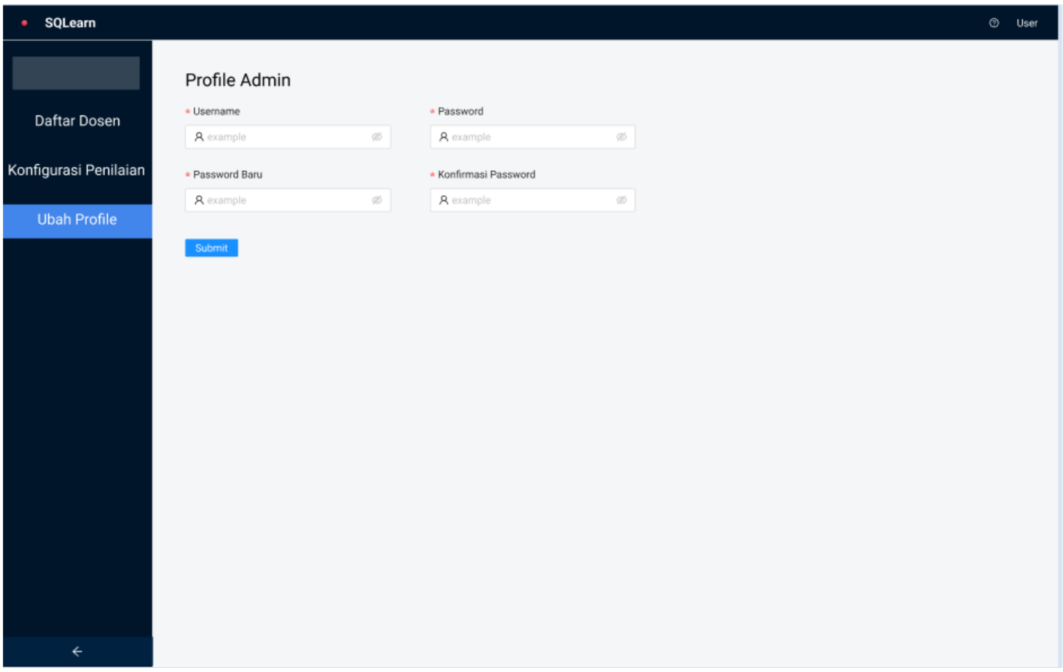
Threshold saat ini : 1

* Nilai Threshold: example

Figure 4. 27 Threshold Configuration Page

The threshold of similarity value. Currently, the threshold value is 1. Therefore, the constructed SQL by students must be exactly the same with defined answer by the lecturer.

23. Admin Change Profile Page Design



Profile Admin

* Username: example

* Password: example

* Password Baru: example

* Konfirmasi Password: example

Figure 4. 28 Admin Change Profile Page

Admins can change their username and password.

CHAPTER V. IMPLEMENTATION AND TESTING

5.1. System Implementation

The system implementation starts from designing the database and being followed up to converting the mockup design to User Interface codes in frontend part using React JS combined with Next JS. The first iteration was using mockup API for the data. After user interface has been developed and the logic is created, then, it's time to configure the database's table and structure and settle the backend part with Node JS with Express JS as the micro-framework.

5.2. Database Implementation

The database implementation is as attached as follows.

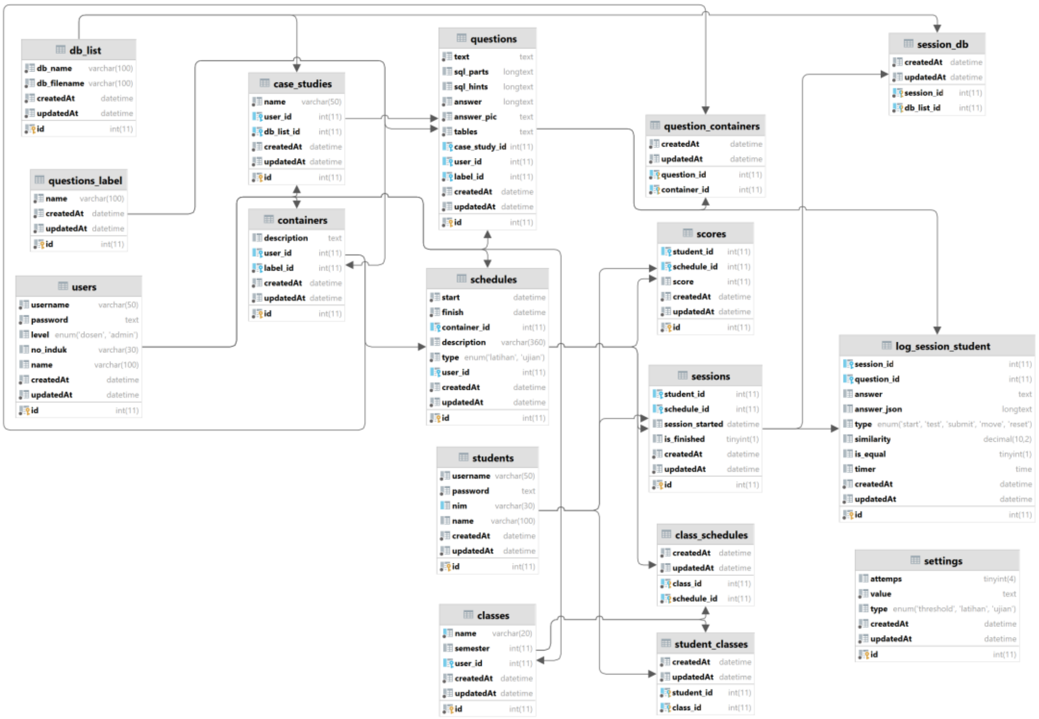


Figure 5. 1 Database Implementation

Each table structure is as listed in Chapter IV with additional columns createdAt and updatedAt. This happened due to the usage of library ORM (Object

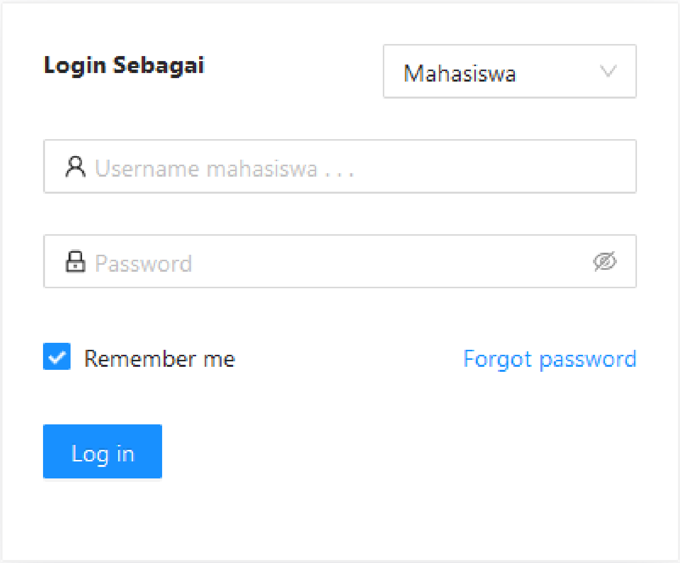
Relational Mapping) named Sequelize. This library is to help connect backend with database, so that the data fetching in backend part becomes easier to handle.

5.3. Design Interface Implementation

The design interface implementation of the application is as structured previously based on the mockups. Below are listed the designs of each respective feature that supports how the system works for available roles. Such as role lecturer, student, and admins.

5.3.1. Login Page Interface

The landing page of SQLearn is the login. To proceed to the system, user will have to login with their credentials and respective role. Each role will have different features provided in the system.



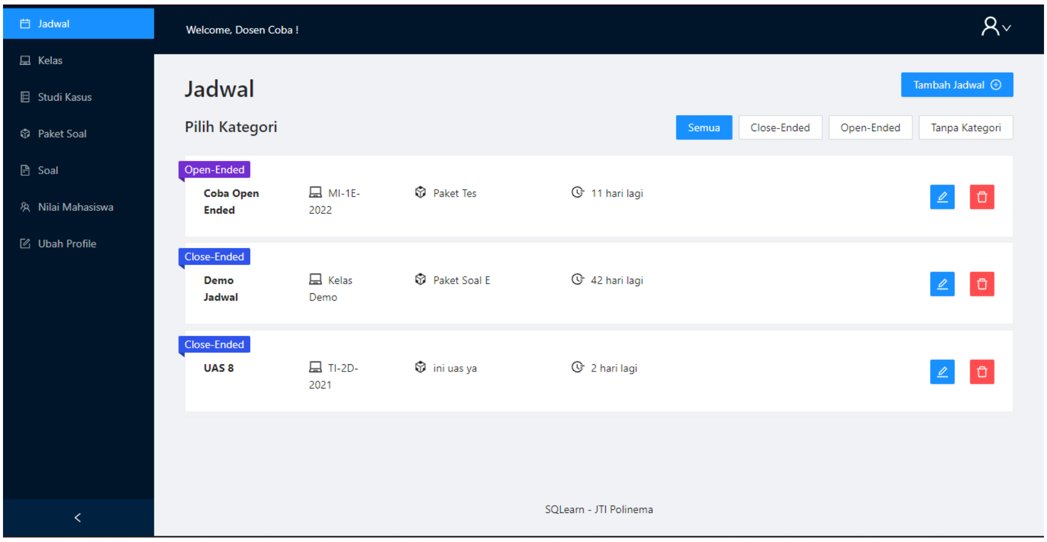
Login Sebagai: Mahasiswa

☑ Remember me

Figure 5. 2 Login Page Implementation

5.3.2. Lecturer Schedule Interface

In this page, lecturers are provided schedule list that are available for all classes. As this research focuses on close-ended approach, it is provided filtering for close-ended and open-ended schedule so that it will be easier to categorize. There is a button ‘Tambah Jadwal’ to make a new schedule.



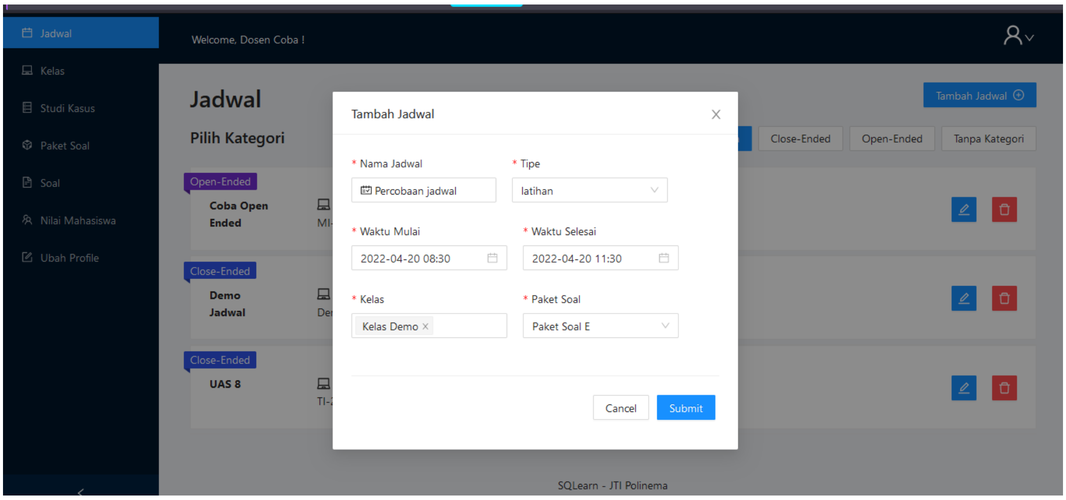
Open-Ended Coba Open Ended — MI-1E-2022 — Paket Tes — 11 hari lagi

Close-Ended Demo Jadwal — Kelas Demo — Paket Soal E — 42 hari lagi

Close-Ended UAS 8 — TI-2D-2021 — ini uas ya — 2 hari lagi

Figure 5. 3 Schedule Page Implementation – Lecturer

In creating a new schedule, lecturer will need to fill provide some information, assign class that has access to the schedule, and the question set. Once question set is defined, the question within that set will be randomized so that each student may get different question order compared to their classmates.



Tambah Jadwal

* Nama Jadwal: Percobaan jadwal

* Tipe: latihan

* Waktu Mulai: 2022-04-20 08:30

* Waktu Selesai: 2022-04-20 11:30

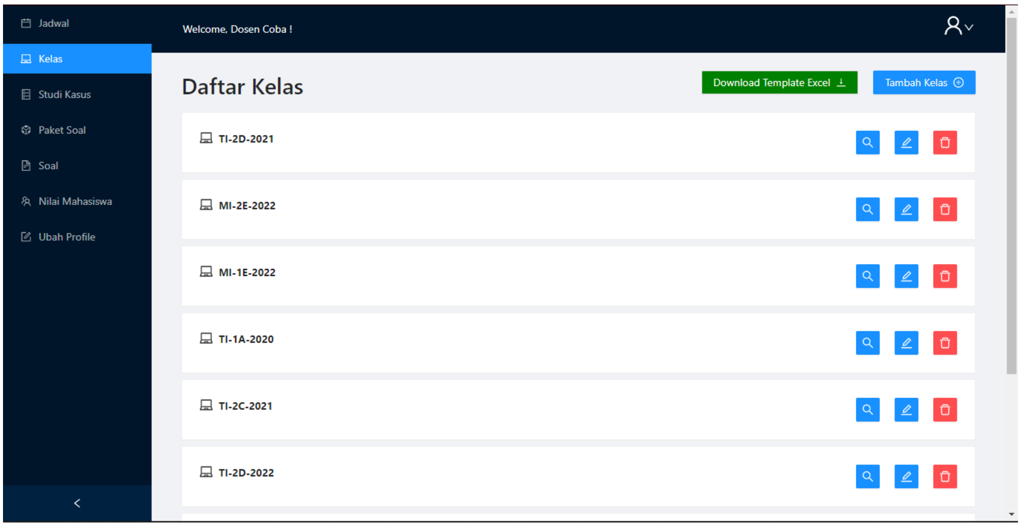
* Kelas: Kelas Demo

* Paket Soal: Paket Soal E

Figure 5. 4 Add schedule - Lecturer

5.3.3. Classes Interface

In this page, lecturers are given the list of registered classes that are being taught by the lecturer. In detail page, there are provided student list on each class as well.

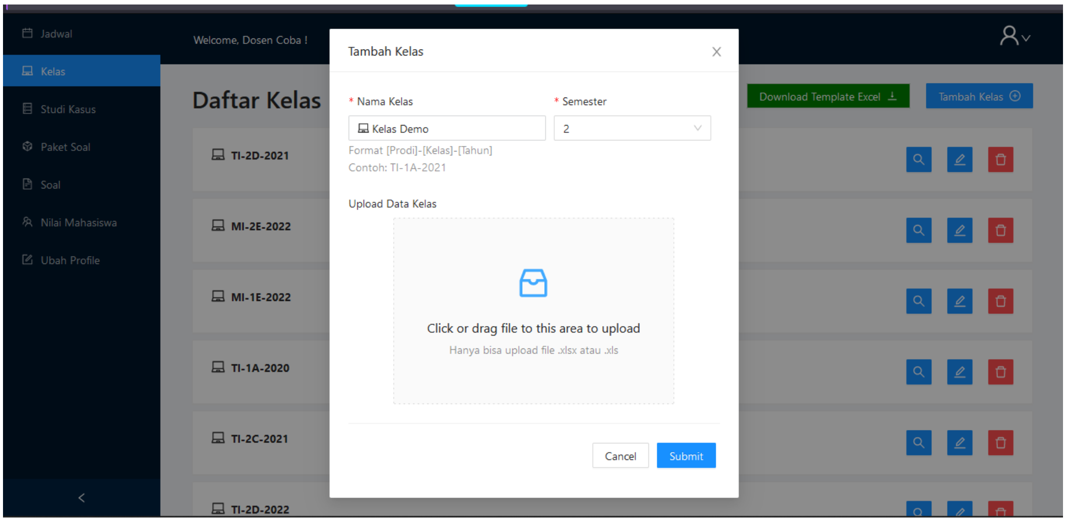


Daftar Kelas

- TI-2D-2021
- MI-2E-2022
- MI-1E-2022
- TI-1A-2020
- TI-2C-2021
- TI-2D-2022

Figure 5. 5 Class Page Implementation

Lecturer are also capable of adding new class. To assign students in the class faster, the system provides an option to upload excel file.



Tambah Kelas

* Nama Kelas: Kelas Demo

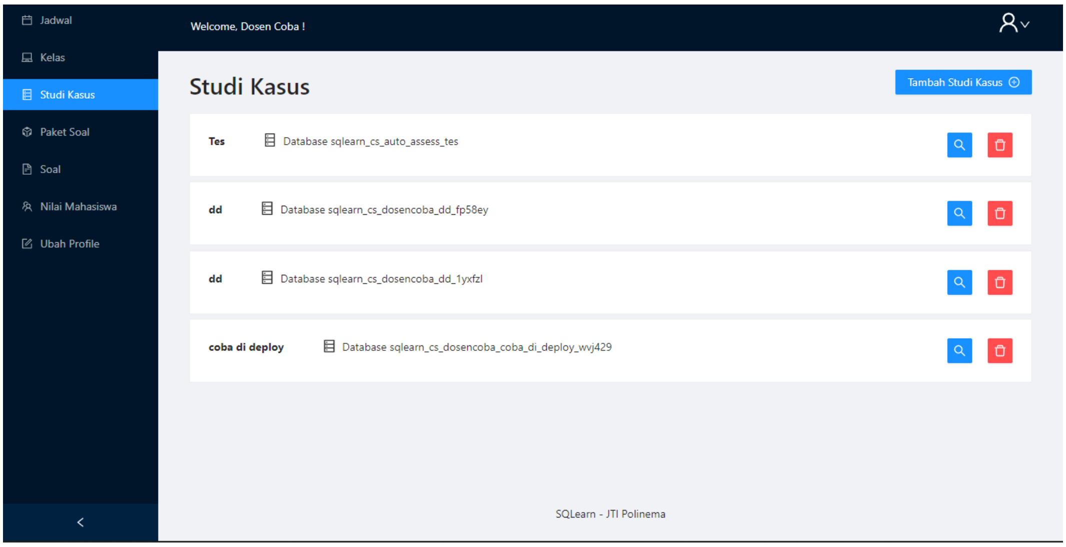
Format [Prodi]-[Kelas]-[Tahun]
Contoh: TI-1A-2021

* Semester: 2

Figure 5. 6 Add class – Lecturer

5.3.4. Case Studies Interface

In Case Studies page, lecturers are given the active database that are available in the system. These databases are used to be referenced when answering the questions. To add new case study, lecturers are required to upload .SQL file.



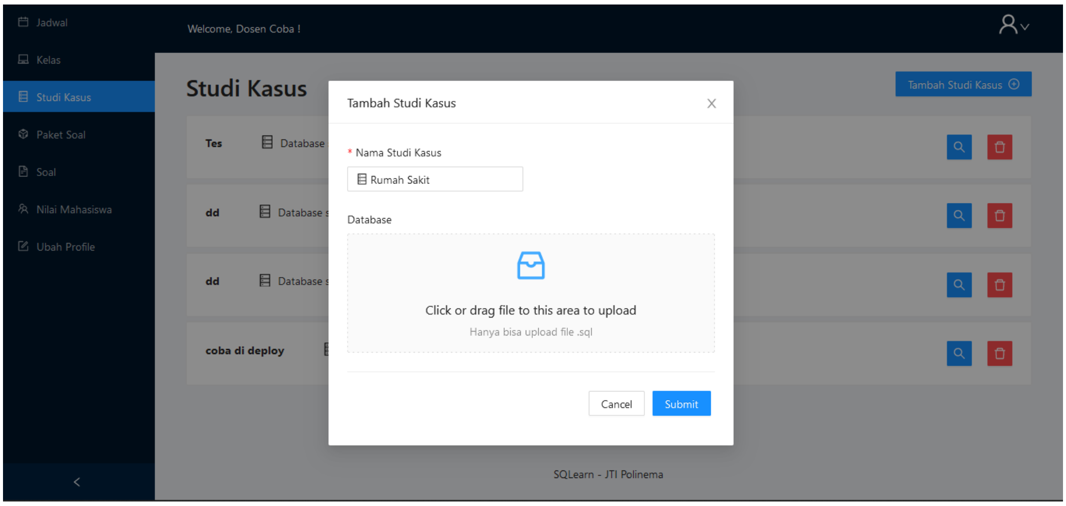
Welcome, Dosen Coba !

Studi Kasus

Tes	Database sqlearn_cs_auto_assess_tes
dd	Database sqlearn_cs_dosencoba_dd_fp58ey
dd	Database sqlearn_cs_dosencoba_dd_1yxfzl
coba di deploy	Database sqlearn_cs_dosencoba_coba_di_deploy_wvj429

SQLearn - JTI Polinema

Figure 5. 7 Case Study Page Implementation



Welcome, Dosen Coba !

