

SISTEM ASESMEN PEMBELAJARAN SQL DENGAN METODE COSINE SIMILARITY DAN SQL PARSER BERBASIS WEBSITE

Triyono Rifan¹⁾, Fawwaz Ali Akbar^{*2)}, Afina Lina Nurlaili³⁾

1. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
2. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia
3. Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional “Veteran” Jawa Timur, Indonesia

Article Info

Kata Kunci: *Cosine Similarity; Pembelajaran SQL; Sistem Asesmen; SQL Parser; Query SQL;*

Keywords: *Cosine Similarity; Sistem Asesmen; SQL Learning; SQL Parser; Query SQL*

Article history:

Received 10 September 2024

Revised 15 Oktober 2024

Accepted 5 November 2024

Available online 4 December 2024

DOI :

<https://doi.org/10.29100/jipi.v9i4.5630>

* Corresponding author.

Fawwaz Ali Akbar

E-mail address:

fawwaz_ali.fik@upnjatim.ac.id

ABSTRAK

Perkembangan teknologi informasi dan komunikasi telah mempengaruhi berbagai aspek kehidupan, termasuk pada bidang pendidikan. Penilaian esai, khususnya dalam mata pelajaran teknis seperti pemrograman SQL, menghadapi tantangan efisiensi dan akurasi dalam penilaian manual. Penelitian ini bertujuan untuk mengembangkan dan menguji sistem penilaian otomatis berbasis web yang menggunakan SQL Parser dan algoritma Cosine Similarity. Sistem ini dirancang untuk mengevaluasi query SQL siswa secara efisien dan akurat, mengurangi beban kerja pengajar, serta memberikan umpan balik cepat dan konstruktif kepada siswa. Metode penelitian melibatkan analisis arsitektur sistem, perancangan diagram UML, dan penerapan SQL Parser untuk tokenisasi query SQL, diikuti dengan perhitungan kemiripan menggunakan Cosine Similarity. Pengujian sistem menggunakan teknik *black box* menunjukkan bahwa sistem dapat menilai jawaban siswa dengan akurasi yang baik, diukur melalui nilai MAE dan RMSE yang rendah, serta mampu memberikan penilaian dalam waktu yang cepat. Hasil penelitian ini menunjukkan bahwa sistem penilaian otomatis ini efektif dalam meningkatkan kualitas dan efisiensi proses pembelajaran SQL di pendidikan tinggi.

ABSTRACT

The development of information and communication technology has affected various aspects of life, including education. Essay grading, particularly in technical subjects such as SQL programming, faces efficiency and accuracy challenges in manual grading. This research aims to develop and test a web-based automated assessment system that uses SQL Parser and Cosine Similarity algorithm. The system is designed to efficiently and accurately evaluate students' SQL queries, reduce teachers' workload, and provide quick and constructive feedback to students. The research method involves system architecture analysis, UML diagram design, and implementation of SQL Parser for SQL query tokenization, followed by similarity calculation using Cosine Similarity. System testing using black box techniques shows that the system can assess students' answers with good accuracy, measured through low MAE and RMSE values, and is able to provide assessments in a fast time. The results of this study indicate that this automated scoring system is effective in improving the quality and efficiency of the SQL learning process in higher education.

I. PENDAHULUAN

TEKNOLOGI informasi dan komunikasi telah mengalami perkembangan pesat yang turut memengaruhi berbagai aspek kehidupan, termasuk bidang pendidikan. Salah satu inovasi penting dalam pendidikan adalah pengembangan sistem asesmen otomatis, yang dirancang untuk membantu pengajar dalam mengevaluasi kemampuan siswa secara lebih efisien dan akurat [1]. Sistem asesmen ini mencakup berbagai bentuk, termasuk ujian pilihan ganda dan esai, dengan tujuan memberikan umpan balik yang konstruktif dan cepat kepada siswa [2].

Ujian esai memegang peranan penting dalam mengevaluasi pemahaman dan kemampuan analitis siswa [3]. Namun, penilaian manual terhadap jawaban esai memiliki beberapa tantangan signifikan berupa ketidakefisienan dan potensi ketidakakuratan dalam penilaian [4]. Pengajar perlu menghabiskan banyak waktu dan rentan terhadap bias subjektif untuk membaca dan menilai setiap jawaban, serta menghadapi variasi dalam cara siswa mengekspresikan pemahaman mereka. Hal ini terutama berlaku untuk mata pelajaran teknis seperti basis data, di

mana query SQL sering kali memiliki variasi yang sah dalam penulisan tetapi menghasilkan output yang sama [5].

Structured Query Language (SQL) adalah bahasa pemrograman yang digunakan untuk mengelola dan memanipulasi basis data [6]. Penguasaan SQL merupakan keterampilan penting bagi mahasiswa di bidang informatika. Dalam lingkungan pendidikan, penilaian kemampuan SQL sering kali dilakukan melalui ujian yang meminta siswa menulis query SQL. Tantangan dalam penilaian manual query SQL terletak pada variasi jawaban yang mungkin memiliki struktur berbeda tetapi tetap valid secara fungsional. Penilaian manual terhadap query semacam ini tidak hanya memakan waktu tetapi juga rentan terhadap ketidakakuratan dan inkonsistensi [4].

Untuk mengatasi tantangan ini, sistem penilaian otomatis berbasis web yang menggunakan SQL Parser dan algoritma *Cosine Similarity* dapat menjadi solusi yang efektif. SQL Parser berfungsi untuk memecah query menjadi komponen-komponen yang lebih kecil dan menganalisisnya secara sintaksis dan semantis [4] [5]. SQL parser memiliki kemampuan untuk memecahkan sebuah query SQL menjadi token untuk mengetahui informasi query tersebut. Query SQL dipecah menjadi token-token yang diperlukan pada proses parsing atau pemecahan setiap kata *query SQL* [8]. Ketika token terdeteksi, parser yang dilakukan saat itu dibuat untuk menyimpan nama tabel dan kolom yang terkait dengan token tersebut [6]. Sementara itu, algoritma *Cosine Similarity* digunakan untuk mengukur kemiripan antara jawaban siswa dan kunci jawaban dengan menghitung seberapa dekat kedua query tersebut dalam ruang vektor [7].

Untuk mendukung proses pembelajaran yang lebih efisien dalam menulis query SQL, pada sistem asesmen ini didukung adanya sebuah teks editor untuk memudahkan mahasiswa untuk menulis dan mengirimkan tugas kode mereka secara online. Sistem ini memungkinkan pengajar untuk memantau langkah-langkah yang diambil oleh siswa dalam memecahkan masalah penulisan kode [8]. Misalnya, SQL Formulation Editor (SQL-FE) yang dikembangkan oleh Al-Salmi telah menunjukkan manfaat signifikan dalam membantu mahasiswa merumuskan query SQL dengan tepat [8].