Studi Kasus

Tambah Studi Kasus

* Nama Studi Kasus

Rumah Sakit

Database

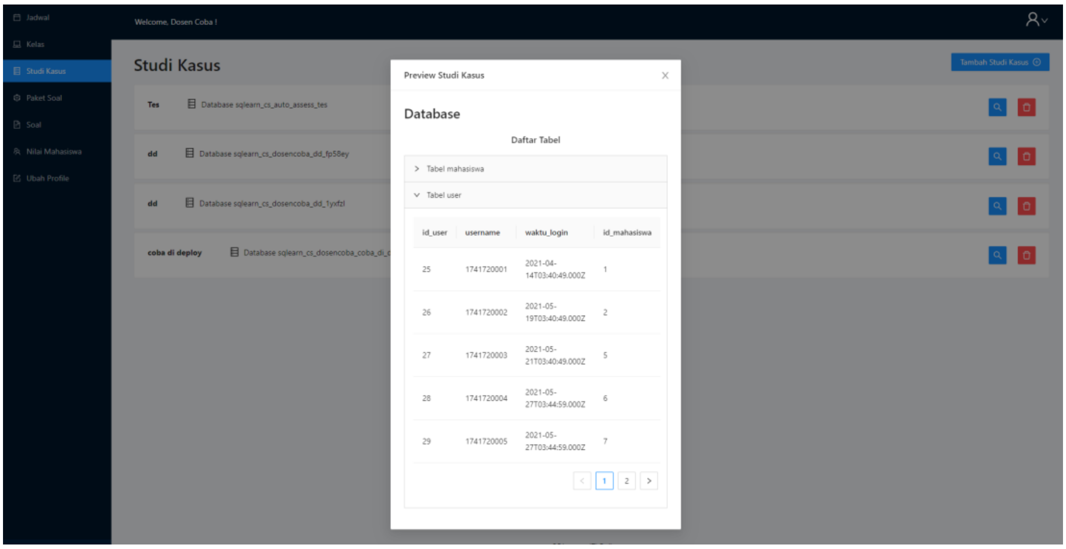
Click or drag file to this area to upload

Hanya bisa upload file .sql

SQLearn - JTI Polinema

Figure 5. 8 Add Case Study - Lecturer

In addition to that, the lecturer also can preview available table of each case study alongside with the records.



Studi Kasus

Tes Database sqlearn_cs_auto_assess_tes

dd Database sqlearn_cs_dosencoba_dd_fp58ey

dd Database sqlearn_cs_dosencoba_dd_1yxfzl

coba di deploy Database sqlearn_cs_dosencoba_coba_di_d

Preview Studi Kasus

Database

Daftar Tabel

> Tabel mahasiswa

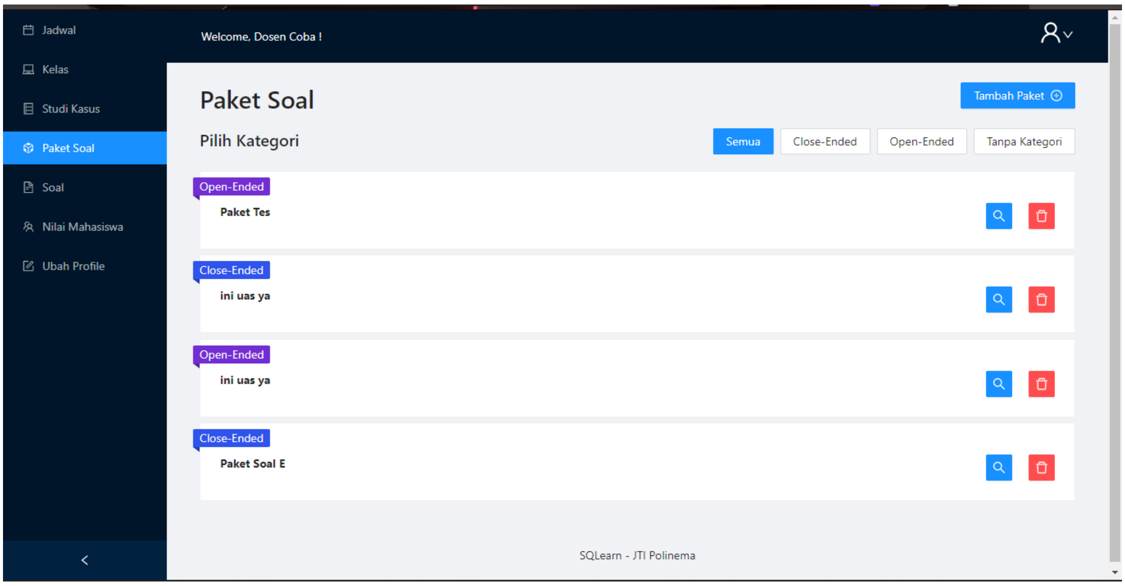
> Tabel user

id_user	username	waktu_login	id_mahasiswa
25	1741720001	2021-04-14T03:40:49.000Z	1
26	1741720002	2021-05-19T03:40:49.000Z	2
27	1741720003	2021-05-21T03:40:49.000Z	5
28	1741720004	2021-05-27T03:44:59.000Z	6
29	1741720005	2021-05-27T03:44:59.000Z	7

Figure 5. 9 Preview Case Study - Lecturer

5.3.5. Question Sets Interface

In this page, lecturers are able to identify question set that will be used while creating schedule, and just like the schedule, it is given filtering between close-ended and open-ended as well.



Paket Soal

Pilih Kategori

Open-Ended **Paket Tes**

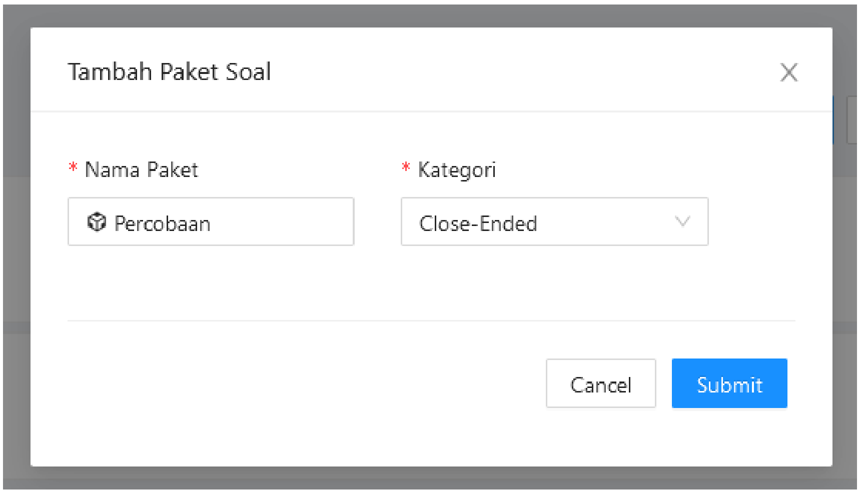
Close-Ended **ini uas ya**

Open-Ended **ini uas ya**

Close-Ended **Paket Soal E**

Figure 5. 10 Question Set Page Implementation

To create a new question set, it is quite straightforward. Lecturer will only be required to input question set name and its category.



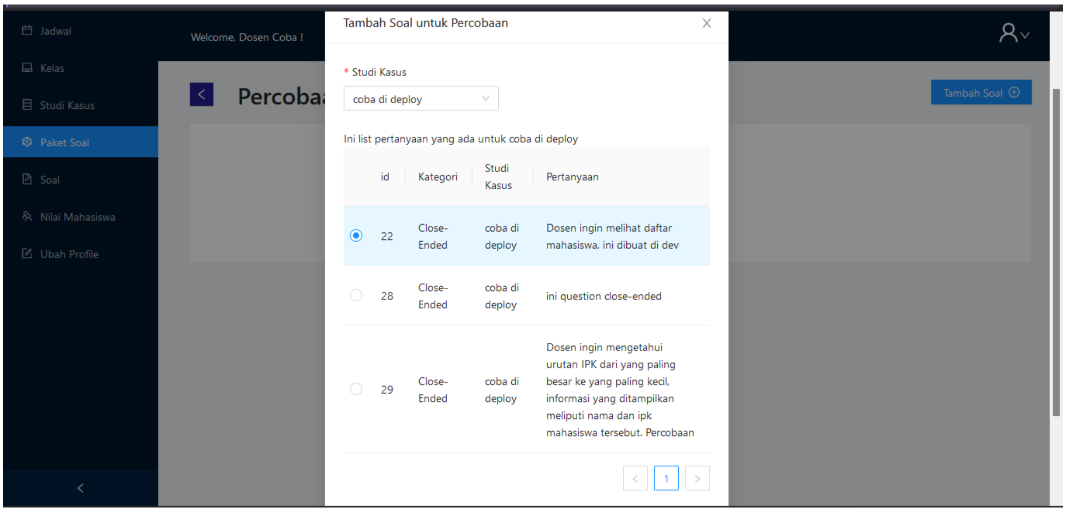
Tambah Paket Soal

* Nama Paket: Percobaan

* Kategori: Close-Ended

Figure 5. 11 Create New Question Set – Lecturer

After new question set has been created, In the question set list, the lecturer can select available question and add it to newly created question set. To do that, lecturer needs to choose one of many case studies that have been created.



Tambah Soal untuk Percobaan

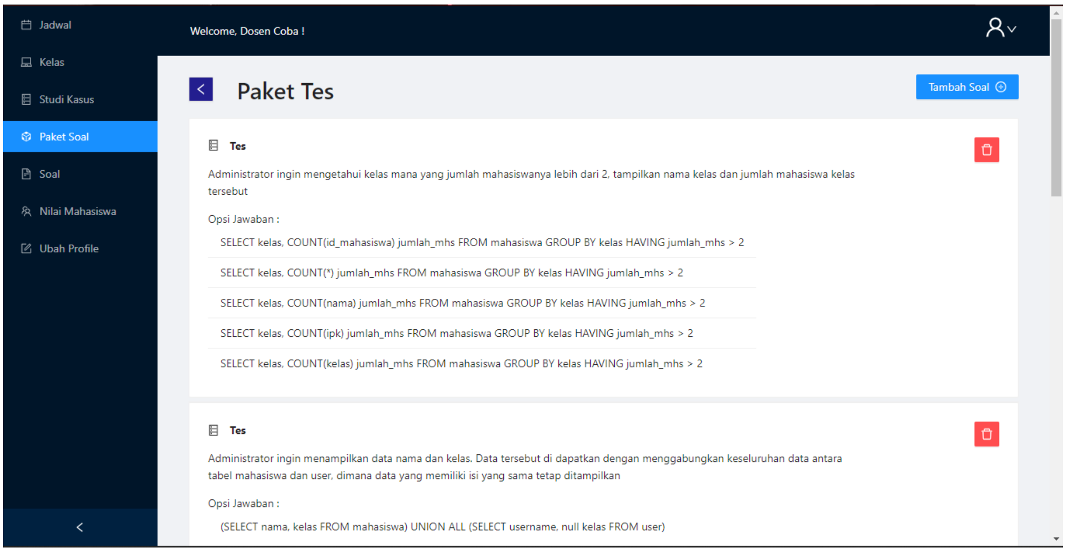
* Studi Kasus: coba di deploy

Ini list pertanyaan yang ada untuk coba di deploy

	id	Kategori	Studi Kasus	Pertanyaan
●	22	Close-Ended	coba di deploy	Dosen ingin melihat daftar mahasiswa, ini dibuat di dev
○	28	Close-Ended	coba di deploy	ini question close-ended
○	29	Close-Ended	coba di deploy	Dosen ingin mengetahui urutan IPK dari yang paling besar ke yang paling kecil, informasi yang ditampilkan meliputi nama dan ipk mahasiswa tersebut. Percobaan

Figure 5. 12 Select Question to Question Set – Lecturer

After some questions are added to the question set, lecturers can see the question list within that question set and may remove the question if requirement changes.



Tes

Administrator ingin mengetahui kelas mana yang jumlah mahasiswanya lebih dari 2, tampilkan nama kelas dan jumlah mahasiswa kelas tersebut

Opsi Jawaban :

```
SELECT kelas, COUNT(id_mahasiswa) jumlah_mhs FROM mahasiswa GROUP BY kelas HAVING jumlah_mhs > 2
SELECT kelas, COUNT(*) jumlah_mhs FROM mahasiswa GROUP BY kelas HAVING jumlah_mhs > 2
SELECT kelas, COUNT(nama) jumlah_mhs FROM mahasiswa GROUP BY kelas HAVING jumlah_mhs > 2
SELECT kelas, COUNT(ipk) jumlah_mhs FROM mahasiswa GROUP BY kelas HAVING jumlah_mhs > 2
SELECT kelas, COUNT(kelas) jumlah_mhs FROM mahasiswa GROUP BY kelas HAVING jumlah_mhs > 2
```

Tes

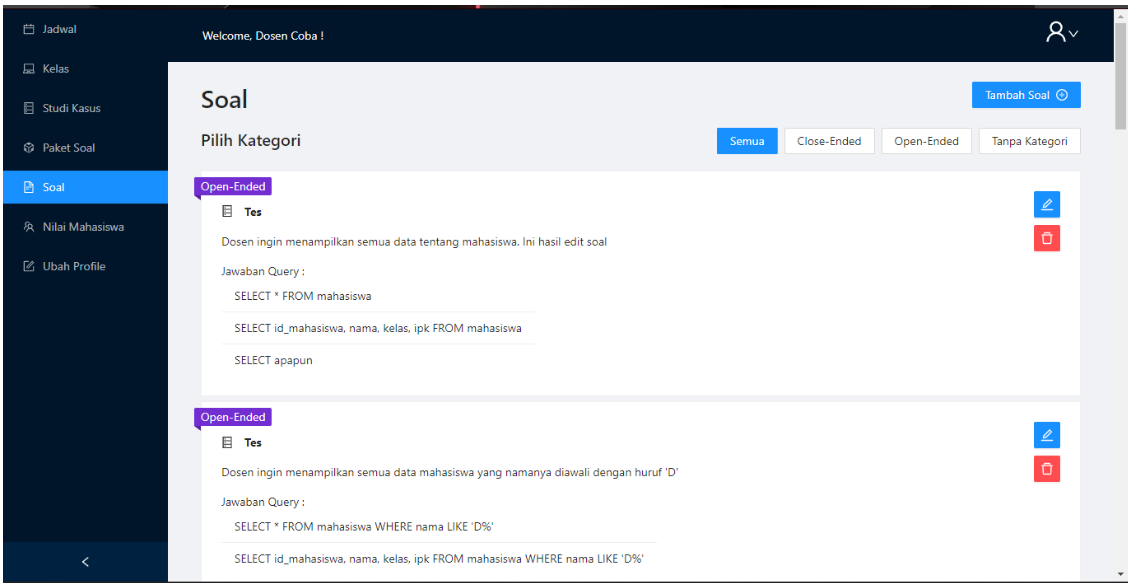
Administrator ingin menampilkan data nama dan kelas. Data tersebut di dapatkan dengan menggabungkan keseluruhan data antara tabel mahasiswa dan user, dimana data yang memiliki isi yang sama tetap ditampilkan

Opsi Jawaban :

```
(SELECT nama, kelas FROM mahasiswa) UNION ALL (SELECT username, null kelas FROM user)
```

5.3.6. Question List Interface

In question list page, the system provides all questions from any case studies that have been created previously. When lecturer wants to create a new question, they can click the button ‘Tambah Soal’ and then it will display a form.



Tes

Dosen ingin menampilkan semua data tentang mahasiswa. Ini hasil edit soal

Jawaban Query :

```
SELECT * FROM mahasiswa
SELECT id_mahasiswa, nama, kelas, ipk FROM mahasiswa
SELECT apapun
```

Tes

Dosen ingin menampilkan semua data mahasiswa yang namanya diawali dengan huruf 'D'

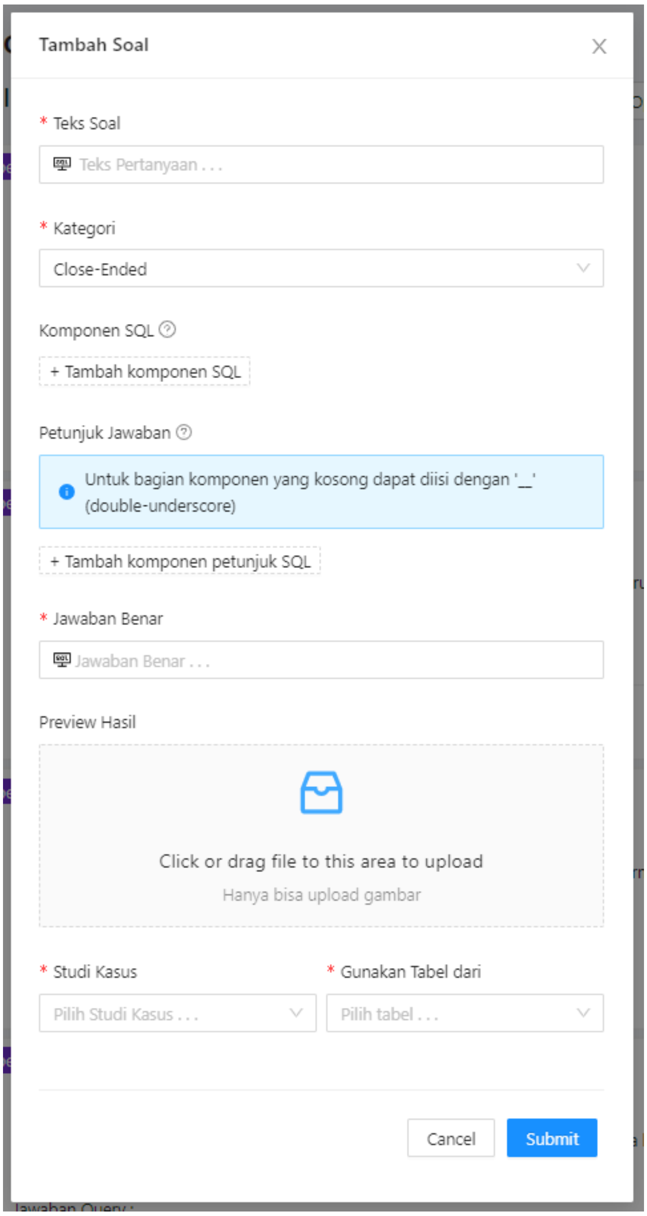
Jawaban Query :

```
SELECT * FROM mahasiswa WHERE nama LIKE 'D%'
SELECT id_mahasiswa, nama, kelas, ipk FROM mahasiswa WHERE nama LIKE 'D%'
```

Figure 5. 13 Question List Page Implementation

Once the form is opened, lecturer will need to fill some information such as:

1. Text question
2. Category. This research is focused on the close-ended approach
3. SQL parts. These parts will be the component that will be dragged and dropped
4. SQL Hints. The component that will provide some hint to the students while answering this particular question
5. Correct answer. Since lecturer can add incorrect SQL parts to answer the question, lecturer needs to add the correct answer as well.
6. Preview image. It is the expected table outcome when the query is built
7. Case study. Each question will need to be connected with one case study
8. Tables. There are multiple tables in a case study, therefore it is required to define which table to be used for this particular question



Tambah Soal

* Teks Soal

Teks Pertanyaan . . .

* Kategori

Close-Ended

Komponen SQL

+ Tambah komponen SQL

Petunjuk Jawaban

Untuk bagian komponen yang kosong dapat diisi dengan '__' (double-underscore)

+ Tambah komponen petunjuk SQL

* Jawaban Benar

Jawaban Benar . . .

Preview Hasil

Click or drag file to this area to upload

Hanya bisa upload gambar

* Studi Kasus

Pilih Studi Kasus . . .

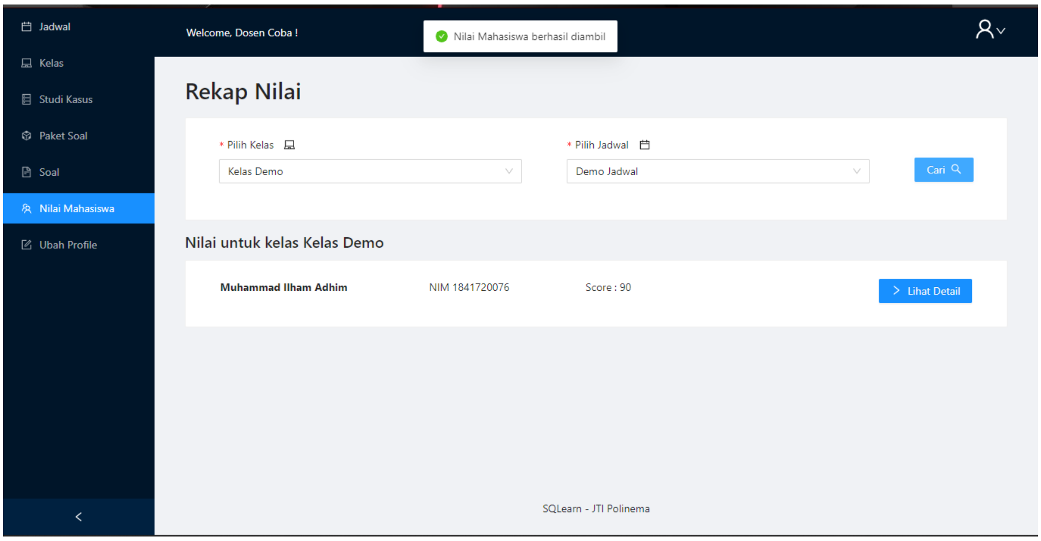
* Gunakan Tabel dari

Pilih tabel . . .

Figure 5. 14 Add Close-Ended question – Lecture

5.3.7. Students' score Lecture Interface

In this page, lecturer can check the scores acquired by the students after they've done practices from the schedules created by the lecturer. To get the score of each student, lecturer needs to input class and schedule respectively.



Welcome, Dosen Coba !

Nilai Mahasiswa berhasil diambil

Rekap Nilai

* Pilih Kelas: Kelas Demo

* Pilih Jadwal: Demo Jadwal

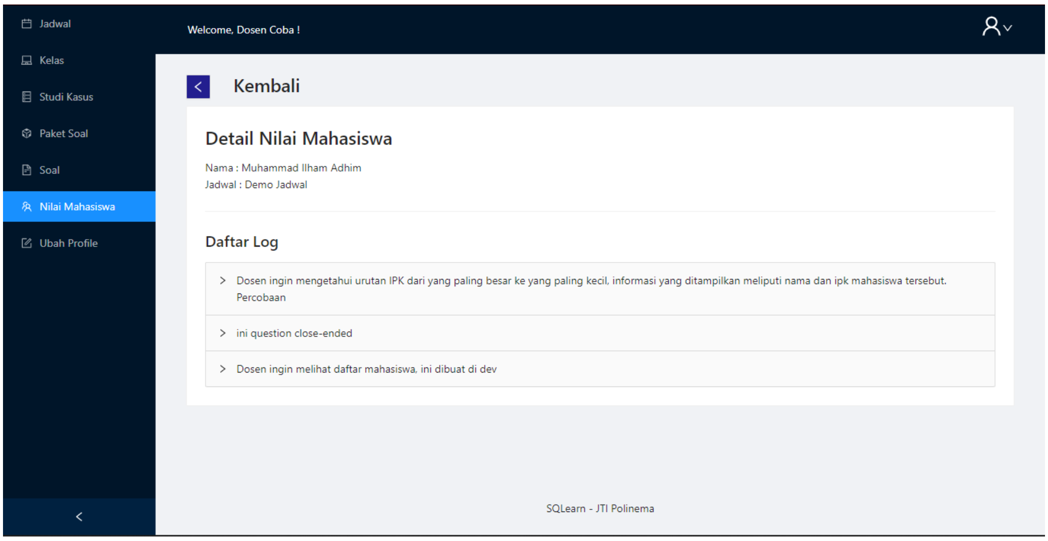
Nilai untuk kelas Kelas Demo

Muhammad Ilham Adhim NIM 1841720076 Score : 90

SQLearn - JTI Polinema

Figure 5. 15 Student's score Page Implementation

Furthermore, lecturer can check the detail of student’s score. This includes the recorded logs when the student did their practices. Such recorded logs are drag-and-drop movement and answer submission.



Welcome, Dosen Coba !

Detail Nilai Mahasiswa

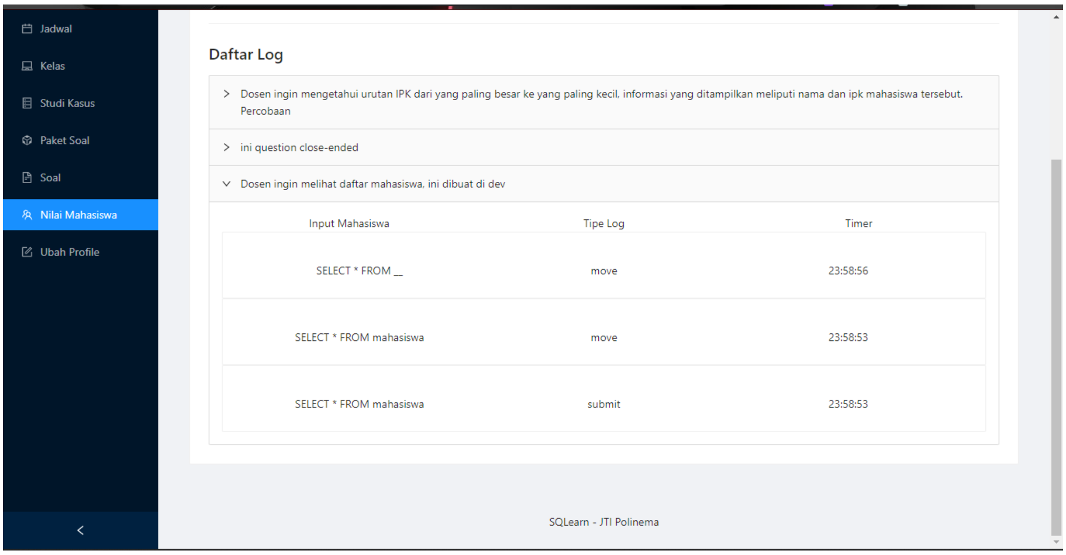
Nama : Muhammad Ilham Adhim
Jadwal : Demo Jadwal

Daftar Log

- Dosen ingin mengetahui urutan IPK dari yang paling besar ke yang paling kecil, informasi yang ditampilkan meliputi nama dan ipk mahasiswa tersebut. Percobaan
- ini question close-ended
- Dosen ingin melihat daftar mahasiswa, ini dibuat di dev

SQLearn - JTI Polinema

Figure 5. 16 Detail Score Each Student



Daftar Log

> Dosen ingin mengetahui urutan IPK dari yang paling besar ke yang paling kecil, informasi yang ditampilkan meliputi nama dan ipk mahasiswa tersebut. Percobaan

> ini question close-ended

Dosen ingin melihat daftar mahasiswa, ini dibuat di dev

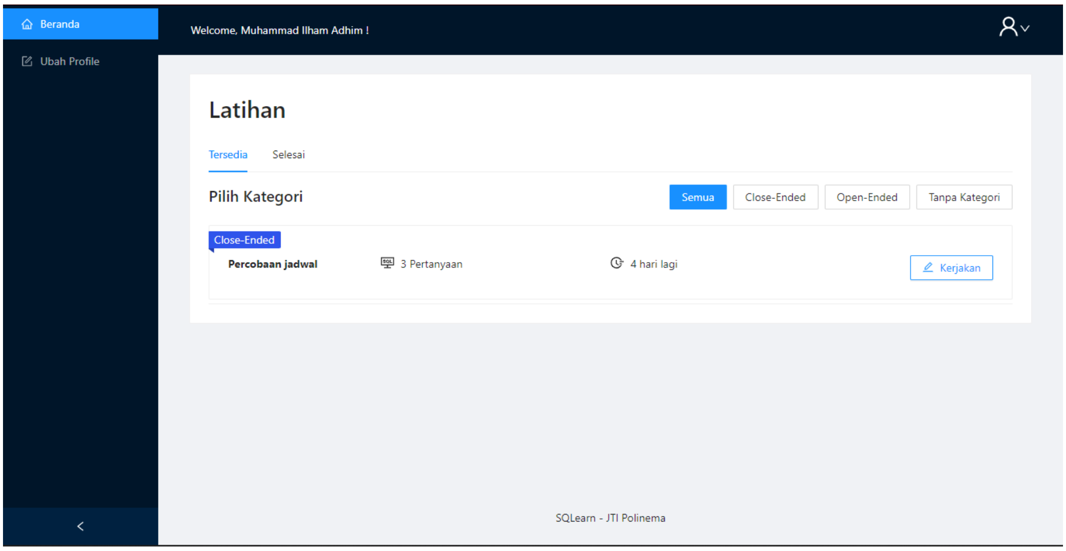
Input Mahasiswa	Tipe Log	Timer
SELECT * FROM _	move	23:58:56
SELECT * FROM mahasiswa	move	23:58:53
SELECT * FROM mahasiswa	submit	23:58:53

SQLearn - JTI Polinema

Figure 5. 17 Logs for each question each Student

5.3.8. Student Schedule Interface

The student’s role landing page will display available practices schedule given by their lecturer and needs to be done before due date. The focus on this research is close-ended approach, and students can filter the practices by that category as well.



Welcome, Muhammad Ilham Adhim !

Latihan

Pilih Kategori

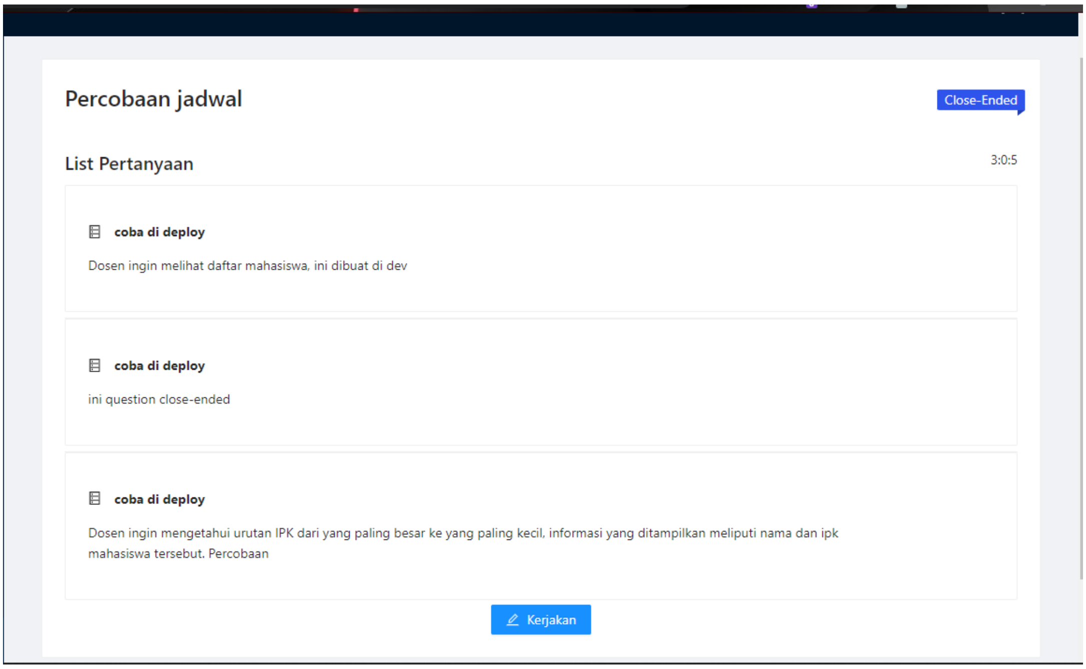
Close-Ended

Percobaan jadwal 3 Pertanyaan 4 hari lagi

SQLearn - JTI Polinema

Figure 5. 18 Student Page Implementation – Available Schedule

When selecting the practice, students will be shown information as follows. This includes the maximum duration to solve all the questions, preview of question list, and its respective database and case study.



Percobaan jadwal

List Pertanyaan

3:0:5

coba di deploy

Dosen ingin melihat daftar mahasiswa, ini dibuat di dev

coba di deploy

ini question close-ended

coba di deploy

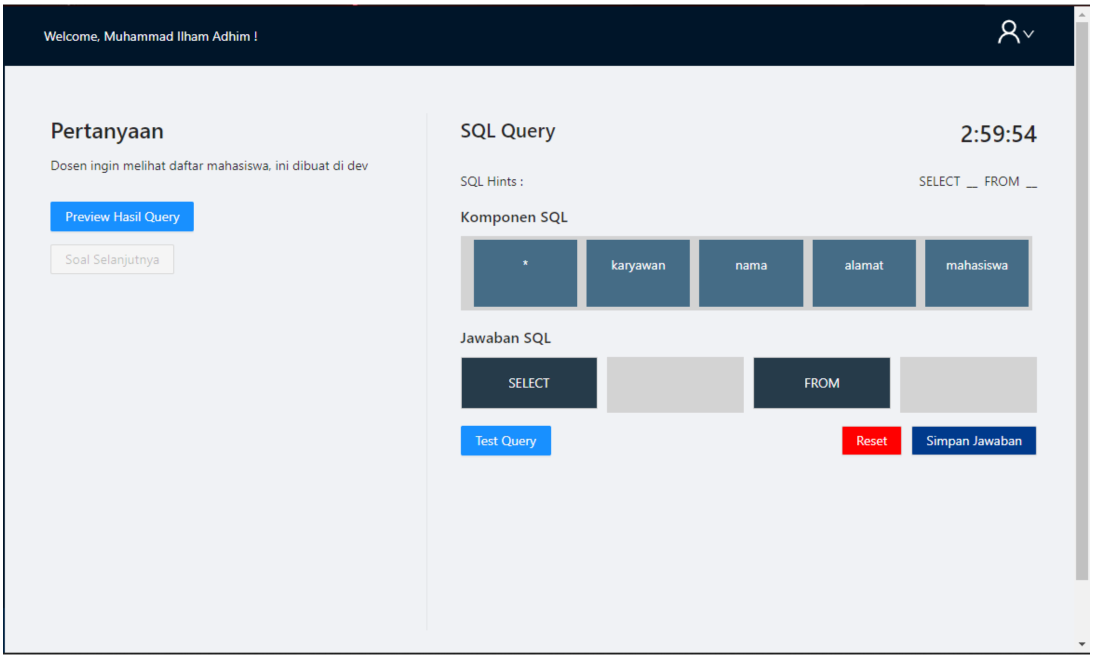
Dosen ingin mengetahui urutan IPK dari yang paling besar ke yang paling kecil, informasi yang ditampilkan meliputi nama dan ipk mahasiswa tersebut. Percobaan

Figure 5. 19 Student Practice Set Page Implementation

5.3.9. Student Practice Process Interface

This is the page for students to answer the questions. As described in lecturer part, ‘Komponen SQL’ is the SQL parts that can be dragged and dropped all the way to ‘Jawaban SQL’. As for the parts that are already exist in ‘Jawaban SQL’ are SQL Hints.

If in the process of solving the question, students are hesitant to submit the answer directly, they can test the query first and the system will notify them if the constructed SQL query is correct or not. Should it be incorrect, they can easily reset the constructed query and both ‘Komponen SQL’ and ‘Jawaban SQL’ will be restored to its original form. After they are sure enough to continue to the next question, they need to click the ‘Simpan Jawaban’ first and then proceed.



Welcome, Muhammad Ilham Adhim !

Pertanyaan

Dosen ingin melihat daftar mahasiswa, ini dibuat di dev

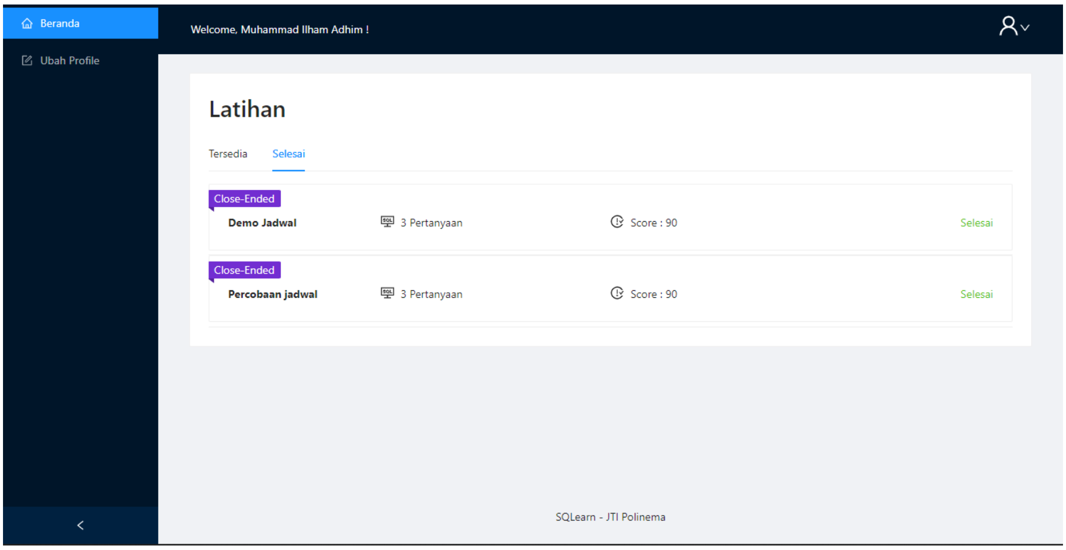
SQL Query 2:59:54

SQL Hints : SELECT _ FROM _

Figure 5. 20 Student Practice Page Implementation

5.3.10. Student Score Interface

After student finished their practices, they can see their result in landing page within ‘Selesai’ tab.



Welcome, Muhammad Ilham Adhim !

Latihan

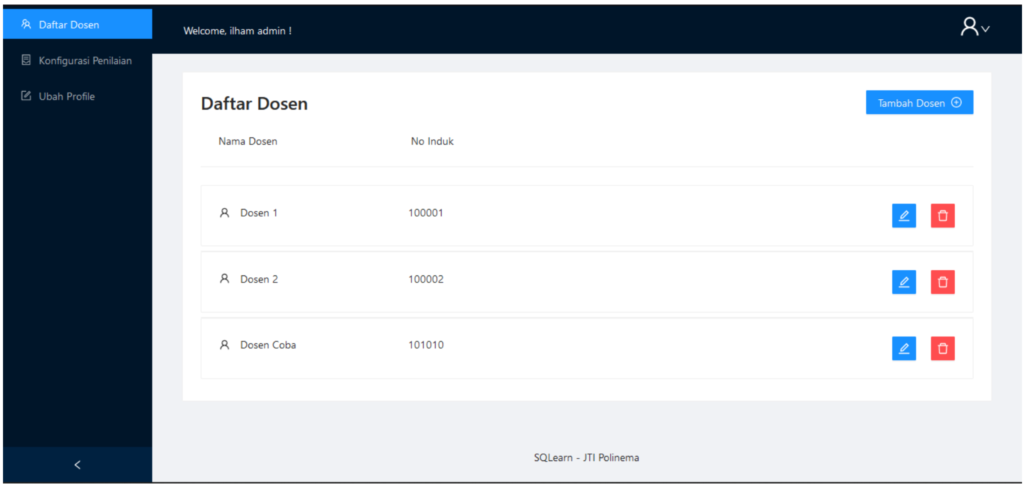
Close-Ended Demo Jadwal	3 Pertanyaan	Score : 90	Selesai
Close-Ended Percobaan jadwal	3 Pertanyaan	Score : 90	Selesai

SQLearn - JTI Polinema

Figure 5. 21 Student Schedule - Done

5.3.11. Admin Lecturer List Interface

This is the landing page of admin role. Admins can register new lecturer and manage all existing database lecturer in the system.



Welcome, ilham admin !

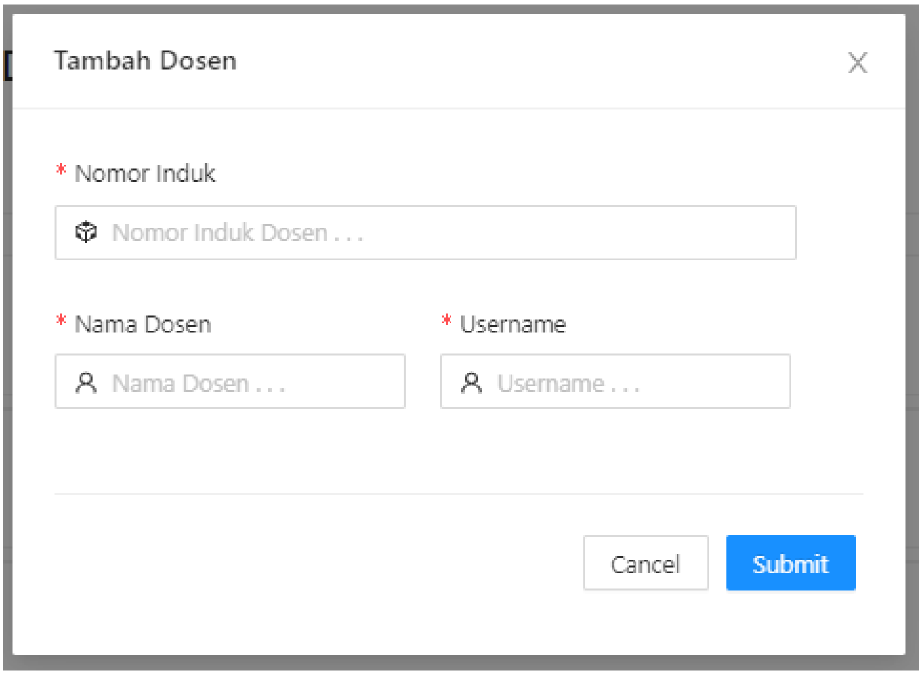
Daftar Dosen

Nama Dosen	No Induk
Dosen 1	100001
Dosen 2	100002
Dosen Coba	101010

SQLearn - JTI Polinema

Figure 5. 22 Admin Lecturer List Implementation

In order to add lecturer, admins should click button ‘Tambah Dosen’. Then, it is required to add NIP / Nomor Induk, Lecturer name, and its username. By default, the password is the same with the username.