Implementasi sistem ini tidak hanya bertujuan untuk menghemat waktu pengajar dalam menilai jawaban tetapi juga untuk meningkatkan akurasi dan konsistensi penilaian. Dengan umpan balik yang cepat dan objektif, siswa dapat segera mengetahui kekurangan dalam jawaban mereka dan memperbaikinya. Hal ini mendukung proses pembelajaran yang lebih efektif dan efisien.

Penelitian ini bertujuan untuk mengembangkan dan menguji sistem penilaian otomatis berbasis web untuk query SQL, yang diharapkan dapat memberikan kontribusi signifikan terhadap pendidikan tinggi di bidang informatika. Sistem ini diharapkan dapat membantu pengajar dalam penilaian esai SQL, mengurangi beban kerja manual, dan memberikan umpan balik yang lebih baik kepada siswa. Melalui penerapan teknologi ini, diharapkan kualitas dan efisiensi proses pembelajaran SQL dapat ditingkatkan secara signifikan.

II. METODE PENELITIAN

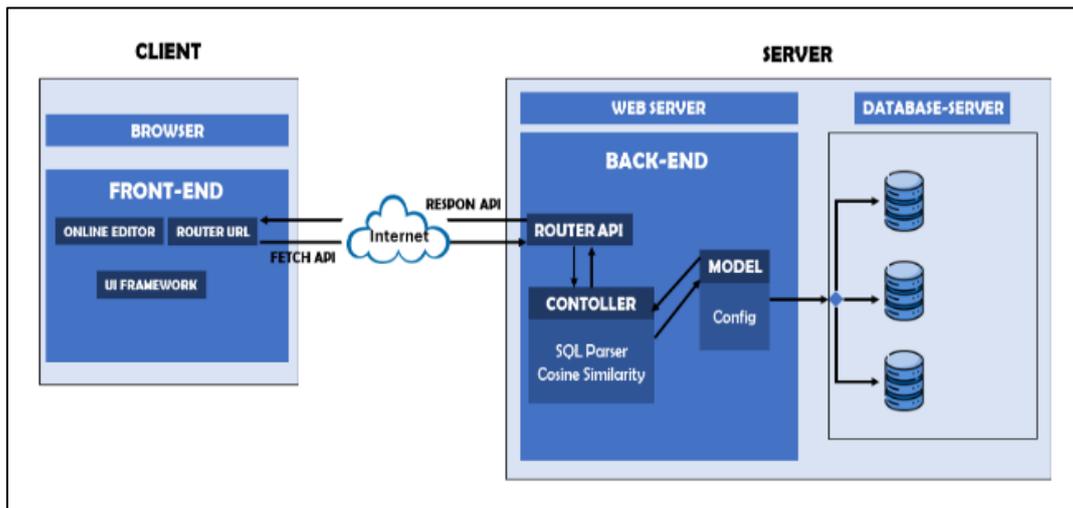
Metode yang digunakan pada perhitungan persentase kemiripan jawaban dan kunci jawaban menggunakan metode *Cosine Similarity* yang sebelumnya telah diproses pada tahapan analisa kode dengan SQL Parser. Adapun alur penelitian yang digambarkan pada Gambar 1.



Gambar 1. Diagram Alur Penelitian

A. Tahap Analisis dan Perancangan

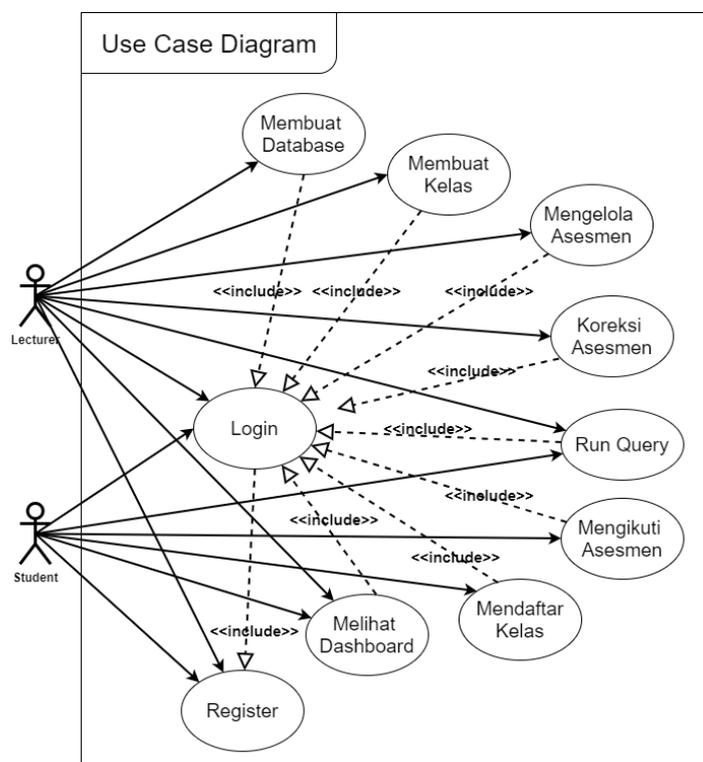
Pada tahap analisis dan perancangan terdapat analisis terhadap arsitektur sistem, diagram UML seperti use case diagram untuk menentukan aksesibilitas pengguna pada sistem yang dibangun dan menjadi focus utama pada proses perancangan sistem asesmen, dan analisis proses bisnis sistem. Adapun analisis arsitektur sistem asesmen pada penelitian ini.



Gambar 2. Arsitektur Sistem

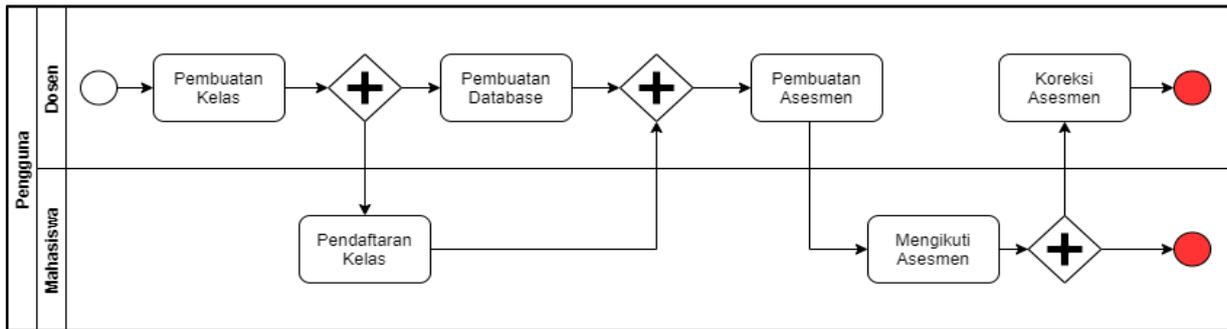
Pada Gambar 2 merupakan arsitektur sistem dengan proses client server. Pada sisi client merupakan tampilan atau *front-end* sistem asesmen yang dibangun dengan tampilan menarik dan terdapat sebuah editor untuk membantu proses penulisan SQL. Terdapat REST API yang menghubungkan data dari client ke server dan sebaliknya. Client akan melakukan request data kepada server dan hasil respon akan dikembalikan oleh server kepada client [12].