Tambah Dosen

* Nomor Induk

Nomor Induk Dosen . . .

* Nama Dosen

Nama Dosen . . .

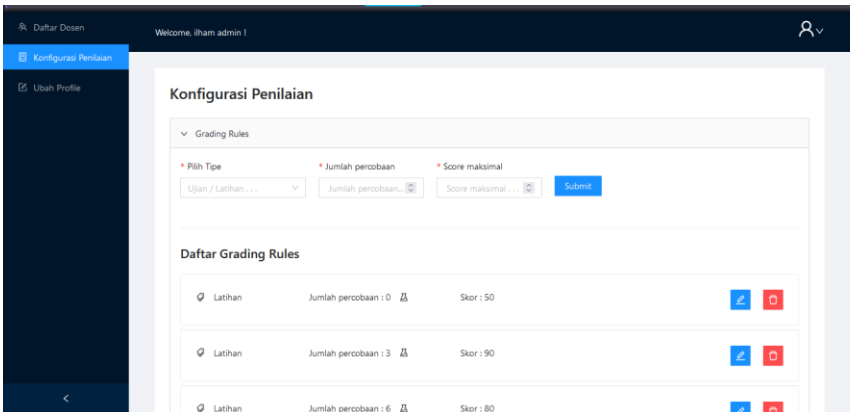
* Username

Username . . .

Figure 5. 23 Add Lecturer by Admin

5.3.12. Admin Grading Configuration Interface

This is the page where the grading configuration take place. Admins can set grading rules alongside with input from students score and the type whether it is practice or an exam.



Konfigurasi Penilaian

* Pilih Tipe: Ujian / Latihan . . .
* Jumlah percobaan: Jumlah percobaan...
* Score maksimal: Score maksimal . . .

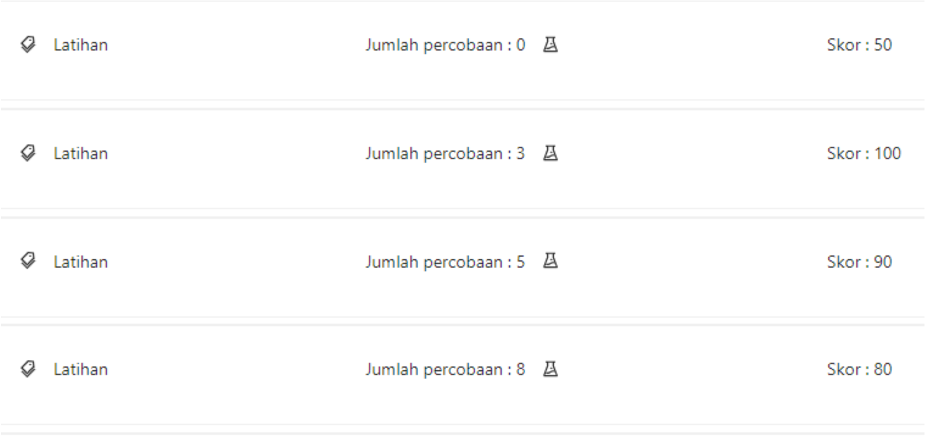
Daftar Grading Rules

Latihan	Jumlah percobaan : 0	Skor : 50
Latihan	Jumlah percobaan : 3	Skor : 90
Latihan	Jumlah percobaan : 6	Skor : 80

Figure 5. 24 Admin Grading Configuration Implementation

5.4. Grading Rules

Currently, in the system has 2 grading rules category which is for exercise and exam (Latihan dan Ujian). These grading rules can have multiple thresholds for system to check the amount of student trials on each question. The explanation of grading rules and threshold is as follows:



Latihan	Jumlah percobaan : 0	Skor : 50
Latihan	Jumlah percobaan : 3	Skor : 100
Latihan	Jumlah percobaan : 5	Skor : 90
Latihan	Jumlah percobaan : 8	Skor : 80

Figure 5. 25 Grading Rules for Assessment

For the first list, if the number of trials in a question is 0. This case happens when students doesn’t check query or submit answer at all for that particular question until the schedule time is finished. Therefore, for that question, student got 50.

If number of trials in ta question is lower than 3 and higher than 0, then students will get perfect score for that question. If number of trials in ta question is lower than 5 and higher than 3, then students will get 90 for that question. This logic applies the same for all available breakpoints.

For simulation, there is a case where students are required to answer 10 questions. First question, the student checks the answer for 4 times, and finish all remaining 9 question with 2 number of trials. Then, that means for this 1st question they will get 90, and get 100 for the rest of the questions. The points accumulated, which is 990. Then divide it with the number of questions. Therefore, the final score of the student is 99.

5.5. Testing Result

This part covers the result of student's exercise in SQLearn platform, collected pre-test and post-test scores, as well as the result of processed data.

5.5.1. Participant

Participants of this research were 28 first-year (2nd semester) students in Information Technology Major, Politeknik Negeri Malang.

5.5.2. Design Experiment

Design experiment is used for planning how the experiment should work in the process of data gathering. Such design experiment is illustrated as follows.

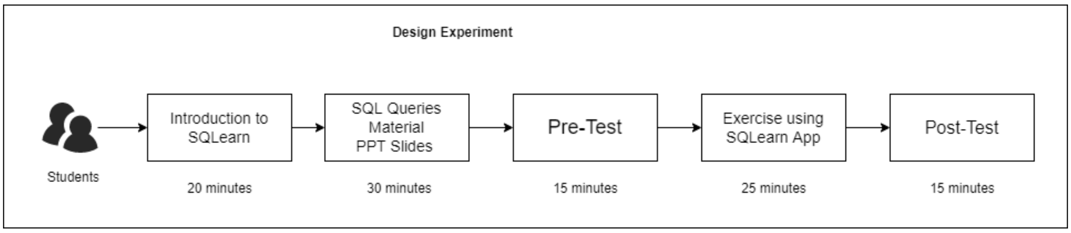


Figure 5. 26 Design Experiment

In field, the experiment is being held via *Zoom Meeting*. Whereas the pre-test and post-test questions are provided in *Google Forms* to make easier access and use for the students via daring.

As illustrated in Figure 5.25, there are 5 main activities in order to collect the data, which are Introduction to SQLearn, SQL Queries Material PPT Slides, Pre-Test, using SQLearn Application, and Post-Test. Introduction to SQLearn allows students to ensure they can log in to the system while researcher's demonstrating the usage of SQL drag-and-drop for 2 SQL Query exercises. Then, SQL Query material was conducted for 20 minutes this covers topics ranging from using basic SQL Query SELECT, ORDER BY, GROUP BY, JOINS, and SQL conditions (WHERE, LIKE, BETWEEN). Next up, participants are directed to do pre-test that consists of 10 SQL Query questions for 15 minutes in *Google Form.* After that, participants are asked to check their SQLearn application and do another 10 in-app questions for 25 minutes, this time using drag-and-drop while given SQL parts and hints to solve the problem. Lastly, the post-test consists of 10 SQL Query questions for 15 minutes. The questions being used in post-test are similar with those in pre-test. This way, it can be used to determine the effect of SQLearn application to the score.

Both pre-test and post-test session were guided by the researcher. *Google Form* were given only multiple-choice options, question detail is being shown 1,5 minutes per question in slides. The detail of pre-test, post-test, and SQLearn questions are provided in attachments.

5.5.3. Data Processing

Pre-test and post-test scores were collected in this research and will be processed to answer the and research problem stated previously. Data processing starts from normality test by using *Kolmogorov-Smirnov* test. Then it will proceed to Wilcoxon Signed Ranks Test to determine the score's difference of pre-test and post-test. The method also gives insight of its significances. Data collected are as follows:

No	Pre-Test	Post-Test
1	70	80
2	60	80
3	70	80
4	70	80
5	70	80
6	70	80
7	70	80
8	60	70
9	70	80
10	50	70
11	60	80
12	70	80
13	60	80
14	70	80
15	50	60
16	30	50
17	60	60
18	70	70
19	60	70
20	70	80
21	60	60
22	60	80
23	60	60
24	30	60
25	60	80
26	50	80
27	40	60
28	80	80

Table 5. 1 Pre-Test and Post-Test Result

After the data has been collected from mentioned flow above, it can be illustrated with box-plot to determine if there is any outlier in our data to be removed. From the first box-plot, there are 3 data that can be considered outlier and needs to be removed.

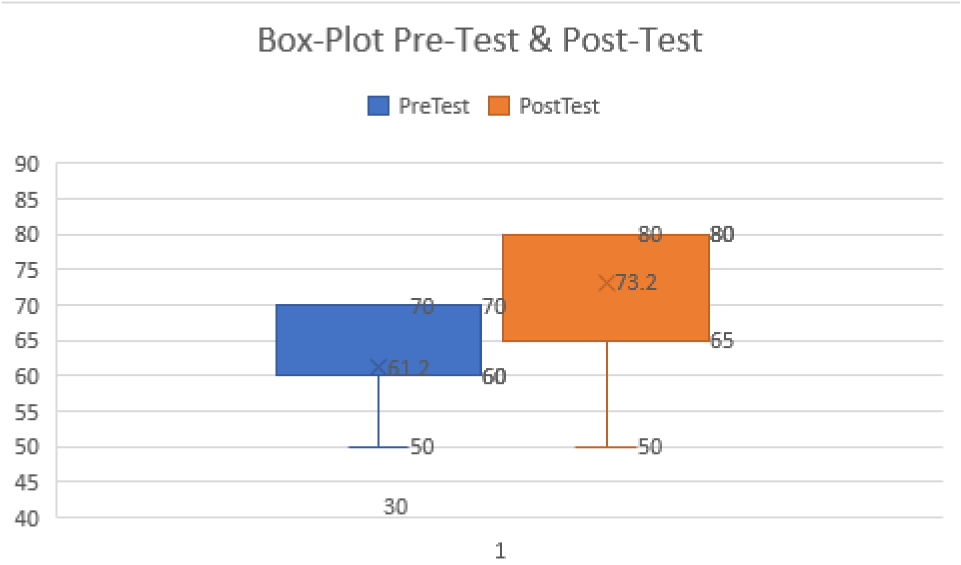


Figure 5. 27 Raw Data Pre-test and Post-Test in Box-Plot

After removing the outliers, the corrected box-plot becomes as follows:

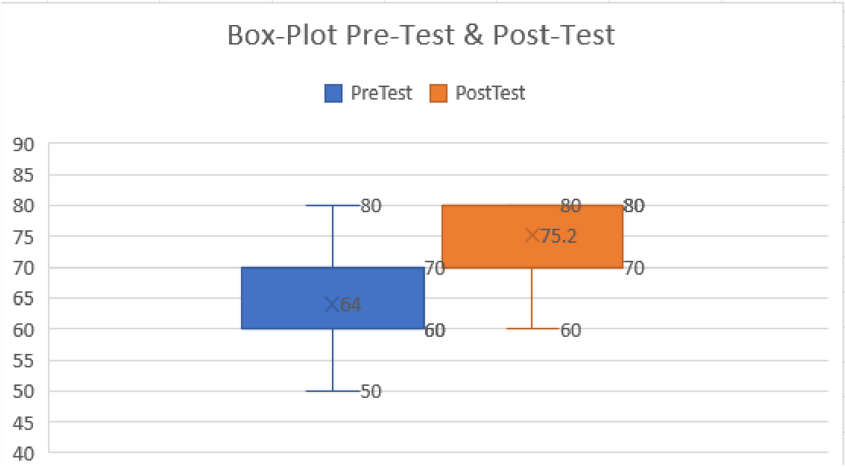


Figure 5. 28 Box Plot for Pre-Test & Post-Test

Furthermore, the data can be displayed as line chart to observe the overall picture of pre-test and post-test score that has been conducted to participants.

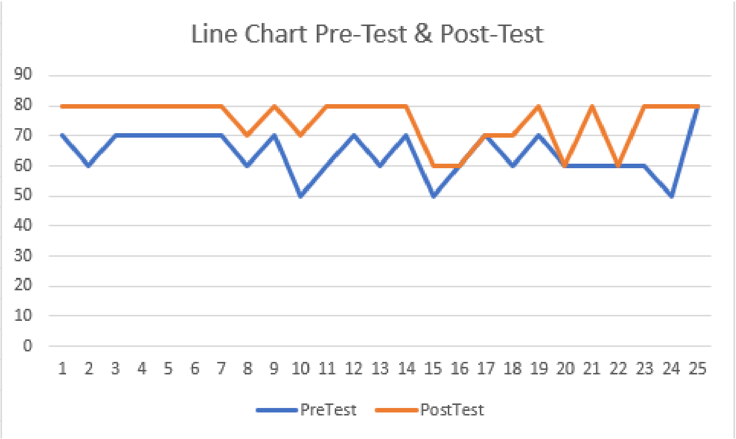


Figure 5. 29 Line Chart for Pre-Test & Post-Test

After collecting such data, it will then be processed according below illustration

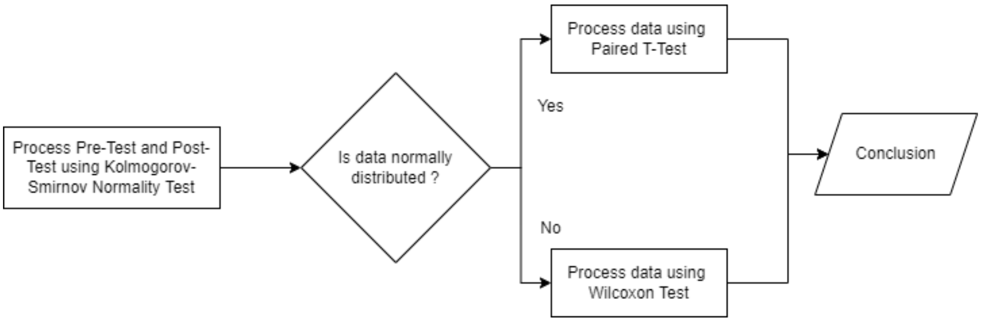

Process Pre-Test and Post-Test using Kolmogorov-Smirnov Normality Test
Is data normally distributed ?
Yes
No
Process data using Paired T-Test
Process data using Wilcoxon Test
Conclusion

Figure 5. 30 Data Processing Flowchart

a. *Kolmogorov-Smirnov* Normality Test

One-Sample Kolmogorov-Smirnov Test			
		Pre-Test	Post-Test
N		25	25
Normal Parameters	Mean	64.0000	75.2000
	Std. Deviation	7.6376	7.7028
Most Extreme Differences	Absolute	0.264	0.413
	Positive	0.220	0.267
	Negative	-0.264	-0.413
Test Statistic		0.264	0.413
Asymp. Sig. (2-tailed)		0.000	0.000

Table 5. 2 Kolmogorov-Smirnov Test

Based on Table 5.2, it is defined that total data for both pre-test and post-test are 25 participants. Mean of pre-test is 64 and post-test is 75.2000. Std. Deviation for pre-test is 7.6376 and post-test is 7.7028. The *Asymp. Sig (2-tailed)* score for both pre-test and post-test are 0 (< 0.05). Based on decision-making basis, both data pre-test and post-test are not distributed normally.

b. Wilcoxon Signed Ranks Test

From Normality Test, we can conclude that the data collected is not normally distributed. Therefore, next data processing is to determine if there is any difference between students' average pre-test and post-test score by using Non-Parametric Wilcoxon Signed Ranks Test as follows:

Ranks				
		N	Mean Rank	Sum of Ranks
Pre-Test & Post-Test	Negative Ranks	0	0.00	0.00
	Positive Ranks	20	10.50	210.00
	Ties	5		
	Total	25		

Table 5. 3 Wilcoxon Signed Ranks Test

Based on Table 5.3, it's known that there is no data that considered as *negative ranks* which means that no student has their post-test score lower than pre-test. As for the positive ranks, there are 20 students has their post-test score greater than pre-test score, the average increasing score is 12.50 points and its sum is 216.00. In addition, there is 5 students that has same score in their pre-test and post-test.

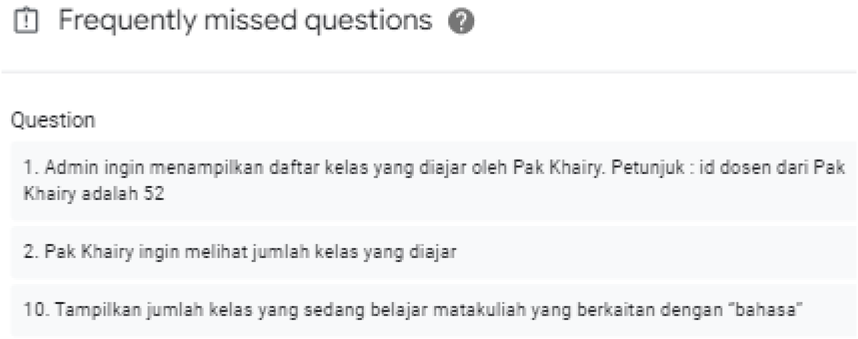
Test Statistics	
	PostTest - PreTest
Z	-4.064
Asymp. Sig. (2-tailed)	0.000

Table 5. 4 Wilcoxon Signed Ranks Test Statistics

Based on Table 5.4, it is determined that Z score is -4.064 and *Asymp. Sig. (2-tailed)* is 0.000 (< 0.05). These values will be used for specifying whether the data collected has significant difference between tested samples.

5.5.4. Most Incorrectly Answered Questions

From the collected data, it can also be observed in detail of which topics and type of questions that can be concluded as insignificant through this research. From Google Form, Researcher observed that there are 3 questions that most of students answered incorrectly in pre-Test. Those 3 questions are as follows:

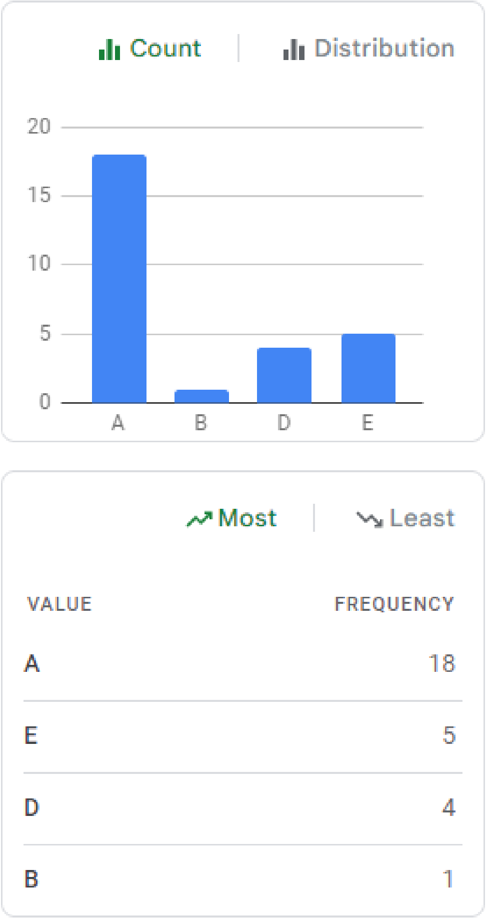


Frequently missed questions

Question
1. Admin ingin menampilkan daftar kelas yang diajar oleh Pak Khairy. Petunjuk : id dosen dari Pak Khairy adalah 52
2. Pak Khairy ingin melihat jumlah kelas yang diajar
10. Tampilkan jumlah kelas yang sedang belajar matakuliah yang berkaitan dengan "bahasa"

Figure 5. 31 Frequently missed Question in pre-test

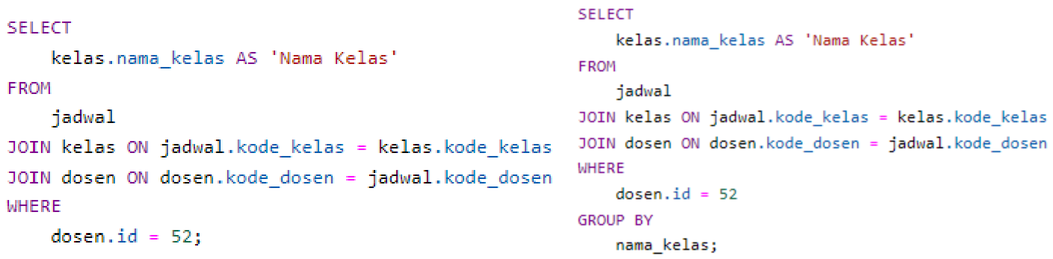
- 1. Admin ingin menampilkan daftar kelas yang diajar oleh Pak Khairy. Petunjuk : id dosen dari Pak Khairy adalah 52



VALUE	FREQUENCY
A	18
E	5
D	4
B	1

Figure 5. 32 Answer count of 1st question in Pre-Test

The answer for this question is B. Which means, only 1 person that answered this question correctly. As for the majority (18 participants) of the participants answered A.



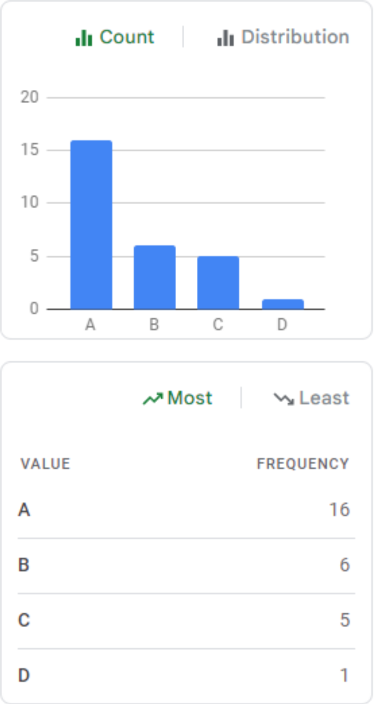
```
SELECT
    kelas.nama_kelas AS 'Nama Kelas'
FROM
    jadwal
JOIN kelas ON jadwal.kode_kelas = kelas.kode_kelas
JOIN dosen ON dosen.kode_dosen = jadwal.kode_dosen
WHERE
    dosen.id = 52;
```

```
SELECT
    kelas.nama_kelas AS 'Nama Kelas'
FROM
    jadwal
JOIN kelas ON jadwal.kode_kelas = kelas.kode_kelas
JOIN dosen ON dosen.kode_dosen = jadwal.kode_dosen
WHERE
    dosen.id = 52
GROUP BY
    nama_kelas;
```

Figure 5. 33 A and B option in 1st question in Pre-Test

From the question, the difference is on whether it uses **group by** keyword or not. Based on figure 5.31, it can be concluded that students have difficulties on determining the use case of group by and when to use it.

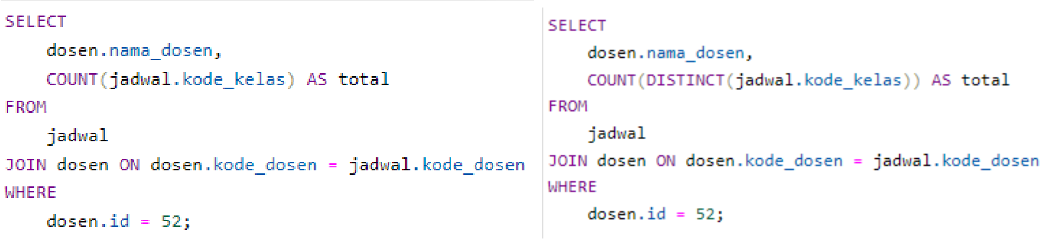
- 2. Pak Khairy ingin melihat jumlah kelas yang diajar



VALUE	FREQUENCY
A	16
B	6
C	5
D	1

Figure 5. 34 Answer count of 2nd question in Pre-Test

The answer for this question is C. Which means, 5 people answered this question correctly. As for the majority (16 participants) of the participants answered A. 6 people answered B and 1 person answered D.



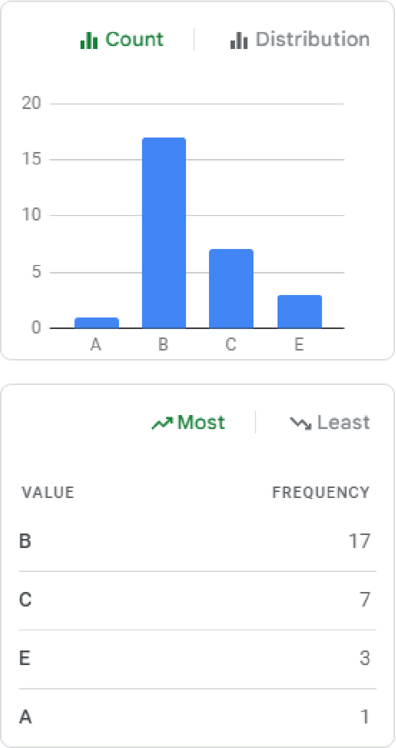
```
SELECT
    dosen.nama_dosen,
    COUNT(jadwal.kode_kelas) AS total
FROM
    jadwal
JOIN dosen ON dosen.kode_dosen = jadwal.kode_dosen
WHERE
    dosen.id = 52;
```

```
SELECT
    dosen.nama_dosen,
    COUNT(DISTINCT(jadwal.kode_kelas)) AS total
FROM
    jadwal
JOIN dosen ON dosen.kode_dosen = jadwal.kode_dosen
WHERE
    dosen.id = 52;
```

Figure 5. 35 A and C option in 2nd question in Pre-Test

From the question, the difference is on whether it uses **distinct keyword with simple condition**. As illustrated on figure 5.33, it can be concluded that students have difficulties on determining the use case of distinct and when to use it.

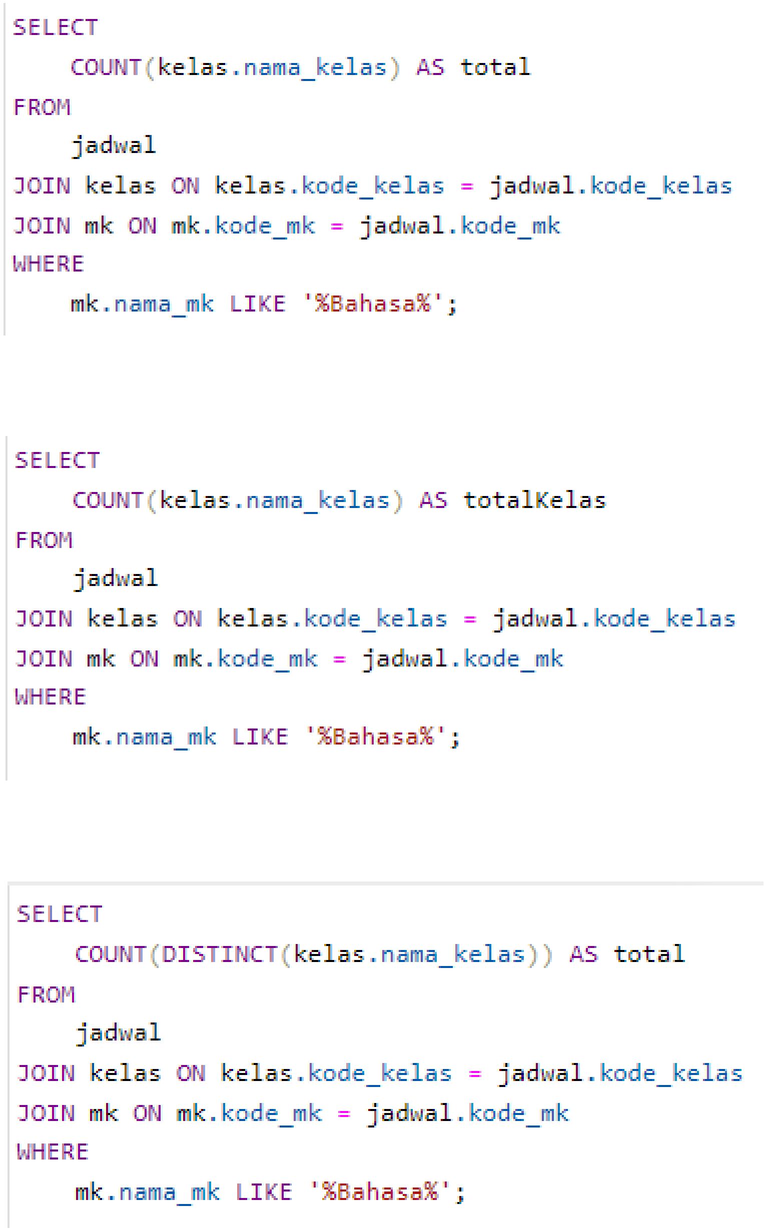
- 10. Tampilkan jumlah kelas yang sedang belajar matakuliah yang berkaitan dengan "bahasa"



VALUE	FREQUENCY
B	17
C	7
E	3
A	1

Figure 5. 36 Answer count of 10th question in Pre-Test

The answer for this question is E. Which means, 3 people answered this question correctly. As for the majority (17 participants) of the participants answered B. 7 people answered C and 1 person answered A.



```
SELECT
    COUNT(kelas.nama_kelas) AS total
FROM
    jadwal
JOIN kelas ON kelas.kode_kelas = jadwal.kode_kelas
JOIN mk ON mk.kode_mk = jadwal.kode_mk
WHERE
    mk.nama_mk LIKE '%Bahasa%';
```

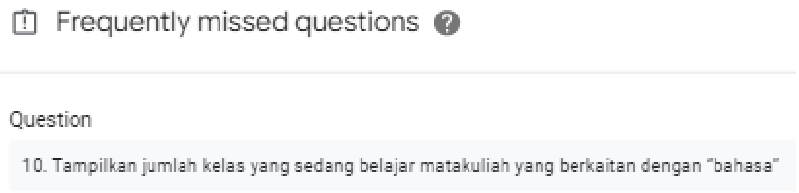
```
SELECT
    COUNT(kelas.nama_kelas) AS totalKelas
FROM
    jadwal
JOIN kelas ON kelas.kode_kelas = jadwal.kode_kelas
JOIN mk ON mk.kode_mk = jadwal.kode_mk
WHERE
    mk.nama_mk LIKE '%Bahasa%';
```

```
SELECT
    COUNT(DISTINCT(kelas.nama_kelas)) AS total
FROM
    jadwal
JOIN kelas ON kelas.kode_kelas = jadwal.kode_kelas
JOIN mk ON mk.kode_mk = jadwal.kode_mk
WHERE
    mk.nama_mk LIKE '%Bahasa%';
```

Figure 5. 37 B, C, and E option in 10th question in Pre-Test

From the question, the difference is on whether it uses **distinct** keyword and the **expected column name aliases**. As illustrated on figure 5.35, it can be concluded that students have difficulties on determining the use case of distinct and when to use it.

Based on Google Form result, the frequently missed question in post-test is as follows



Frequently missed questions

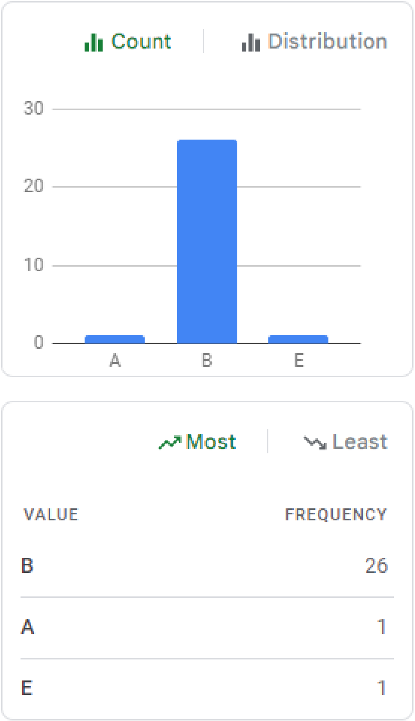
Question

10. Tampilkan jumlah kelas yang sedang belajar matakuliah yang berkaitan dengan “bahasa”

Figure 5. 38 Frequently missed Question in post-test

However, to further determine the significances of SQLearn, researcher will proceed to compare mentioned questions in pre-test and see how participants perform in post-test on the same questions.

- 1. Admin ingin menampilkan daftar kelas yang diajar oleh Pak Khairy. Petunjuk : id dosen dari Pak Khairy adalah 52

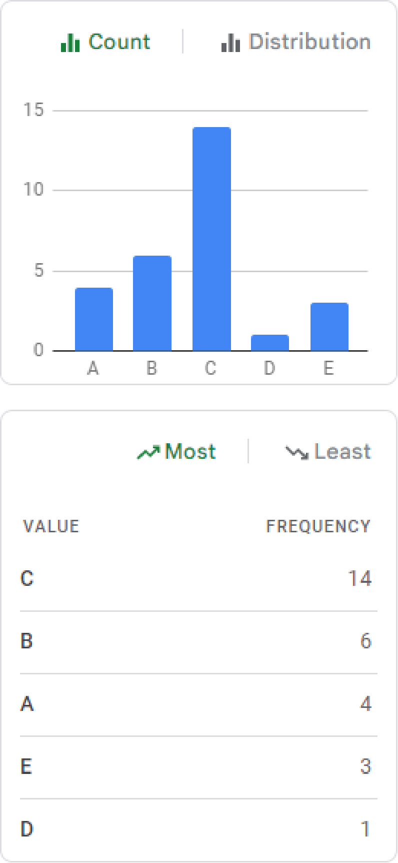


VALUE	FREQUENCY
B	26
A	1
E	1

Figure 5. 39 Answer count of 1st question in Post-Test

The answer for this question is B. Which means, 26 participants have answered this question correctly and the usage of SQLearn has positive impact on **group by** topic.

- 2. Pak Khairy ingin melihat jumlah kelas yang diajar

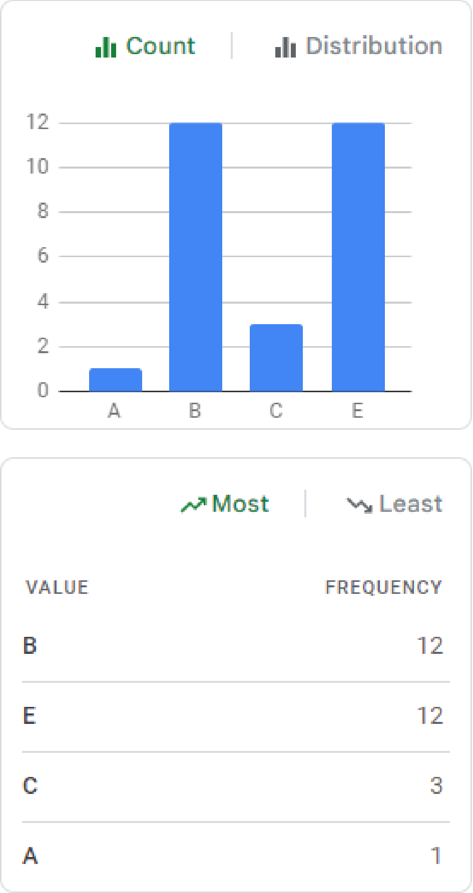


VALUE	FREQUENCY
C	14
B	6
A	4
E	3
D	1

Figure 5. 40 Answer count of 2nd question in Post-Test

The answer for this question is C. Which means, 14 people answered this question correctly. Compared to previously in pre-test, it was only 5 people. So, the usage of SQLearn has positively impacted 9 more people in answering **distinct keyword with simple condition**

- 10. Tampilkan jumlah kelas yang sedang belajar matakuliah yang berkaitan dengan “bahasa”



VALUE	FREQUENCY
B	12
E	12
C	3
A	1

Figure 5. 41 Answer count of 10th question in Post-Test

The answer for this question is E. Which means, 12 people answered this question correctly. This signifies 9 more people has been positively impacted after using SQLearn in **distinct** keyword and the **expected column name aliases**. However, in overall it is still most frequently missed question in post-test. This is due to total of 16 people answering incorrect option. Some possibilities of this may occur due to student’s burnout factor after solving lots of questions in pre-test, SQLearn questions, and post-test itself.

CHAPTER VI. RESULT AND DISCUSSION

6.1. Result

There are 28 students that participated in this research. After observing the box-plot illustration, it is signified that there are 3 data are considered as outlier. Therefore, the data will be processed is 25 data. Based on acquired data, there are 20 participants that has their scores increased in post-test, 5 participants that has the same score from their pre-test and post-test, and 1 participant that has their score decreased in post-test.

Based on testing result in *Kolmogorov-Smirnov* Test Table 5.2, the average of participants' pre-test score is 64.000 and average of post-test score is 75.200. This signifies that there is 9.2 points difference in pre-test and post-test average score. Next up, it also come into consideration that whether this difference has any significance or not. Since the data is not normally distributed, the data will be processed further with Wilcoxon Signed Ranks Test.

Based on testing result of Wilcoxon Signed Ranks Test in Table 5.4, it is known that the value of *Asymp. Sig (2-tailed)* is below 0.05, therefore it can be concluded that the such average improvement is proved being significant statistically. Hence, the usage of SQL Query Construction with Drag-and-drop alongside with close-ended approach towards students understanding has positive impact significantly.

6.2. Discussion

Based on test result that has been conducted, discussion are as follows:

1. Drag-and-drop method and SQL Query construction in SQLearn can be used as SQL query exercise in database subject.

2. Drag-and-drop method and SQL Query Construction with close-ended approach in SQLearn has significant positive impact towards students understanding in SQL Query topic in database subject

3. *Kolmogorov-Smirnov* test can be used for testing the distribution normality of pre-test and post-test data

4. Wilcoxon Signed Ranks test method can be used to identify the significances between 2 samples.

CHAPTER VII. CONCLUSION

7.1 Conclusion

Based on this research result of SQL Query Construction practice with a close-ended approach and drag-and-drop implementation, it is concluded that this method has a significant positive impact with increasing 11.2 points on students' post-test scores in SQL Queries topic in database subject.

7.2 Suggestion

Based on this research result and built application with the idea of providing SQL Parts and Hints by lecturer as close-ended approach and *drag-and-drop* implementation. There are some room for improvement for further research which are listed as follows:

1. Logs recorded in the database can be optimized and being used for mapping students understanding in more detail.
2. Broaden *drag-and-drop* implementation so that it is not limited to SELECT statements.

REFERENCES

Ali, A. M., Tumian, A., & Abu Seman, M. S. (2017). A conceptual approach for understanding computer programming skills development. *2017 International Conference on Research and Innovation in Information Systems (ICRIIS),* 1–5. https://doi.org/10.1109/ICRIIS.2017.8002526

Baburajan, V., e Silva, J. de A., & Pereira, F. C. (2021). Open-Ended Versus Closed-Ended Responses: A Comparison Study Using Topic Modeling and Factor Analysis. *IEEE Transactions on Intelligent Transportation Systems, 22*(4), 2123–2132. https://doi.org/10.1109/TITS.2020.3040904

Borthick, A. F., Bowen, P. L., Jones, D. R., & Hung Kam Tse, M. (2001). The effects of information request ambiguity and construct incongruence on query development. *Decision Support Systems*, *32*(1), 3–25. https://doi.org/10.1016/S0167-9236(01)00097-5

Fedorenko, E., Ivanova, A., Dhamala, R., & Bers, M. U. (2019). The Language of Programming: A Cognitive Perspective. *Trends in Cognitive Sciences*, *23*(7), 525–528. https://doi.org/10.1016/j.tics.2019.04.010

Bosse, Y., & Gerosa, M. A. (2017). Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage. *ACM SIGSOFT Software Engineering Notes*, *41*(6), 1–6. https://doi.org/10.1145/3011286.3011301

Broatch, J. E., Dietrich, S., & Goelman, D. (2019). Introducing Data Science Techniques by Connecting Database Concepts and dplyr. *Journal of Statistics Education*, *27*(3), 147–153. https://doi.org/10.1080/10691898.2019.1647768

Faeskorn-Woyke, H., Bertelsmeier, B., & Strohschein, J. (2020). *A Decision Tree Approach for the Classification of Mistakes of Students Learning SQL*. 7.

Huang, X., & Leng, J. (2019). Design of Database Teaching Model Based on Computational Thinking Training. *International Journal of Emerging Technologies in Learning (IJET)*, *14*(08), 52.

https://doi.org/10.3991/ijet.v14i08.10495

Jiandong, H., Jinyu, S., & Suojuan, Z. (2018). Study on the Iterative Teaching Method in Database Curriculum. *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*, 615–617. https://doi.org/10.1109/ITME.2018.00141

Leinonen, J., Pirttinen, N., & Hellas, A. (2020). Crowdsourcing Content Creation for SQL Practice. *Proceedings of the 2020 ACM Conference on Innovation and Technology in Computer Science Education*, 349–355. https://doi.org/10.1145/3341525.3387385

Luthfi, M., & Ayu, A. (2019). *A review of increasing teaching and learning database subjects in computer science*. 9.

Mason, R., Seton, C., & Cooper, G. (2016). Applying cognitive load theory to the redesign of a conventional database systems course. *Computer Science Education*, *26*(1), 68–87. https://doi.org/10.1080/08993408.2016.1160597

Phewkum, C., Kaewchaiya, J., Kobayashi, K., & Atchariyachanvanich, K. (2019). ScrambleSQL: A Novel Drag-and-drop SQL Learning Tool. *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, 340–344. https://doi.org/10.1109/ICSEC47112.2019.8974815

Shang, P. (2017). *Application and Exploration of Blended Learning Mode in the Teaching of Database Course Design:* 2016 7th International Conference on Education, Management, Computer and Medicine (EMCM 2016), Shenyang, China. https://doi.org/10.2991/emcm-16.2017.160

Shin, S.-S. (2020). Structured Query Language Learning: Concept Map-Based Instruction Based on Cognitive Load Theory. *IEEE Access, 8*, 100095–100110. https://doi.org/10.1109/ACCESS.2020.2997934

Sweller, J. (2020). Cognitive load theory and educational technology. *Educational Technology Research and Development*, *68*(1), 1–16. https://doi.org/10.1007/s11423-019-09701-3

Villamor, M. M. (2020). A review on process-oriented approaches for analyzing novice solutions to programming problems. *Research and Practice in Technology Enhanced Learning*, *15*(1), 8. https://doi.org/10.1186/s41039-020-00130-y

Weintrop, D., & Wilensky, U. (2017). Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. *ACM Transactions on Computing Education*, *18*(1), 1–25. https://doi.org/10.1145/3089799

Hoque, A. S. Md. L., Bashiry, G. Md. M., & Uddin, Md. R. (2014). Equivalence of Problems in Problem Based e-Learning of Database. *2014 IEEE Sixth International Conference on Technology for Education*, 106–109. https://doi.org/10.1109/T4E.2014.23

Hyman, D. M. R. (2016). *Open- versus Close-Ended Survey Questions*. 6.