Pada sisi server terdapat komponen yang dibutuhkan seperti router api untuk alamat REST API, konsep MVC framework, komponen SQL Parser dan proses Algoritma Cosine Similarity. Aplikasi *back-end* pada web server akan menghubungkan database secara dinamis. Sistem akan melakukan koneksi ke database tertentu untuk melakukan sebuah perintah query yang dikirim oleh client [13]. Perubahan koneksi ke database tertentu dilakukan oleh konfigurasi koneksi pada sisi *back-end model*. Pada sisi pengguna terdapat diagram use case pada Gambar 3.



Gambar 3. Use Case Diagram

Pada Gambar 3 merupakan diagram use case untuk pengguna sistem asesmen yakni sebagai dosen dan mahasiswa. Pada sisi dosen dapat melakukan aktivitas seperti melihat dashboard, membuat data kelas, membuat database melalui sistem asesmen, membuat dan mengoreksi asesmen, login, register. Pada sisi mahasiswa dapat mendaftarkan kelas, mengikuti asesmen, register. Pada kedua pengguna dapat melakukan run query yang bertujuan untuk mengirimkan sebuah query SQL pada sistem.



Gambar 4. Proses Bisnis Sistem

Pada Gambar 4 merupakan proses bisnis sistem asesmen. Proses pertama dilakukan oleh seorang dosen untuk membuat data kelas, lalu dapat lakukan secara parallel antara mahasiswa mendaftarkan kelas dan dosen membuat database, lalu proses pembuatan asesmen dan mahasiswa mengikuti asesmen yang dibuat, serta hasil akhir koreksi asesmen.

B. Tahap Pembangunan Sistem

Pada tahap pembangunan sistem dari hasil analisis dan rancangan yang sudah dibuat. Setelah hasil analisis selesai, dilanjutkan proses penulisan baris kode website sesuai dengan analisis proses bisnis dengan bahasa pemrograman PHP, MySQL dan framework Laravel sebagai penyusun sistem back-end atau server-side, serta tailwind CSS, Vue JS, Monaco Editor sebagai penyusun sistem front-end atau client-side [14].

C. Implementasi Metode SQL Parser dan Cosine Similarity

Proses analisis query SQL dalam penelitian ini dilakukan dengan menggunakan SQL Parser dan algoritma Cosine Similarity. SQL Parser bertugas untuk memecah query SQL menjadi unit-unit kecil yang disebut token [15]. Tabel 1 menjelaskan contoh query jawaban mahasiswa dan kunci jawaban.

TABEL I
 CONTOH QUERY

T(A)	<i>SELECT id, nama, kelas FROM student WHERE id=2</i>
T(B)	<i>select id, kelas, nama from student where id=2</i>

Keterangan:

T(A) = Query kunci jawaban

T(B) = Query Jawaban Mahasiswa

Inisialisasi query SQL pada Tabel 1 akan dipecah menjadi token-token seperti pada Tabel II berikut.

TABEL II
 PROSES SQL PARSER

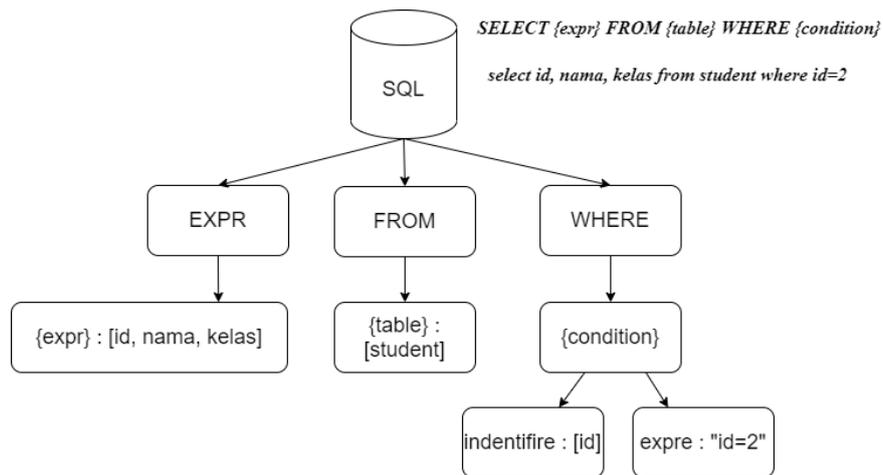
T(A)	<i>[SELECT, id, nama, kelas, FROM, student, WHERE, id, =, 2]</i>
T(B)	<i>[select, id, kelas, nama, from, student, where, id, =, 2]</i>

Keterangan:

T(A) = Query kunci jawaban

T(B) = Query Jawaban Mahasiswa

SQL parser akan memecahkan string SQL seperti Tabel II dan dilakukan tokenisasi dari awal kata query. Pada Tabel II tokenisasi akan diawali oleh kata 'SELECT'[9]. Ketika token tersebut ditemukan maka SQL parser akan menyimpan informasi nama kolom dan akan disimpan pada token SELECT tersebut dan dilakukan untuk token FROM menyimpan informasi nama tabel. Setelah itu, SQL Parser melakukan analisis sintaksis untuk memastikan struktur query sesuai dengan aturan bahasa SQL. Proses ini melibatkan pembangunan struktur pohon sintaksis (parse tree), seperti pada Gambar 5, dari token-token tersebut untuk memverifikasi bahwa query tersebut valid secara sintaksis, misalnya memastikan klausa SELECT diikuti oleh nama kolom dan klausa FROM diikuti oleh nama tabel [9]. Selain itu, SQL Parser juga melakukan analisis semantik untuk memastikan bahwa operasi yang dimaksudkan logis dan konsisten dengan basis data yang ada, seperti memeriksa keberadaan tabel dan kolom dalam basis data serta memahami hubungan antar tabel jika terdapat operasi join.



Gambar 5. Bentuk Hasil SQL Parser

Setelah query SQL dianalisis oleh SQL Parser, hasil tokenisasi dan analisis tersebut diubah menjadi bentuk vektor untuk perhitungan kemiripan. Dalam konteks ini, pembobotan dilakukan dengan memberikan nilai pada setiap token berdasarkan frekuensi kemunculannya dalam query, menggunakan metode Term Frequency (TF) seperti pada Tabel 3, di mana setiap elemen merepresentasikan frekuensi atau bobot token terkait.

TABEL III
 HASIL PEMBOBOTAN SETIAP KATA DARI SQL PARSER

No.	Term	T(A)	T(B)
1.	select	1	1
2.	from	1	1
3.	where	1	1
4.	id	1	1
5.	nama	1	1
6.	kelas	1	1
7.	student	1	1

Algoritma Cosine Similarity kemudian digunakan untuk menghitung tingkat kemiripan antara vektor query siswa dan vektor kunci jawaban. Cosine Similarity mengukur kesamaan antara dua vektor dengan menghitung cosinus dari sudut di antara mereka, menggunakan rumus seperti pada persamaan 1.