Jamil, M. G., & Isiaq, S. O. (2019). Teaching technology with technology: Approaches to bridging learning and teaching gaps in simulation-based programming education. *International Journal of Educational Technology in Higher Education*, *16*(1), 25. https://doi.org/10.1186/s41239- 019-0159-9

Lin, W.-L., & Lien, Y.-W. (2013). The Different Role of Working Memory in Open-Ended Versus Closed-Ended Creative Problem Solving: A Dual-Process Theory Account. *Creativity Research Journal*, *25*(1), 85–96. https://doi.org/10.1080/10400419.2013.752249

Phewkum, C., Kaewchaiya, J., Kobayashi, K., & Atchariyachanvanich, K. (2019). ScrambleSQL: A Novel Drag-and-drop SQL Learning Tool. *2019 23rd International Computer Science and Engineering Conference (ICSEC)*, 340–344. https://doi.org/10.1109/ICSEC47112.2019.8974815

Puspitasari, D., Arhandi, P. P., Saputra, P. Y., Syaifudin, Y. W., Himawan, H. A., & Sholihah, P. A. (2019). Online judge MySQL for learning process of database practice course. *IOP Conference Series: Materials Science and Engineering*, *523*(1), 012046. https://doi.org/10.1088/1757- 899X/523/1/012046

Sastry, K. S. (2015). *An Effective Approach for Teaching Database Course*. 11.

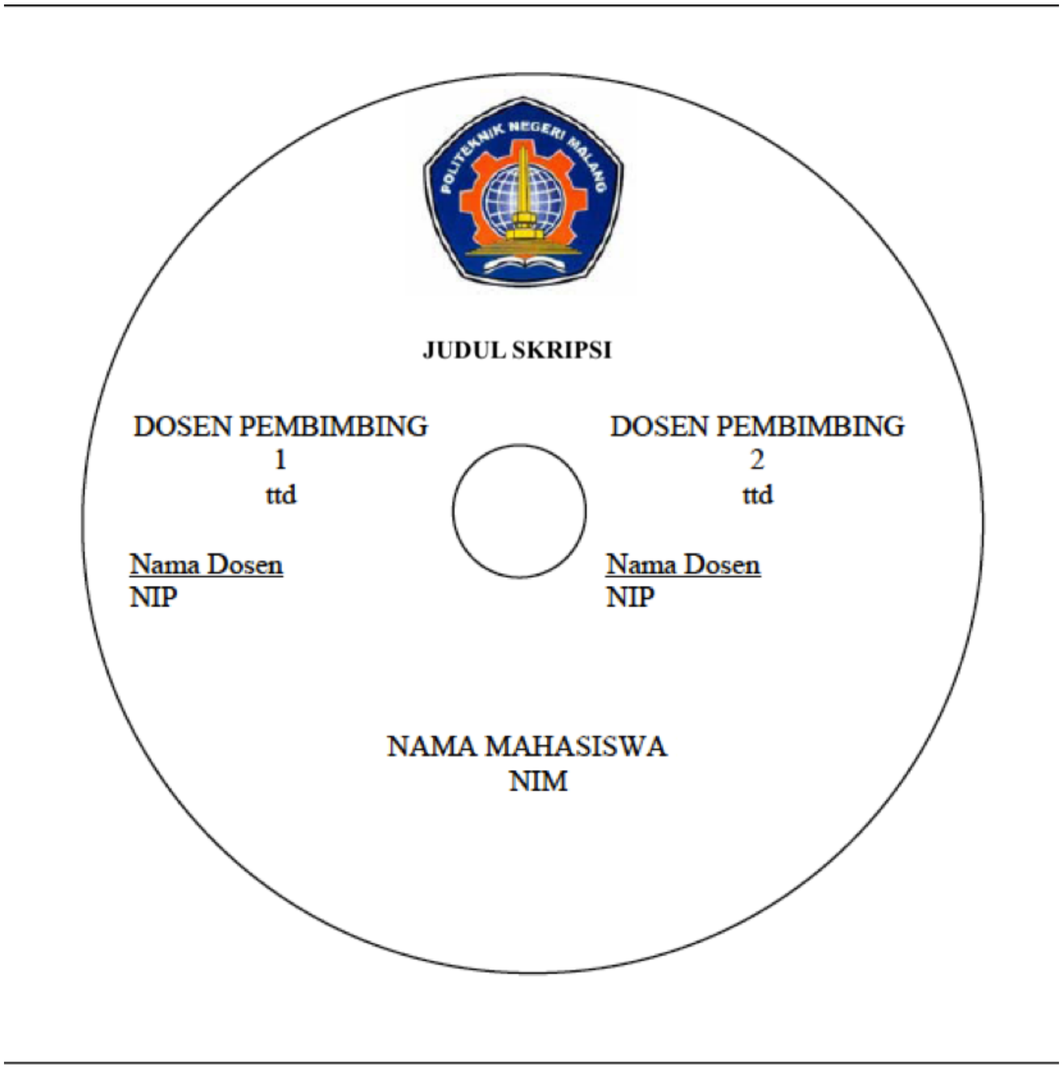
Taipalus, T. (2019). *A Notation for Planning SQL Queries*. *30*, 9.

Surbakti, K. (2018). Kajian Mengenai Pentingnya Basis Data Bagi Sekolah Saat Ini. Jurnal 74 Curere, 02(02), 2597–9515.

- Contoh Penomoran dan Penyajian Table

Table 3.1 Table Data Mahasiswa

Nama Atribut	**Data Type**	**Description**
ID_MAHASISWA	VARCHAR (11)	NOT NULL, PRIMARY_KEY
NAMA_MAHASISWA	VARCHAR (50)	NOT NULL
TEMPAT LAHIR	VARCHAR (15)	NOT NULL
TANGGAL_LAHIR	DATE	NOT NULL
ALAMAT	VARCHAR (100)	NOT NULL
NO_TELP	VARCHAR (11)	NOT NULL



JUDUL SKRIPSI

DOSEN PEMBIMBING
1
ttd

<u>Nama Dosen</u>
NIP

DOSEN PEMBIMBING
2
ttd

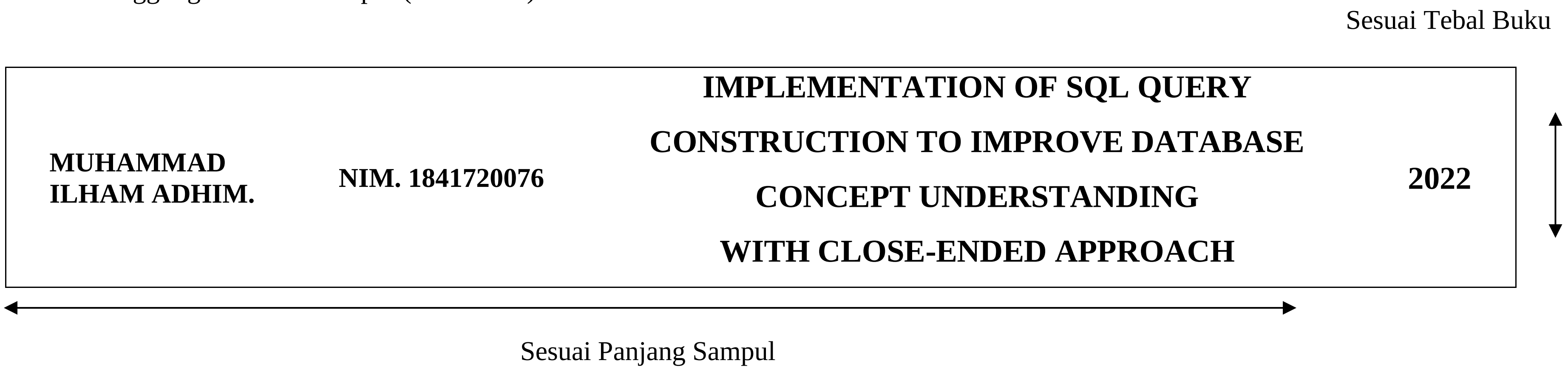
<u>Nama Dosen</u>
NIP

NAMA MAHASISWA
NIM

Lampiran 17. Format Cover CD

Lampiran tidak memiliki nomor halaman

Format Punggung Halaman Sampul (hard cover)



Sesuai Tebal Buku

MUHAMMAD ILHAM ADHIM.

NIM. 1841720076

IMPLEMENTATION OF SQL QUERY CONSTRUCTION TO IMPROVE DATABASE CONCEPT UNDERSTANDING WITH CLOSE-ENDED APPROACH

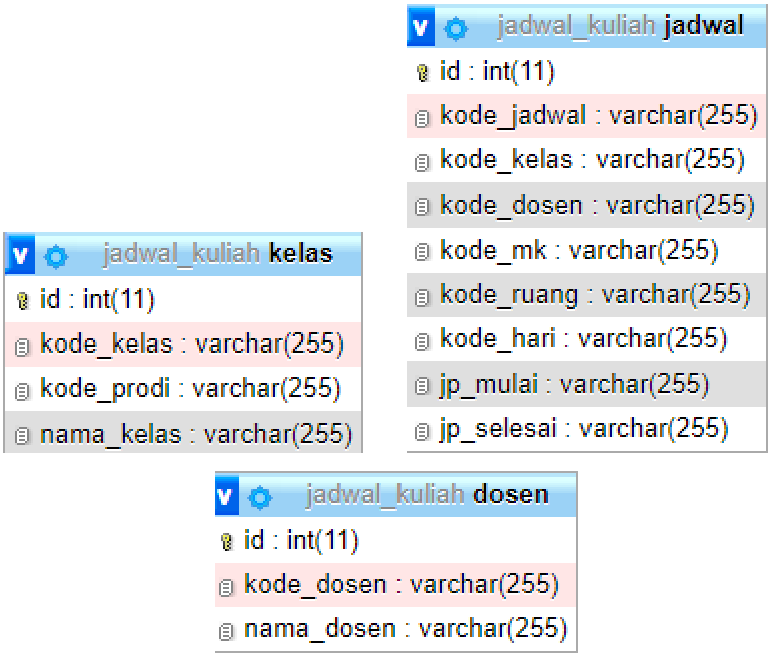
2022

Sesuai Panjang Sampul

Lampiran 18. Contoh Punggung Halaman Sampul (*Hard Cover*)

Attachment 1 – PreTest PostTest Questions

1. Tampilkan daftar kelas yang diajar oleh pak Khairy. ID dosen untuk Pak Khairy adalah 52

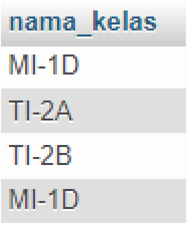


jadwal_kuliah jadwal
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah kelas
id : int(11)
kode_kelas : varchar(255)
kode_prodi : varchar(255)
nama_kelas : varchar(255)

jadwal_kuliah dosen
id : int(11)
kode_dosen : varchar(255)
nama_dosen : varchar(255)

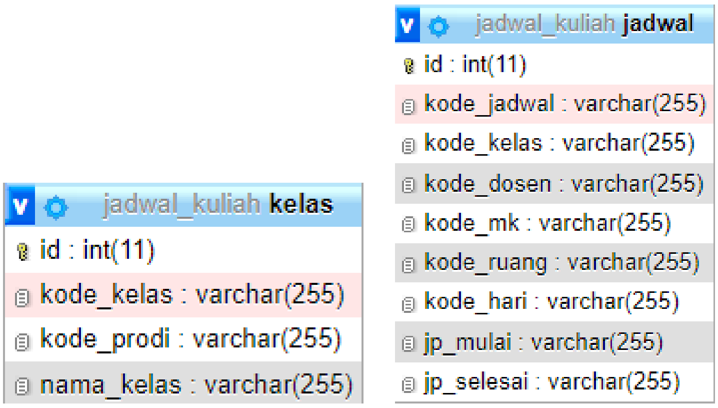
Tabel yang digunakan



nama_kelas
MI-1D
TI-2A
TI-2B
MI-1D

Preview hasil

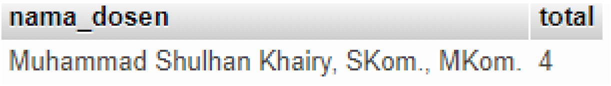
2. Pak Khairy ingin melihat jumlah kelas yang diajar



jadwal_kuliah jadwal
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah kelas
id : int(11)
kode_kelas : varchar(255)
kode_prodi : varchar(255)
nama_kelas : varchar(255)

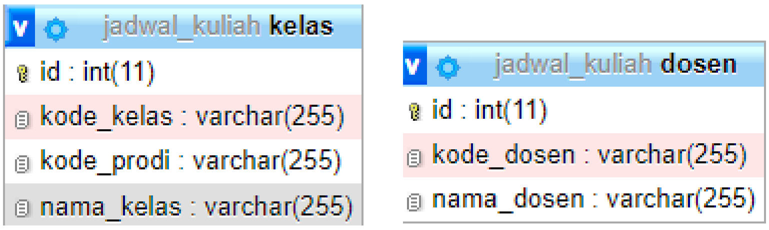
Tabel yang digunakan



nama_dosen	total
Muhammad Shulhan Khairy, SKom., MKom.	4

Preview hasil

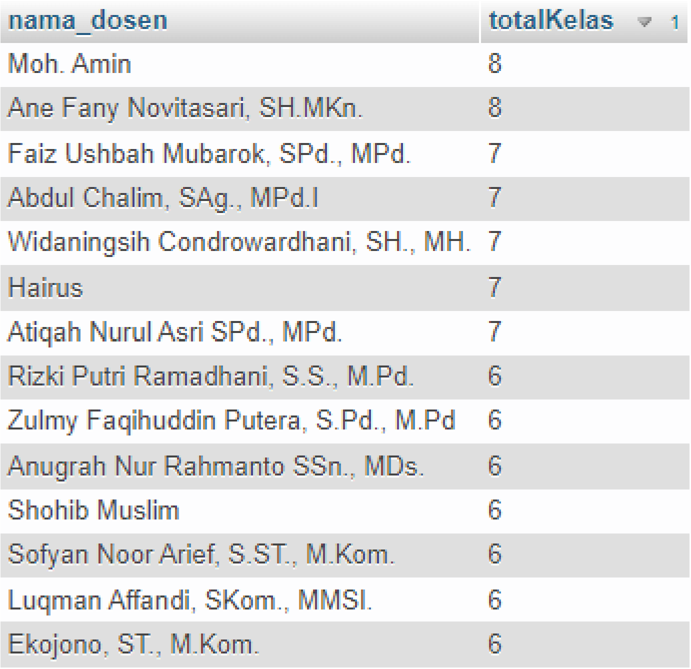
3. Admin ingin menampilkan nama dosen yang mengajar di 5 kelas atau lebih



jadwal_kuliah **kelas**
- id : int(11)
- kode_kelas : varchar(255)
- kode_prodi : varchar(255)
- nama_kelas : varchar(255)

jadwal_kuliah **dosen**
- id : int(11)
- kode_dosen : varchar(255)
- nama_dosen : varchar(255)

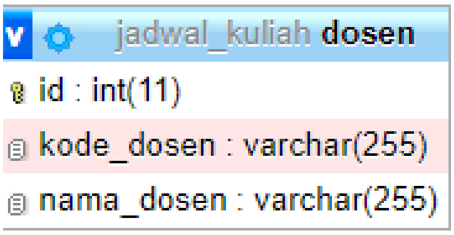
Tabel yang digunakan



nama_dosen	totalKelas ▼ 1
Moh. Amin	8
Ane Fany Novitasari, SH.MKn.	8
Faiz Ushbah Mubarok, SPd., MPd.	7
Abdul Chalim, SAg., MPd.I	7
Widaningsih Condrowardhani, SH., MH.	7
Hairus	7
Atiqah Nurul Asri SPd., MPd.	7
Rizki Putri Ramadhani, S.S., M.Pd.	6
Zulmy Faqihuddin Putera, S.Pd., M.Pd	6
Anugrah Nur Rahmanto SSn., MDs.	6
Shohib Muslim	6
Sofyan Noor Arief, S.ST., M.Kom.	6
Luqman Affandi, SKom., MMSI.	6
Ekojono, ST., M.Kom.	6

Preview hasil

4. Admin ingin melihat nama dosen yang memiliki gelar M.Kom



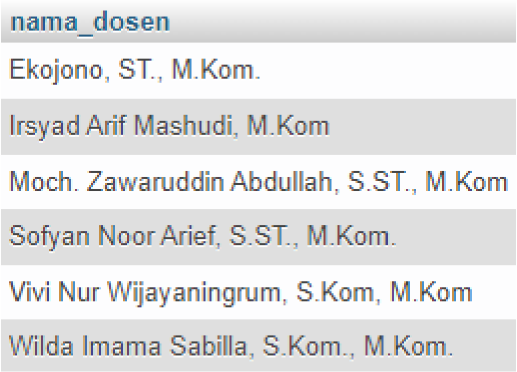
jadwal_kuliah dosen

id : int(11)

kode_dosen : varchar(255)

nama_dosen : varchar(255)

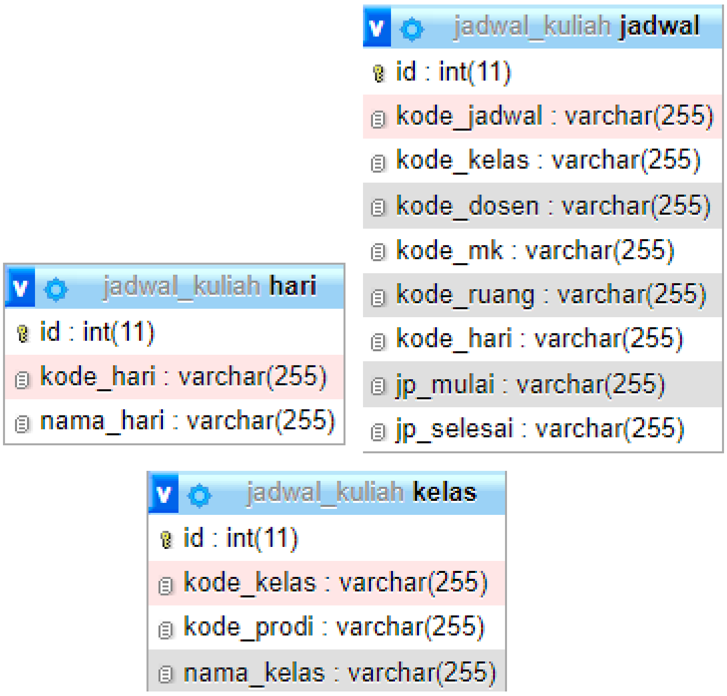
Tabel yang Digunakan



nama_dosen
Ekojono, ST., M.Kom.
Irsyad Arif Mashudi, M.Kom
Moch. Zawaruddin Abdullah, S.ST., M.Kom
Sofyan Noor Arief, S.ST., M.Kom.
Vivi Nur Wijayaningrum, S.Kom, M.Kom
Wilda Imama Sabilla, S.Kom., M.Kom.

Preview hasil

5. Dosen ingin mengetahui nama kelas yang memiliki jadwal kuliah di hari senin



jadwal_kuliah jadwal

id : int(11)

kode_jadwal : varchar(255)

kode_kelas : varchar(255)

kode_dosen : varchar(255)

kode_mk : varchar(255)

kode_ruang : varchar(255)

kode_hari : varchar(255)

jp_mulai : varchar(255)

jp_selesai : varchar(255)

jadwal_kuliah hari

id : int(11)

kode_hari : varchar(255)

nama_hari : varchar(255)

jadwal_kuliah kelas

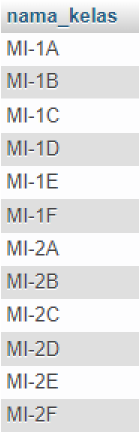
id : int(11)

kode_kelas : varchar(255)

kode_prodi : varchar(255)

nama_kelas : varchar(255)

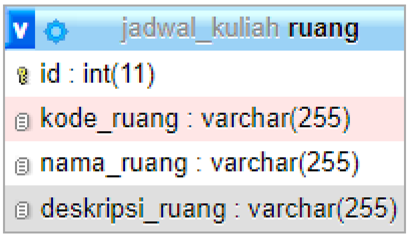
Tabel yang digunakan



nama_kelas
MI-1A
MI-1B
MI-1C
MI-1D
MI-1E
MI-1F
MI-2A
MI-2B
MI-2C
MI-2D
MI-2E
MI-2F

Preview Hasil

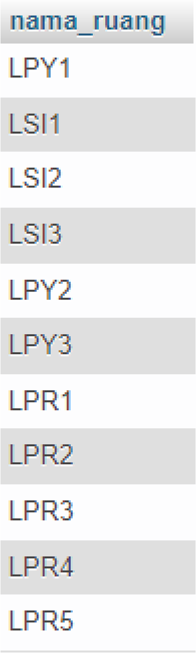
6. Pak Khairy ingin melihat jadwal mengajar untuk hari Senin sampai Jumat



jadwal_kuliah **ruang**

- id : int(11)
- kode_ruang : varchar(255)
- nama_ruang : varchar(255)
- deskripsi_ruang : varchar(255)

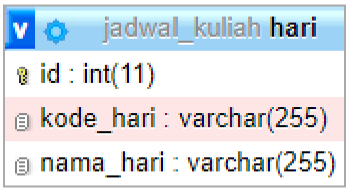
Tabel yang digunakan



nama_ruang
LPY1
LSI1
LSI2
LSI3
LPY2
LPY3
LPR1
LPR2
LPR3
LPR4
LPR5

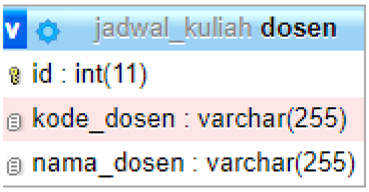
Preview Hasil

7. Pak Khairy ingin mengetahui daftar nama dosen yang mengajar di hari jumat



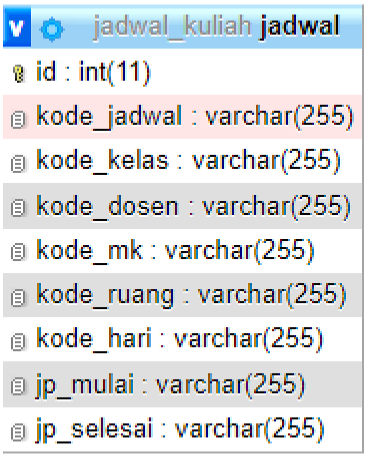
jadwal_kuliah **hari**

- id : int(11)
- kode_hari : varchar(255)
- nama_hari : varchar(255)



jadwal_kuliah **dosen**

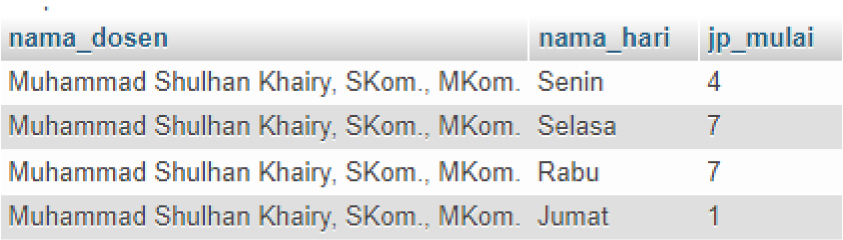
- id : int(11)
- kode_dosen : varchar(255)
- nama_dosen : varchar(255)



jadwal_kuliah **jadwal**

- id : int(11)
- kode_jadwal : varchar(255)
- kode_kelas : varchar(255)
- kode_dosen : varchar(255)
- kode_mk : varchar(255)
- kode_ruang : varchar(255)
- kode_hari : varchar(255)
- jp_mulai : varchar(255)
- jp_selesai : varchar(255)

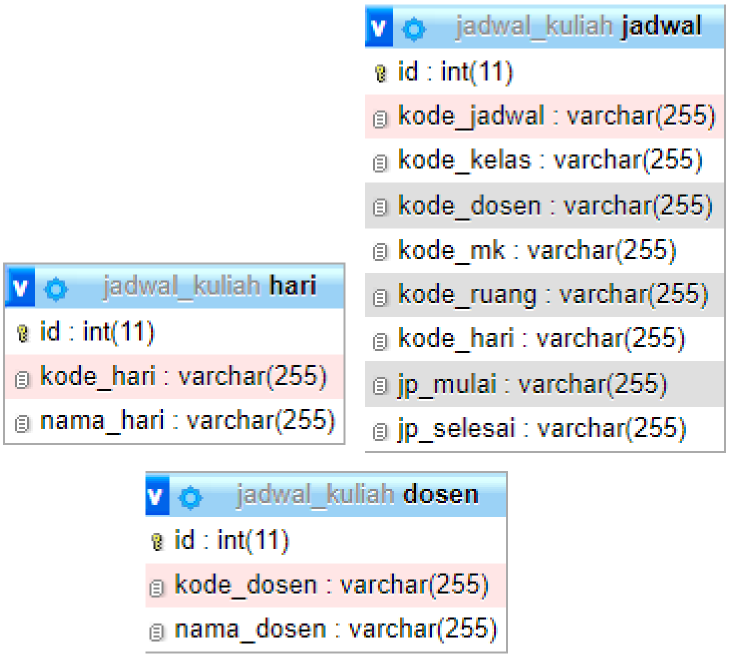
Tabel yang digunakan



nama_dosen	nama_hari	jp_mulai
Muhammad Shulhan Khairy, SKom., MKom.	Senin	4
Muhammad Shulhan Khairy, SKom., MKom.	Selasa	7
Muhammad Shulhan Khairy, SKom., MKom.	Rabu	7
Muhammad Shulhan Khairy, SKom., MKom.	Jumat	1

Preview Hasil

8. Admin ingin melihat daftar nama dosen dan total kelas dari dosen yang paling sering mengajar di hari jumat

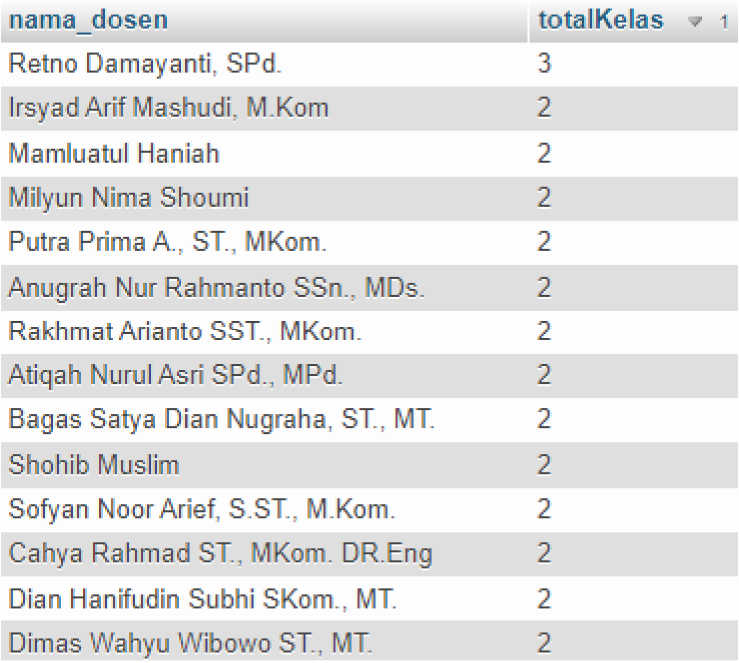


jadwal_kuliah jadwal
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah hari
id : int(11)
kode_hari : varchar(255)
nama_hari : varchar(255)

jadwal_kuliah dosen
id : int(11)
kode_dosen : varchar(255)
nama_dosen : varchar(255)

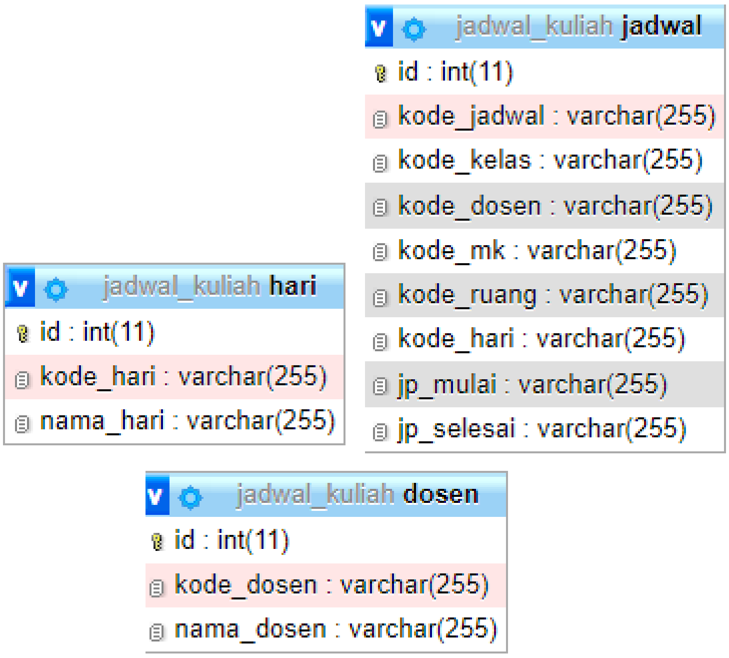
Tabel yang digunakan



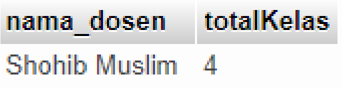
nama_dosen	totalKelas
Retno Damayanti, SPd.	3
Irsyad Arif Mashudi, M.Kom	2
Mamluatul Haniah	2
Milyun Nima Shoumi	2
Putra Prima A., ST., MKom.	2
Anugrah Nur Rahmanto SSn., MDs.	2
Rakhmat Arianto SST., MKom.	2
Atiqah Nurul Asri SPd., MPd.	2
Bagas Satya Dian Nugraha, ST., MT.	2
Shohib Muslim	2
Sofyan Noor Arief, S.ST., M.Kom.	2
Cahya Rahmad ST., MKom. DR.Eng	2
Dian Hanifudin Subhi SKom., MT.	2
Dimas Wahyu Wibowo ST., MT.	2

Preview Hasil

9. Tampilkan nama dosen dan total kelas yang diajar oleh dosen yang paling banyak mengajar di hari senin.



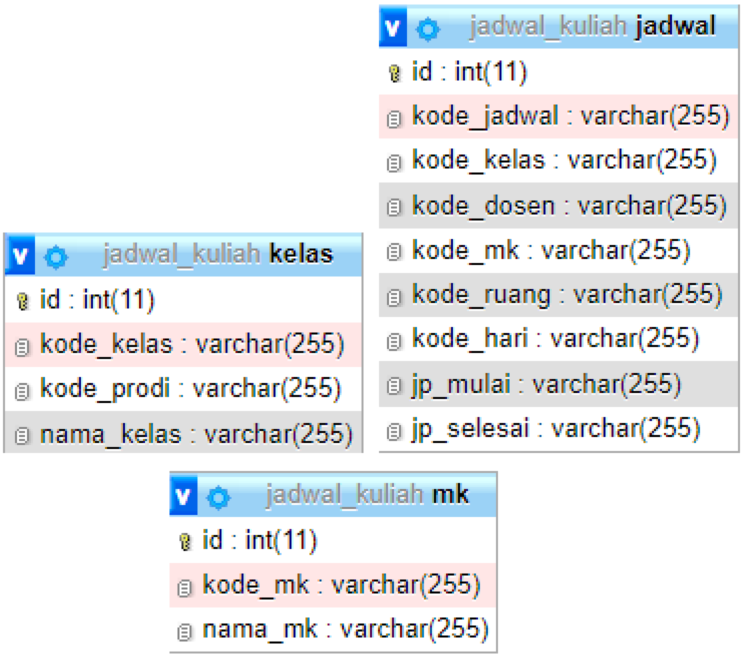
Tabel yang digunakan



nama_dosen	totalKelas
Shohib Muslim	4

Preview Hasil

10. Tampilkan jumlah kelas yang sedang belajar matakuliah yang berkaitan dengan “bahasa”

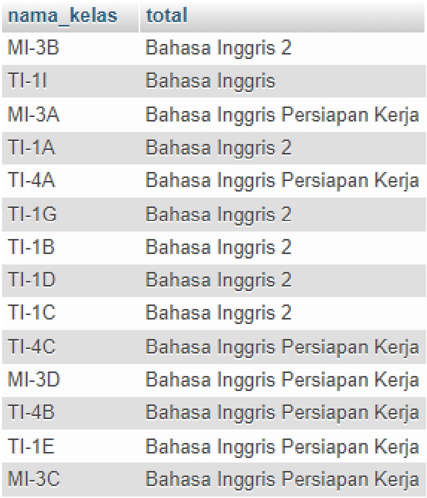


jadwal_kuliah jadwal
- id : int(11)
- kode_jadwal : varchar(255)
- kode_kelas : varchar(255)
- kode_dosen : varchar(255)
- kode_mk : varchar(255)
- kode_ruang : varchar(255)
- kode_hari : varchar(255)
- jp_mulai : varchar(255)
- jp_selesai : varchar(255)

jadwal_kuliah kelas
- id : int(11)
- kode_kelas : varchar(255)
- kode_prodi : varchar(255)
- nama_kelas : varchar(255)

jadwal_kuliah mk
- id : int(11)
- kode_mk : varchar(255)
- nama_mk : varchar(255)

Tabel yang digunakan

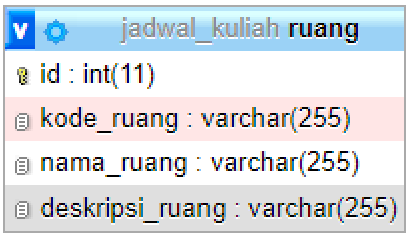


nama_kelas	total
MI-3B	Bahasa Inggris 2
TI-1I	Bahasa Inggris
MI-3A	Bahasa Inggris Persiapan Kerja
TI-1A	Bahasa Inggris 2
TI-4A	Bahasa Inggris Persiapan Kerja
TI-1G	Bahasa Inggris 2
TI-1B	Bahasa Inggris 2
TI-1D	Bahasa Inggris 2
TI-1C	Bahasa Inggris 2
TI-4C	Bahasa Inggris Persiapan Kerja
MI-3D	Bahasa Inggris Persiapan Kerja
TI-4B	Bahasa Inggris Persiapan Kerja
TI-1E	Bahasa Inggris Persiapan Kerja
MI-3C	Bahasa Inggris Persiapan Kerja

Preview Hasil

Attachment 2 - SQLearn Exercises

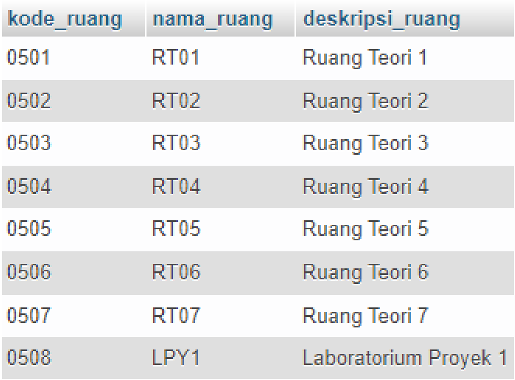
1. Dosen ingin melihat daftar mahasiswa dengan kode_ruang antara 0501 dan 0508. Data yang dibutuhkan cukup kode_ruang, nama_ruang, dan deskripsi_ruang



jadwal_kuliah **ruang**

- id : int(11)
- kode_ruang : varchar(255)
- nama_ruang : varchar(255)
- deskripsi_ruang : varchar(255)

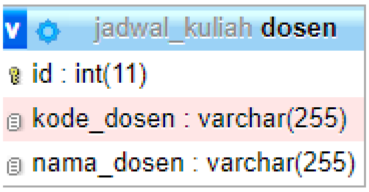
Tabel yang digunakan



kode_ruang	nama_ruang	deskripsi_ruang
0501	RT01	Ruang Teori 1
0502	RT02	Ruang Teori 2
0503	RT03	Ruang Teori 3
0504	RT04	Ruang Teori 4
0505	RT05	Ruang Teori 5
0506	RT06	Ruang Teori 6
0507	RT07	Ruang Teori 7
0508	LPY1	Laboratorium Proyek 1

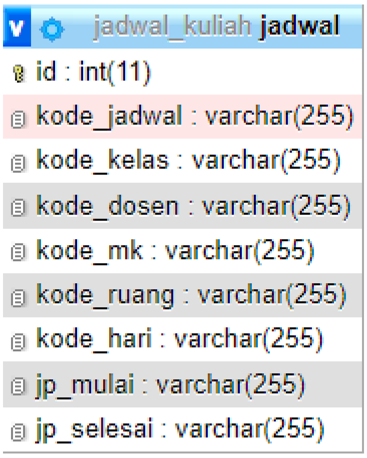
Hasil SQL Query

2. Tampilkan Daftar dosen yang mengajar di prodi MI



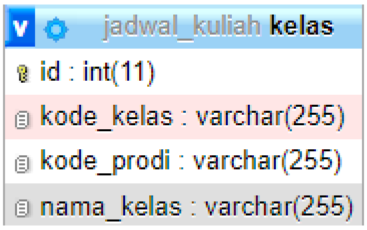
jadwal_kuliah **dosen**

- id : int(11)
- kode_dosen : varchar(255)
- nama_dosen : varchar(255)



jadwal_kuliah **jadwal**

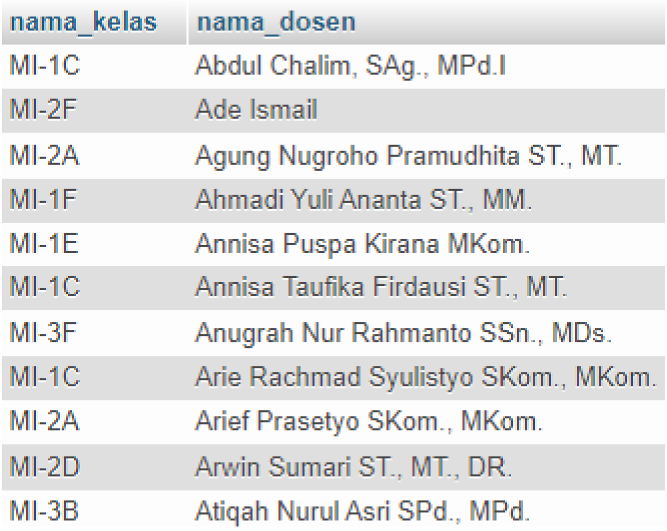
- id : int(11)
- kode_jadwal : varchar(255)
- kode_kelas : varchar(255)
- kode_dosen : varchar(255)
- kode_mk : varchar(255)
- kode_ruang : varchar(255)
- kode_hari : varchar(255)
- jp_mulai : varchar(255)
- jp_selesai : varchar(255)



jadwal_kuliah **kelas**

- id : int(11)
- kode_kelas : varchar(255)
- kode_prodi : varchar(255)
- nama_kelas : varchar(255)

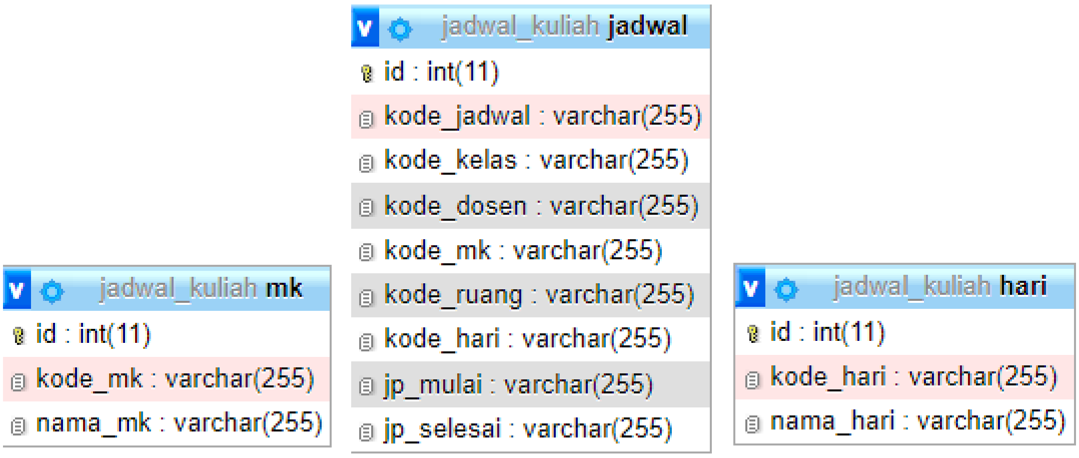
Tabel yang digunakan



nama_kelas	nama_dosen
MI-1C	Abdul Chalim, SAg., MPd.I
MI-2F	Ade Ismail
MI-2A	Agung Nugroho Pramudhita ST., MT.
MI-1F	Ahmadi Yuli Ananta ST., MM.
MI-1E	Annisa Puspa Kirana MKom.
MI-1C	Annisa Taufika Firdausi ST., MT.
MI-3F	Anugrah Nur Rahmanto SSn., MDs.
MI-1C	Arie Rachmad Syulistyo SKom., MKom.
MI-2A	Arief Prasetyo SKom., MKom.
MI-2D	Arwin Sumari ST., MT., DR.
MI-3B	Atiqah Nurul Asri SPd., MPd.