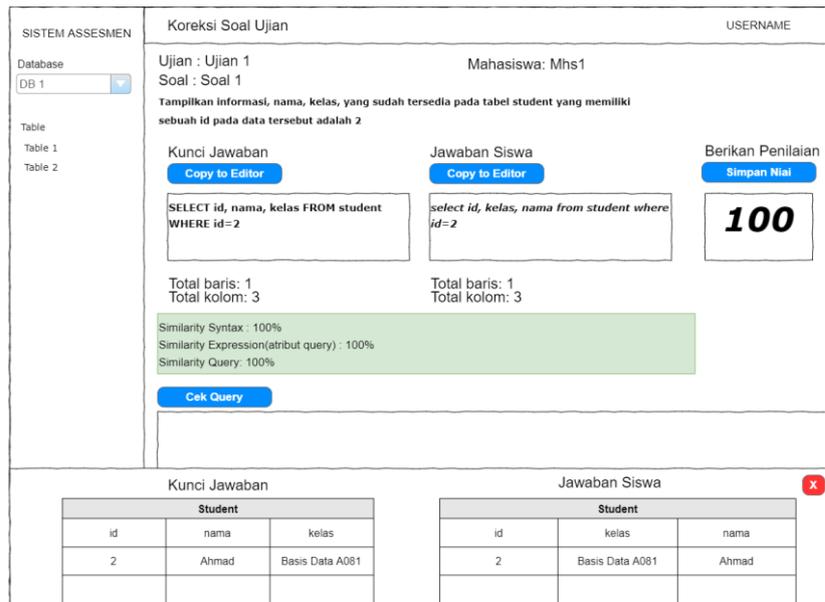
$$\cos = \frac{\sum_{i=1}^N a_i b_i}{\sqrt{\sum_{i=1}^N a_i^2} * \sqrt{\sum_{i=1}^N b_i^2}} \quad (1)$$

Di mana a_i dan b_i adalah elemen-elemen dari vektor query siswa dan kunci jawaban masing-masing. Nilai Cosine Similarity berkisar antara 0 hingga 1, dengan 1 menunjukkan kemiripan sempurna dan 0 menunjukkan tidak ada kemiripan. Sehingga proses perhitungan terhadap gabungan antara syntax SQL dan expression SQL kedua hasil string yakni sebagai berikut.

$$\cos = \frac{(1 \times 1) + (1 \times 1)}{\sqrt{7} * \sqrt{7}}$$

$$\cos = \frac{7}{\sqrt{7} * \sqrt{7}} = \frac{7}{7} = 1$$

Hasil dari metode perhitungan tersebut adalah 1. Dalam studi kasus Tabel 1, kedua string tersebut memiliki nilai kesamaan 1. Artinya kedua string tersebut sebenarnya sama atau serupa. Oleh karena itu, tingkat kemiripannya adalah 100%. Proses pengecekan query pada sistem juga dilakukan secara output query atau hasil data terhadap query seperti *wireframe* Gambar 6.



Gambar 6. Hasil Output Query

Pada Gambar 6 merupakan hasil output query dari jawaban dan kunci jawaban. Sistem asesmen akan secara langsung menampilkan hasil eksekusi query dan menampilkan perbandingan kedua hasil query.

Dalam penelitian ini, SQL Parser digunakan untuk menganalisis kode query SQL dengan mengubah string query SQL menjadi struktur pohon (*parse tree*), memungkinkan pengecekan sintaksis dan semantik secara lebih mudah. Proses tokenisasi memecah query menjadi bagian-bagian kecil seperti tabel, kolom, dan operator, sementara analisis sintaktik memastikan struktur query valid dan analisis semantik memverifikasi logika query [16]. Hasil dari analisis ini kemudian digunakan bersama algoritma *Cosine Similarity* untuk mengukur kemiripan antara jawaban siswa dan kunci jawaban secara objektif dan konsisten. Implementasi ini memastikan bahwa setiap variasi penulisan query yang sah dapat dinilai dengan tepat, memberikan umpan balik yang konstruktif dan cepat kepada siswa, serta mengurangi beban kerja manual pengajar.

D. Uji Coba Sistem

Setelah proses pembangunan selesai, perlu dilakukan pengujian untuk memastikan fungsionalitas seluruh sistem dapat berjalan dengan baik dan sesuai harapan. Dalam penelitian ini dilakukan teknik black box untuk uji coba sistem. Teknik ini berfokus pada fungsionalitas eksternal sistem yang memungkinkan pengujian dalam mensimulasikan perilaku pengguna. Untuk menilai keberhasilan uji coba, metrik yang digunakan seperti akurasi penilaian dengan cara menghitung nilai MAE (*Mean Absolute Error*) dan RMSE (*Root Mean Square Error*) [17] antara nilai cosine dan penilaian seorang dosen pengampu, kecepatan penilaian sistem untuk mengukur seberapa cepat dalam memberikan penilaian jawaban, dan uji coba keseluruhan fungsionalitas yang dapat berjalan dengan baik dan sesuai harapan.

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (2)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (3)$$

Keterangan:

MAE = Mean Absolute Error, RMSE = Root Mean Square Error

x = nilai actual, y = nilai prediksi (cosine)

i = baris, n = jumlah data

III. HASIL DAN PEMBAHASAN

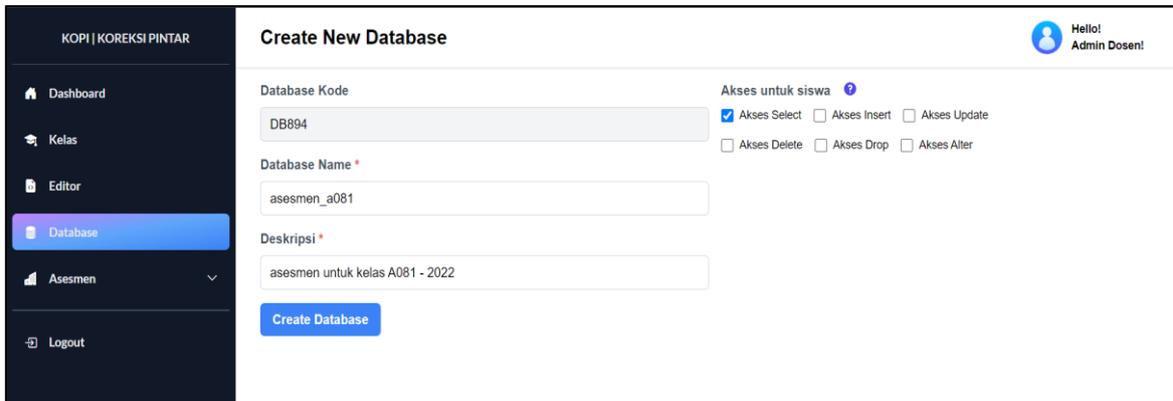
A. Hasil Pembangunan Sistem

Pada hasil penelitian berupa pembangunan sistem asesmen dengan penerapan metode Cosine Similarity dan SQL parser pada proses pengoreksian asesmen. Hasil akhir tampilan pembangunan sistem yaitu halaman login, register, pembuatan kelas, pembuatan database, pembuatan asesmen, dan koreksi asesmen. Arsitektur sistem yang dibangun yakni berupa arsitektur Front-End menggunakan Vue.js dan Back-End menggunakan Laravel yang

terhubung secara konsep Rest API

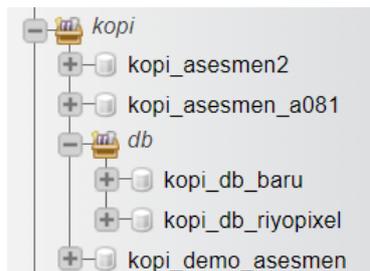
1) Pembuatan Database

Pembuatan Database digunakan untuk dosen dapat membuat database sesuai dengan kebutuhan asesmen. Sistem akan membuat secara otomatis pada database server sesuai dengan data yang dikirim oleh dosen seperti Gambar 10.