Hasil SQL Query

3. Admin ingin menampilkan daftar mata kuliah Yang diajarkan di Hari Senin

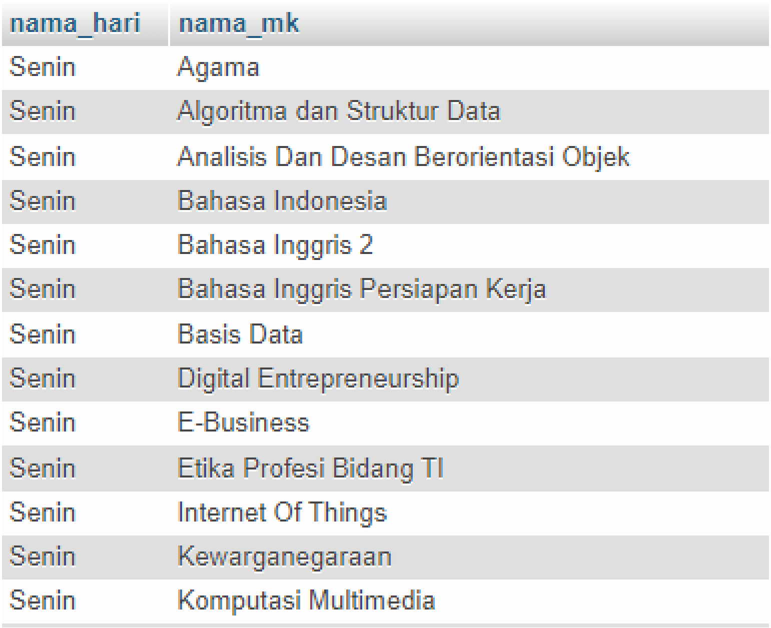


jadwal_kuliah jadwal
- id : int(11)
- kode_jadwal : varchar(255)
- kode_kelas : varchar(255)
- kode_dosen : varchar(255)
- kode_mk : varchar(255)
- kode_ruang : varchar(255)
- kode_hari : varchar(255)
- jp_mulai : varchar(255)
- jp_selesai : varchar(255)

jadwal_kuliah mk
- id : int(11)
- kode_mk : varchar(255)
- nama_mk : varchar(255)

jadwal_kuliah hari
- id : int(11)
- kode_hari : varchar(255)
- nama_hari : varchar(255)

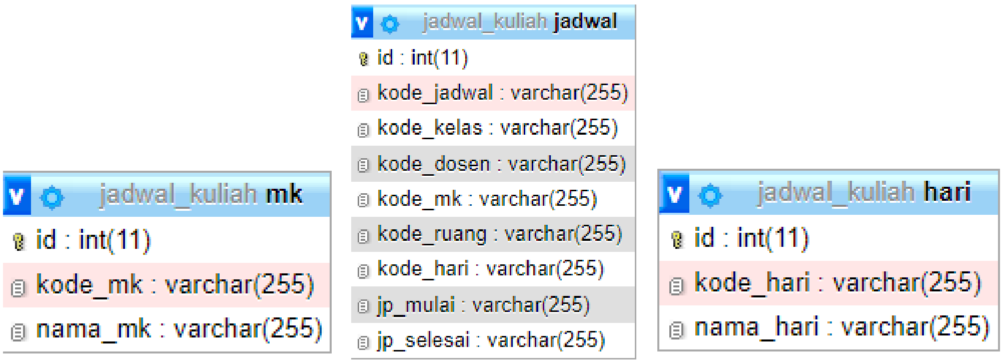
Tabel yang digunakan



nama_hari	nama_mk
Senin	Agama
Senin	Algoritma dan Struktur Data
Senin	Analisis Dan Desan Berorientasi Objek
Senin	Bahasa Indonesia
Senin	Bahasa Inggris 2
Senin	Bahasa Inggris Persiapan Kerja
Senin	Basis Data
Senin	Digital Entrepreneurship
Senin	E-Business
Senin	Etika Profesi Bidang TI
Senin	Internet Of Things
Senin	Kewarganegaraan
Senin	Komputasi Multimedia

Preview Hasil Query

4. Daftar Mata Kuliah Yang diajarkan di hari senin dan tampilkan jumlah kelas yang mempelajari matakuliah tersebut

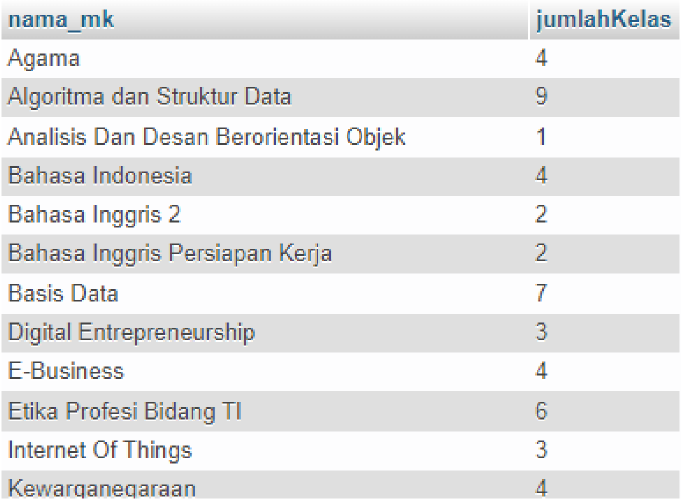


jadwal_kuliah mk
id : int(11)
kode_mk : varchar(255)
nama_mk : varchar(255)

jadwal_kuliah jadwal
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah hari
id : int(11)
kode_hari : varchar(255)
nama_hari : varchar(255)

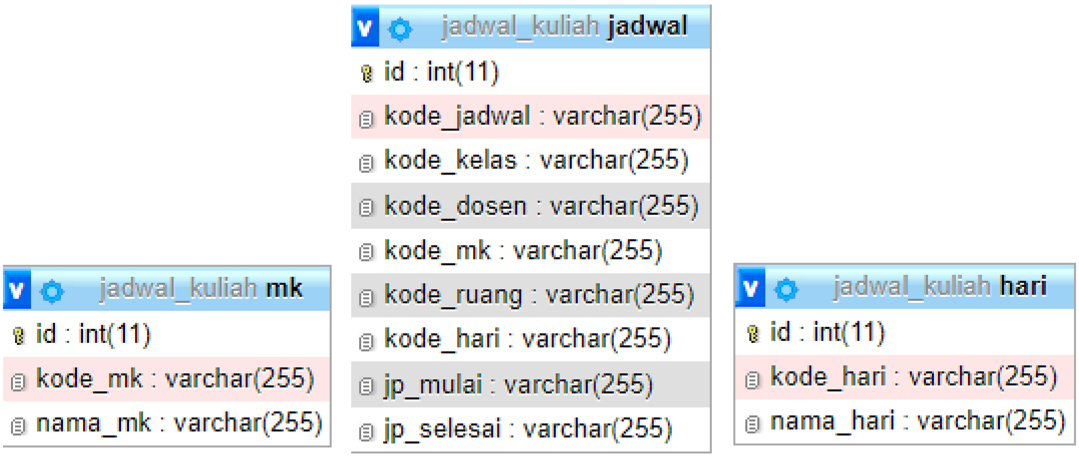
Tabel yang digunakan



nama_mk	jumlahKelas
Agama	4
Algoritma dan Struktur Data	9
Analisis Dan Desan Berorientasi Objek	1
Bahasa Indonesia	4
Bahasa Inggris 2	2
Bahasa Inggris Persiapan Kerja	2
Basis Data	7
Digital Entrepreneurship	3
E-Business	4
Etika Profesi Bidang TI	6
Internet Of Things	3
Kewarganegaraan	4

Preview Hasil Query

5. Daftar Mata Kuliah Yang diajarkan di Hari Kamis dan tampilkan jumlah kelas yang mempelajari matakuliah tersebut.

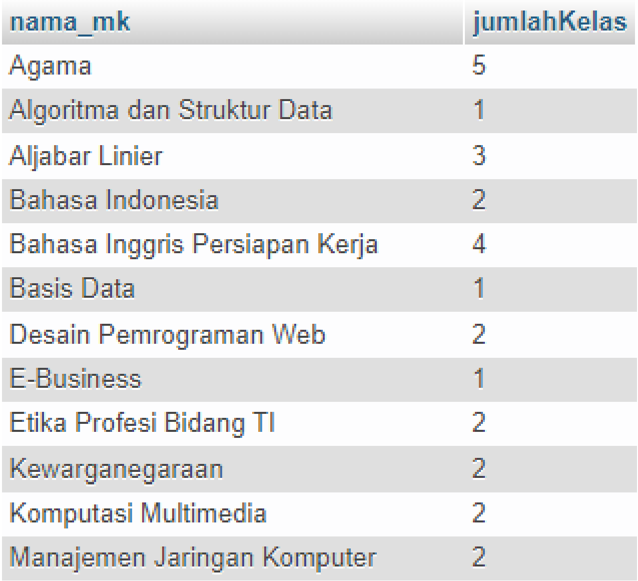


jadwal_kuliah mk
id : int(11)
kode_mk : varchar(255)
nama_mk : varchar(255)

jadwal_kuliah jadwal
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah hari
id : int(11)
kode_hari : varchar(255)
nama_hari : varchar(255)

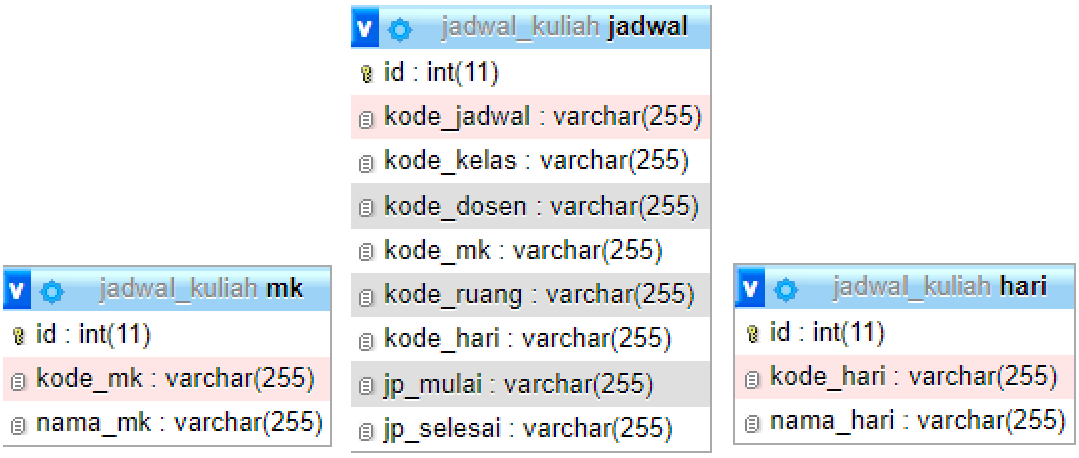
Tabel yang digunakan



nama_mk	jumlahKelas
Agama	5
Algoritma dan Struktur Data	1
Aljabar Linier	3
Bahasa Indonesia	2
Bahasa Inggris Persiapan Kerja	4
Basis Data	1
Desain Pemrograman Web	2
E-Business	1
Etika Profesi Bidang TI	2
Kewarganegaraan	2
Komputasi Multimedia	2
Manajemen Jaringan Komputer	2

Preview Hasil Query

6. Tampilkan nama matakuliah yang paling sering diajarkan di hari senin dan jumlah kelas yang mempelajarinya

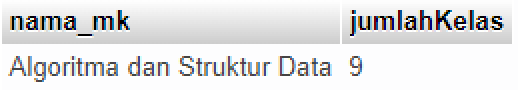


jadwal_kuliah jadwal
- id : int(11)
- kode_jadwal : varchar(255)
- kode_kelas : varchar(255)
- kode_dosen : varchar(255)
- kode_mk : varchar(255)
- kode_ruang : varchar(255)
- kode_hari : varchar(255)
- jp_mulai : varchar(255)
- jp_selesai : varchar(255)

jadwal_kuliah mk
- id : int(11)
- kode_mk : varchar(255)
- nama_mk : varchar(255)

jadwal_kuliah hari
- id : int(11)
- kode_hari : varchar(255)
- nama_hari : varchar(255)

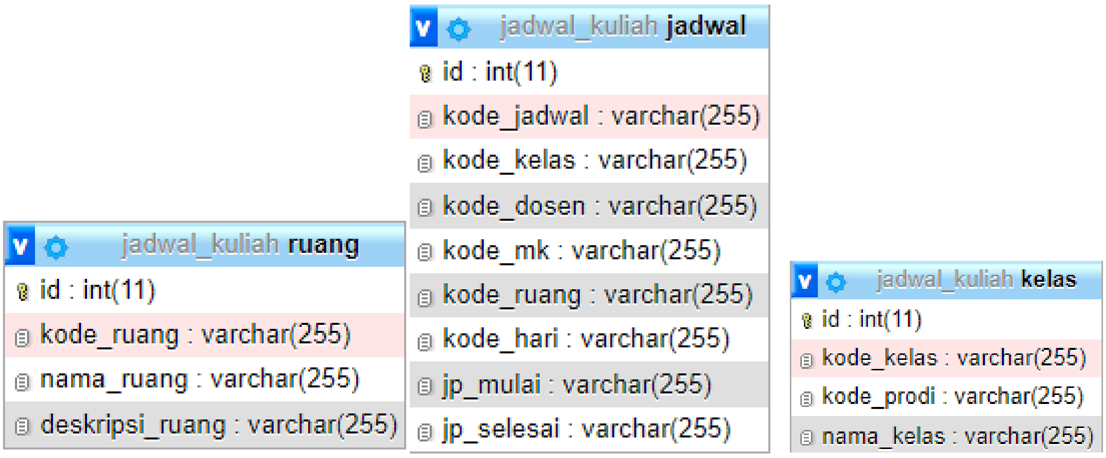
Tabel yang digunakan



nama_mk	jumlahKelas
Algoritma dan Struktur Data	9

Preview Hasil Query

7. Tampilkan nama ruangan yang digunakan oleh prodi MI

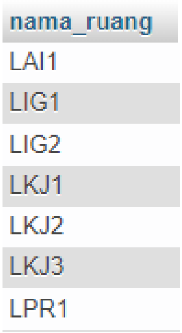


jadwal_kuliah **ruang**
id : int(11)
kode_ruang : varchar(255)
nama_ruang : varchar(255)
deskripsi_ruang : varchar(255)

jadwal_kuliah **jadwal**
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah **kelas**
id : int(11)
kode_kelas : varchar(255)
kode_prodi : varchar(255)
nama_kelas : varchar(255)

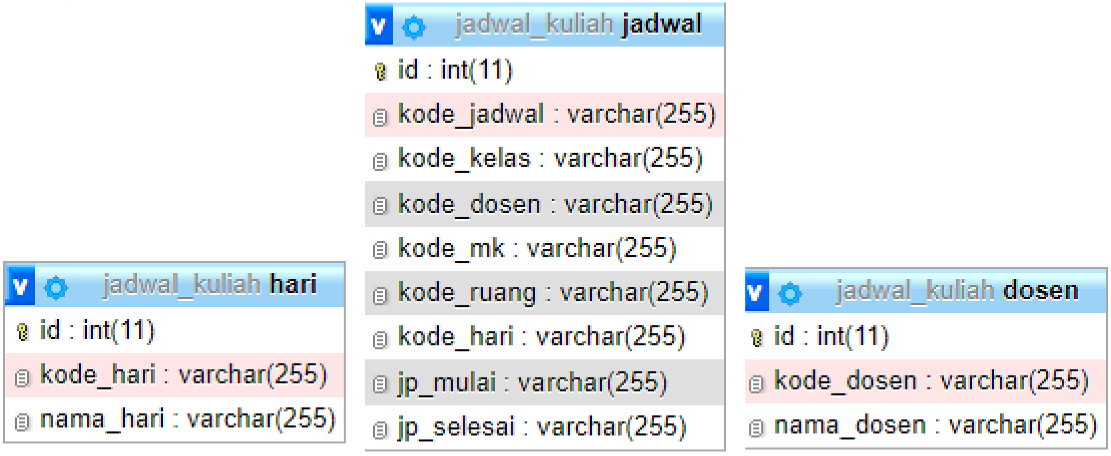
Tabel yang digunakan



nama_ruang
LAI1
LIG1
LIG2
LKJ1
LKJ2
LKJ3
LPR1

Preview Hasil Query

8. Admin ingin melihat daftar nama dosen dan total kelas dari dosen yang paling sering mengajar di hari jumat

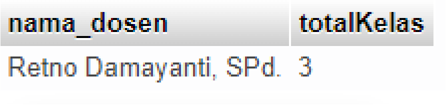


jadwal_kuliah **hari**
id : int(11)
kode_hari : varchar(255)
nama_hari : varchar(255)

jadwal_kuliah **jadwal**
id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

jadwal_kuliah **dosen**
id : int(11)
kode_dosen : varchar(255)
nama_dosen : varchar(255)

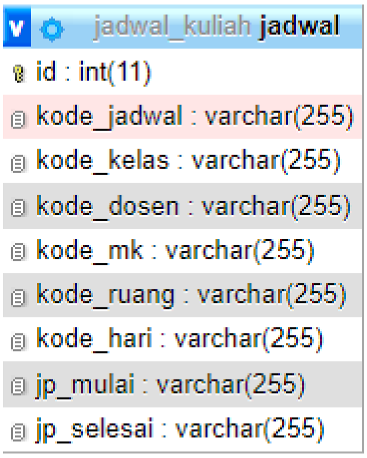
Tabel yang digunakan



nama_dosen	totalKelas
Retno Damayanti, SPd.	3

Preview Hasil Query

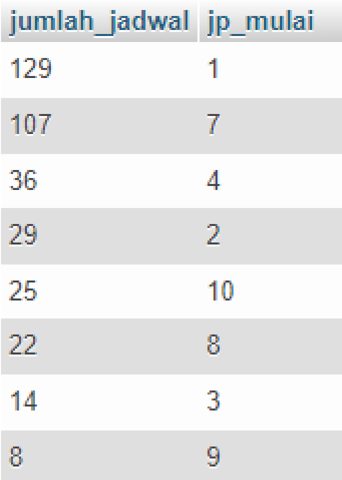
9. Tampilkan jam pelajaran mulai yang paling sering muncul di jadwal kuliah



jadwal_kuliah **jadwal**

id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

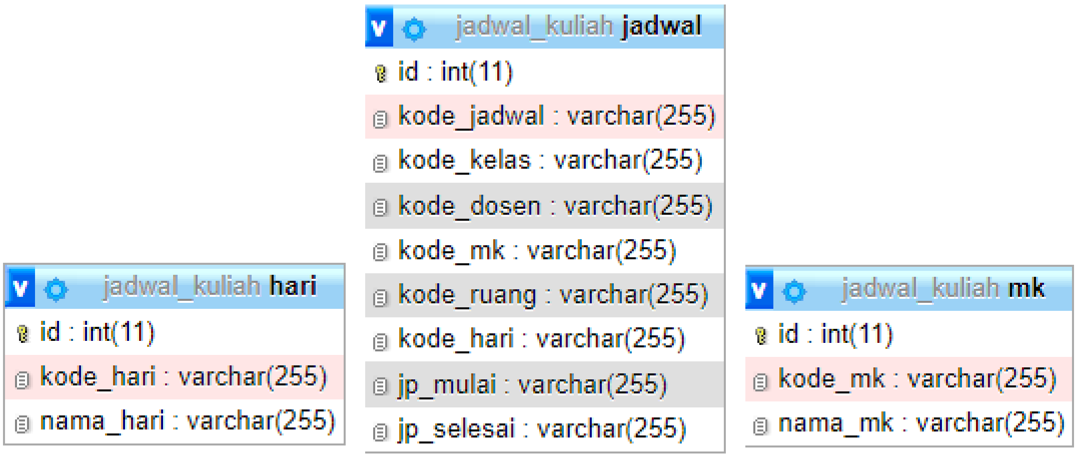
Tabel yang digunakan



jumlah_jadwal	jp_mulai
129	1
107	7
36	4
29	2
25	10
22	8
14	3
8	9

Preview Hasil Query

10. Admin ingin melihat persebaran jumlah mata kuliah yang aktif diajarkan mulai Hari Senin sampai Jumat



jadwal_kuliah **jadwal**

id : int(11)
kode_jadwal : varchar(255)
kode_kelas : varchar(255)
kode_dosen : varchar(255)
kode_mk : varchar(255)
kode_ruang : varchar(255)
kode_hari : varchar(255)
jp_mulai : varchar(255)
jp_selesai : varchar(255)

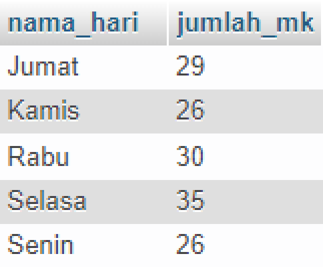
jadwal_kuliah **hari**

id : int(11)
kode_hari : varchar(255)
nama_hari : varchar(255)

jadwal_kuliah **mk**

id : int(11)
kode_mk : varchar(255)
nama_mk : varchar(255)

Tabel yang digunakan



nama_hari	jumlah_mk
Jumat	29
Kamis	26
Rabu	30
Selasa	35
Senin	26

Preview Hasil Query