Gambar 7. Pembuatan Database

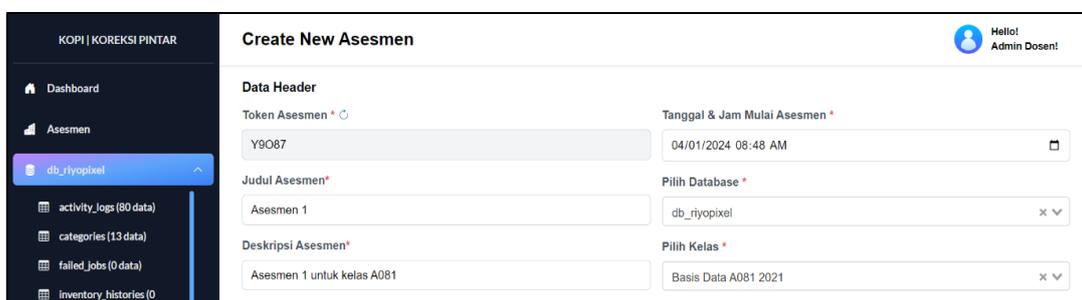
Pada Gambar 10, dosen dapat mengisi data untuk database yang diinginkan, terdapat nama database, deksripsi dan akses siswa. Akses siswa digunakan untuk pencegahan query dari mahasiswa. Pada Gambar 10, mahasiswa hanya dapat mengirimkan query *statement select* data. Sehingga apabila terdapat query insert data atau update data dari mahasiswa tidak dapat dijalankan oleh aplikasi *backend*. Hasil sistem saat membuat database akan memiliki prefix nama database di database server untuk kemudahan grouping seperti Gambar 11.



Gambar 8. Hasil Pembuatan Database di Database Server

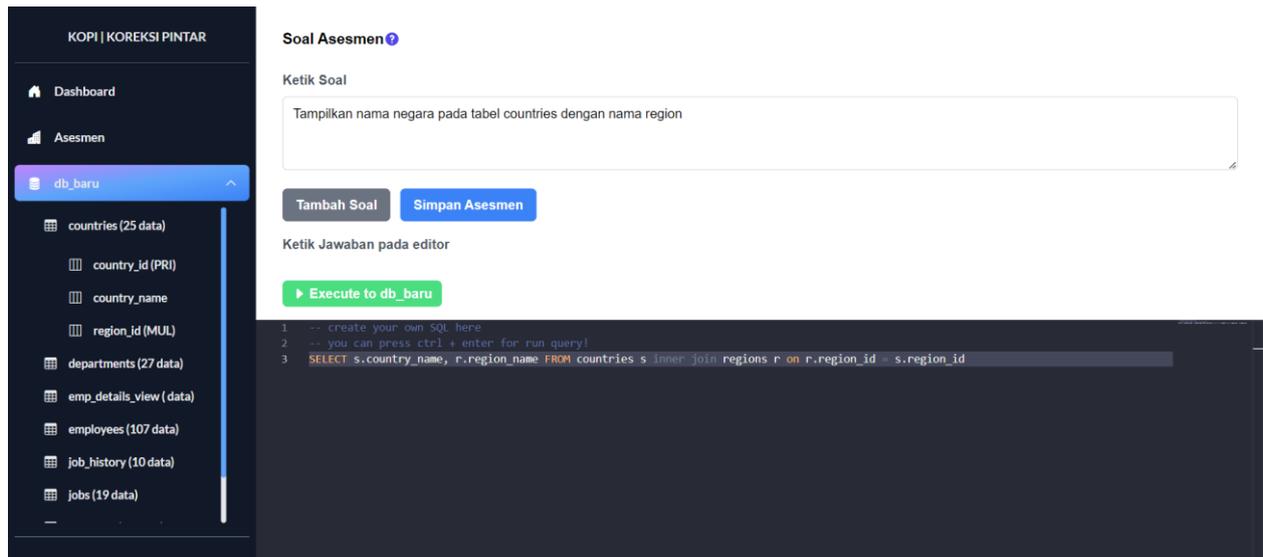
2) Pembuatan Asesmen

Pembuatan asesmen merupakan fitur utama pada sistem yang telah dibangun. Pada halaman ini dosen dapat membuat agenda asesmen, soal dan kunci jawaban seperti Gambar 12.



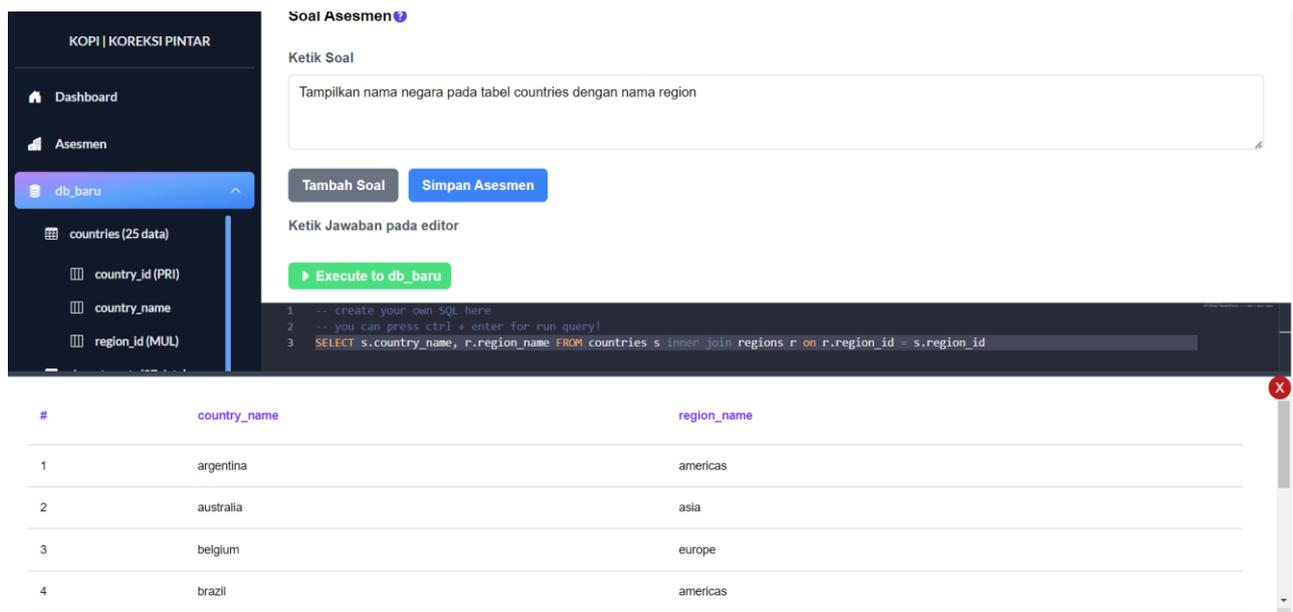
Gambar 9. Pembuatan Asesmen

Pada Gambar 12, dosen dapat membuat soal asesmen pada input soal dan menulis kunci jawaban *query* pada sebuah editor seperti gambar 13. Editor online yang diterapkan pada sistem ini menggunakan *Monaco Editor* yang merupakan mesin kode editor dari visual studio code [18]. Editor tersebut dapat memudahkan dalam penulisan kode query SQL seperti pada Gambar 13.



Gambar 10. Pengetikan Soal dan Kunci Jawaban SQL di Editor

Setelah proses penulisan query, pengguna dapat melihat hasil query dengan menekan tombol execute atau menekan shortcut ctrl+enter pada editor dan sistem akan menjalankan query pada database server dengan menekan tombol berwarna hijau. Lalu sistem akan otomatis menjalankan query ke database server lalu sistem akan menampilkan hasil query berupa data atau pesan error terhadap query seperti Gambar 14.



Gambar 11. Hasil Run Query

Proses run query pada Gambar 14 dapat dilakukan oleh dosen saat pembuatan dan koreksi asesmen dan pengguna mahasiswa pada saat mengerjakan asesmen.

3) Koreksi Asesmen

Koreksi asesmen dapat dilakukan oleh seorang dosen terhadap asesmen yang telah dibuat dan telah dikerjakan oleh mahasiswa. Proses koreksi asesmen menerapkan metode Cosine Similarity untuk menghasilkan persentase kemiripan antara kunci jawaban dan jawaban mahasiswa.

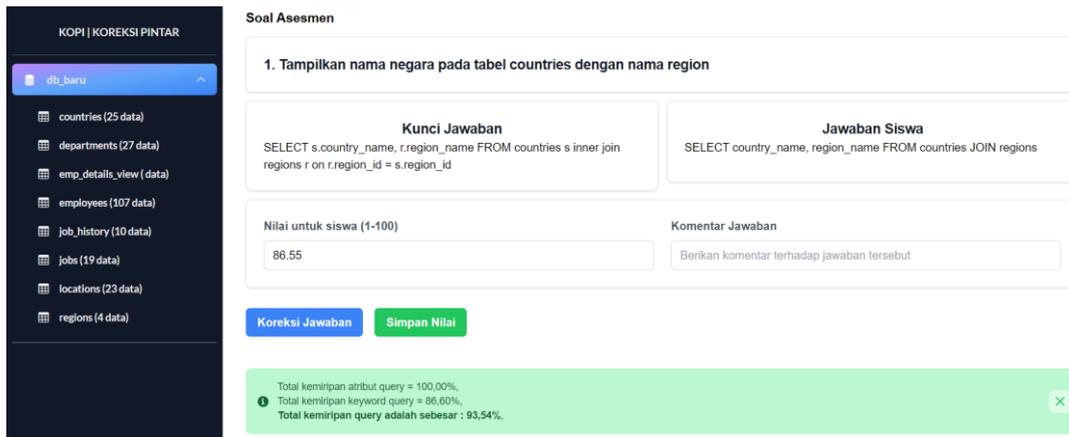
Pra proses penerapan metode harus menganalisa query dengan SQL parser. Proses parser query SQL akan memecah query mengetahui maksud terhadap query seperti Gambar 15.

```
SELECT [column-1, ... , column-n],
FROM [table-1, ... , table -n],
WHERE [condition-1, ..., condition-n]
```

Gambar 12. Parser SQL Statement

Hasil SQL parser membentuk susunan kata dan selanjutnya diproses untuk text folding atau pengubahan seluruh

karakter menjadi huruf kecil dan dilakukan pembobotan seperti Tabel 2. Halaman koreksi asesmen seperti Gambar 16.



Gambar 13. Koreksi Asesmen

B. Penerapan Metode

Penerapan tools SQL Parser dan metode Cosine Similarity terdapat pada halaman koreksi asesmen. SQL Parser digunakan untuk pra proses data sebelum menghitung persentase kemiripan. SQL parser dapat memudahkan dalam menyeleksi dan memilah antara klausa atau atribut tabel dari query dan keyword atau statement query. Cosine Similarity menjadi bagian proses perhitungan tingkat kemiripan kedua kalimat hasil dari hasil SQL Parser. Berikut ini contoh studi kasus untuk proses perhitungan metode.

TABEL IV
 CONTOH SOAL, KUNCI JAWABAN DAN JAWABAN MAHASISWA

Soal	Tampilkan nama negara pada tabel countries dengan nama region
Kunci Jawaban	<i>SELECT s.country_name, r.region_name FROM countries s inner join regions r on r.region_id = s.region_id</i>
Jawaban Mahasiswa 1	<i>SELECT * FROM countries JOIN regions</i>
Jawaban Mahasiswa 2	<i>SELECT country_name, region_name FROM countries JOIN regions</i>
Jawaban Mahasiswa 3	<i>select region_name, country_name from countries join regions</i>

Terdapat beberapa model jawaban mahasiswa pada Tabel IV. Proses perhitungan penerapan metode Cosine Similarity dipilih pada salah satu jawaban mahasiswa, dalam perhitungan diambil data Jawaban Mahasiswa 2 dari Tabel IV. Proses pertama yakni inialisasi terlebih dahulu dan dilakukan *text folding* untuk mengubah karakter menjadi huruf kecil.

TABEL V
 INISIALISASI DAN TEXT FOLDING

Inisialisasi	Query
Kunci Jawaban	<i>select s.country_name, r.region_name from countries s inner join regions r on r.region_id = s.region_id</i>
Jawaban Mahasiswa 2	<i>select country_name, region_name from countries join regions</i>

Dari hasil query pada Tabel V dapat dilanjutkan pada proses SQL parser untuk masing-masing query. Proses parser atau pemecahan query dilakukan seperti Tabel VI.

TABEL VI
 TOKENISASI UNTUK PARSING QUERY SQL

Inisialisasi	Query
Kunci Jawaban	<i>[select, s.country_name, r.region_name, from, countries, s, inner, join, regions, r, on, r.region_id, =,s.region_id]</i>
Jawaban Mahasiswa2	<i>[select, country_name, region_name, from, countries, join, regions]</i>

SQL parser akan melakukan split atau pemecahan seperti Tabel VI dan dilakukan tokenisasi setiap statement SQL untuk menyimpan informasi nama kolom, dan tabel. Sehingga hasil dari SQL parser akan membentuk sebuah

struktur TREE [19] seperti pada Gambar 5. Pada penelitian ini memfilter informasi dari kategori sintaks atau statement SQL, atribut atau nama kolom dan tabel, dan keseluruhan query yang terdapat pada Tabel VII dan Tabel VIII.

TABEL VII
 HASIL PARSER KUNCI JAWABAN

Sintaks SQL	<i>select,from,inner,join</i>
Atribut SQL	<i>country_name,region_name,countries,regions</i>
Keseluruhan	<i>select,from,inner,join,country_name,region_name,countries,regions</i>

TABEL VIII
 HASIL PARSER JAWABAN MAHASISWA

Sintaks SQL	<i>select,from,join</i>
Atribut SQL	<i>country_name,region_name,countries,regions</i>
Keseluruhan	<i>select,from,join,country_name,region_name,countries,regions</i>

Pada Tabel VII dan VIII merupakan hasil parser terhadap query SQL. Pada penelitian ini membagi menjadi tiga kategori yakni hasil sintaks SQL, atribut dan gabungan keduanya. Setiap kategori akan dilakukan proses perhitungan kemiripan jawaban. Hal ini dilakukan karena untuk menjadikan pembandingan untuk pengguna dengan *role* dosen agar dapat memberikan penilaian. Dari hasil ini akan diketahui pemahaman mahasiswa terhadap pembelajaran SQL.

Setelah proses parser, ketiga kategori tersebut dilakukan proses pembobotan setiap kata atau term frequency. Pada proses ini akan dipilih kata unik dari setiap kategori seperti pada Tabel IX hingga XI.

TABEL IX
 KATA UNIK KATEGORI SINTAKS QUERY

Kunci Jawaban	<i>select,from,inner,join</i>
Jawaban	<i>select,from,join</i>
Gabungan	<i>select,from,inner,join</i>

TABEL X
 KATA UNIK KATEGORI ATRIBUT ATAU NAMA KOLOM QUERY

Kunci Jawaban	<i>country_name,region_name,countries,regions</i>
Jawaban	<i>country_name,region_name,countries,regions</i>
Gabungan	<i>country_name,region_name,countries,regions</i>

TABEL XI
 KATA UNIK KATEGORI KESELURUHAN QUERY

Kunci Jawaban	<i>select,from,inner,join,country_name,region_name,countries,regions</i>
Jawaban	<i>select,from,join,country_name,region_name,countries,regions</i>
Gabungan	<i>select,from,inner,join,country_name,region_name,countries,regions</i>

Setelah proses pemilihan kata unik untuk setiap kategori, maka dapat dilakukan proses pembobotan dengan *term frequency*. Hal ini untuk memberikan nilai vektor terhadap frekuensi kemunculan setiap kata dengan nilai 0 atau 1 seperti pada Tabel XII hingga XIV.

TABEL XII
 PEMBOBOTAN KATEGORI SINTAKS SQL

No	Term	T(A)	T(B)
1.	select	1	1
2.	from	1	1
3.	inner	1	0
4.	join	1	1

TABEL XIII
 PEMBOBOTAN KATEGORI ATRIBUT KOLOM SQL

No	Term	T(A)	T(B)
1.	country_name	1	1
2.	region_name	1	1
3.	countries	1	1
4.	regions	1	1

TABEL XIV
 PEMBOBOTAN KATEGORI KESELURUHAN SQL

No	Term	T(A)	T(B)
1.	select	1	1
2.	from	1	1
3.	inner	1	0
4.	join	1	1
5.	country_name	1	1
6.	region_name	1	1
7.	countries	1	1
8.	regions	1	1

Keterangan:
 T(A) = Kunci Jawaban
 T(B) = Jawaban Mahasiswa 2
 Term = Kata

Pada Tabel XII hingga XIV merupakan hasil pembobotan setiap kata dari query SQL. T(A) merupakan query pertama atau query kunci jawaban dan T(B) merupakan query jawaban mahasiswa. Terdapat perbedaan bobot yang mana pada jawaban mahasiswa tidak terdapat kata 'inner' sehingga bobot pada kata tersebut pada kolom T(B) adalah 0 yang berarti tidak terdapat kata tersebut.

Setelah proses pembobotan, hasil term frequency dapat dilanjutkan pada untuk perhitungan metode Cosine Similarity. Berikut ini perhitungan Cosine Similarity dari persamaan 1.

Perhitungan Cosine kategori sintaks SQL sebagai berikut.

$$\cos = \frac{(1 \times 1) + (1 \times 1) + (1 \times 0) + (1 \times 1)}{\sqrt{4} \times \sqrt{3}}$$

$$\cos = \frac{3}{2 \times 1.7320508075689}$$

$$\cos = 0.86602540378444 \times 100\%$$

$$\cos = 86,60 \%$$

Hasil perhitungan untuk kategori sintaks SQL yakni sebesar 86% atau nilai *cosine*-nya adalah 0,86. Perhitungan cosine kategori atribut kolom SQL sebagai berikut

$$\cos = \frac{(1 \times 1) + (1 \times 1) + (1 \times 1) + (1 \times 1)}{\sqrt{4} \times \sqrt{4}}$$

$$\cos = \frac{4}{2 \times 2}$$

$$\cos = 1 \times 100\%$$

$$\cos = 100 \%$$

Hasil perhitungan untuk kategori atribut SQL adalah 100%. Hal ini karena kedua query memiliki makna atribut atau tujuan kolom tabel yang sama pada database. Perhitungan *cosine* kategori keseluruhan query yakni penggabungan kategori atribut dan sintaks SQL sebagai berikut.

$$\cos = \frac{(1 \times 1) + (1 \times 1) + (1 \times 0) + (1 \times 1) + (1 \times 1) + (1 \times 1) + (1 \times 1) + (1 \times 1)}{\sqrt{8} \times \sqrt{7}}$$

$$\cos = \frac{7}{2.8284271247462 \times 2.6457513110646}$$

$$\cos = 0.93541434669349 \times 100\%$$

$$\cos = 93,54\%.$$

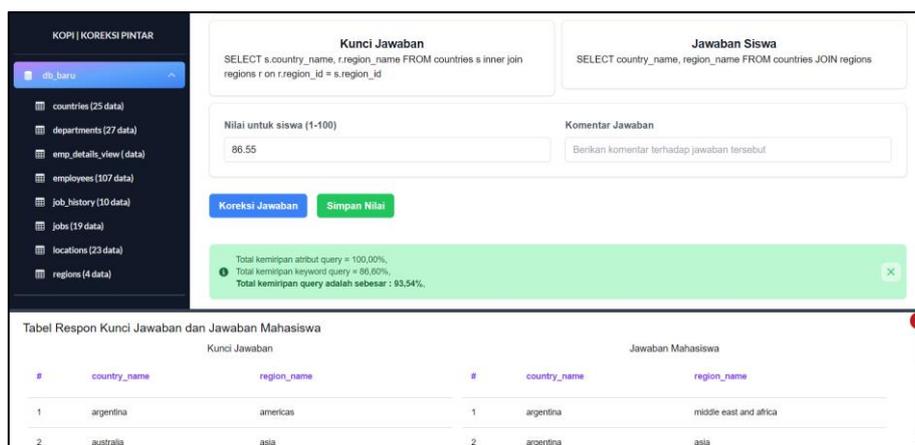
Pada query kunci jawaban dan jawaban mahasiswa 2 dari studi kasus di atas adalah 86,60% untuk keyword atau sintaks SQL, 100% untuk atribut SQL atau nama kolom dan 93,54% untuk keseluruhan query. Perhitungan dilakukan pada setiap jawaban mahasiswa dengan hasil perhitungan pada sistem dan hasil input penilaian dosen pengampu basis data tertera pada Tabel XV.

TABEL XV
 HASIL PERHITUNGAN COSINE SIMILAIRTY DAN PENILAIAN PADA SISTEM

No	Inisial	Cosine (%)	Penilaian (%)
1.	Jawaban Mahasiswa 1	72,17	62,50
2.	Jawaban Mahasiswa 2	93,54	87,50
3.	Jawaban Mahasiswa 3	93,54	87,50

Hasil penilaian pada Tabel XV merupakan nilai akhir terhadap jawaban mahasiswa. Dosen pengampu Basis Data menginputkan nilai terhadap masing-masing jawaban. Setelah hasil proses perhitungan kemiripan, sistem asesmen akan menampilkan output data terhadap kedua query SQL. Sistem secara langsung akan menampilkan data yang

dihasilkan terhadap query tersebut seperti pada Gambar 17.



The screenshot displays the 'KOREKSI PINTAR' interface. On the left is a sidebar with a list of database tables: 'db_basis', 'countries (25 data)', 'departments (27 data)', 'emp_details_view (data)', 'employees (107 data)', 'job_history (10 data)', 'jobs (19 data)', 'locations (23 data)', and 'regions (4 data)'. The main area is divided into sections: 'Kunci Jawaban' (Key Answer) showing a SQL query: 'SELECT s.country_name, r.region_name FROM countries s inner join regions r on r.region_id = s.region_id'; 'Jawaban Siswa' (Student Answer) showing a SQL query: 'SELECT country_name, region_name FROM countries JOIN regions'; 'Nilai untuk siswa (1-100)' (Score for student) with a value of 86.55; and 'Komentar Jawaban' (Answer Comment) with a text input field. Below these are buttons for 'Koreksi Jawaban' and 'Simpan Nilai'. A green notification box shows similarity statistics: 'Total kemiripan atribut query = 100.00%', 'Total kemiripan keyword query = 86.60%', and 'Total kemiripan query adalah sebesar : 93.54%'. At the bottom, a table titled 'Tabel Respon Kunci Jawaban dan Jawaban Mahasiswa' compares the key answer and student answer.

Kunci Jawaban		Jawaban Mahasiswa	
#	country_name	#	country_name
1	argentina	1	argentina
2	australia	2	argentina

Gambar 14. Hasil Output Query dan Cosine Similarity

Pada Gambar 17 merupakan tampilan sistem terhadap output kedua query. Sehingga dosen sebagai penilai dapat melihat data yang tampil apakah sudah sesuai dengan kunci jawaban atau tidak. Dosen dapat memberikan penilaian tersendiri untuk setiap jawaban mahasiswa. Penggunaan perhitungan persentase kemiripan sebagai alat bantu dosen dalam menentukan nilai lebih objektif terhadap jawaban mahasiswa. Serta sebagai asisten untuk mengoreksi jawaban tersebut.

Penelitian ini mengembangkan sistem asesmen berbasis web yang menggunakan SQL Parser dan algoritma *Cosine Similarity* untuk menilai jawaban query SQL secara otomatis. Kinerja dan efisiensi sistem dalam menangani query secara simultan serta dalam mengolah data besar adalah aspek penting yang perlu dibahas. Sistem ini dirancang untuk menangani banyak query secara simultan, yang sering terjadi dalam lingkungan pendidikan di mana banyak siswa mengirimkan jawaban mereka pada waktu yang bersamaan. SQL Parser bekerja dengan memecah setiap query yang diterima menjadi token dan membangun struktur pohon sintaksis (parse tree) untuk analisis lebih lanjut [20]. Proses ini dilakukan secara cepat dan efisien, karena tokenisasi dan pembuatan parse tree adalah operasi yang relatif ringan dan dapat diproses dalam waktu singkat.

Algoritma *Cosine Similarity*, yang digunakan untuk menghitung kemiripan antara query siswa dan kunci jawaban, juga dirancang untuk efisiensi. Setelah query diubah menjadi vektor token, perhitungan *Cosine Similarity* dilakukan menggunakan operasi matematika sederhana yang dapat dieksekusi dengan cepat, bahkan untuk sejumlah besar query. Sistem ini dapat memanfaatkan teknik parallel processing untuk memproses beberapa query secara bersamaan, sehingga meningkatkan *throughput* dan mengurangi waktu tunggu.

Namun, seperti halnya dengan sistem komputasi lainnya, ada beberapa kendala kinerja yang mungkin muncul saat mengolah data dalam jumlah besar. Salah satu kendala utama adalah konsumsi memori. SQL Parser harus menyimpan struktur parse tree untuk setiap query yang diproses, dan jika jumlah query sangat besar, penggunaan memori dapat meningkat secara signifikan. Untuk mengatasi hal ini, sistem dapat menerapkan mekanisme manajemen memori yang efisien, seperti penghapusan struktur parse tree yang tidak lagi diperlukan setelah analisis selesai.

Kendala lain adalah waktu pemrosesan yang dibutuhkan untuk vektorisasi dan perhitungan *Cosine Similarity* pada volume data yang besar. Meskipun perhitungan *Cosine Similarity* relatif cepat, jika jumlah query sangat besar, waktu total yang dibutuhkan untuk memproses semua query dapat meningkat. Solusi untuk masalah ini meliputi penggunaan algoritma optimasi dan teknik komputasi terdistribusi, yang dapat membagi beban pemrosesan ke beberapa mesin atau prosesor. Selain itu, sistem dapat mengalami penurunan kinerja jika terdapat query yang sangat kompleks atau panjang, karena analisis sintaksis dan semantik akan memakan lebih banyak waktu. Untuk mitigasi, sistem dapat dioptimalkan dengan menggunakan algoritma parsing yang lebih efisien atau dengan membatasi kompleksitas query yang dapat diterima oleh sistem.

Metode *Cosine Similarity* memiliki kelebihan dalam hal kesederhanaan, efisiensi, dan kecepatan, membuatnya cocok untuk implementasi awal dan penggunaan luas. Namun, ia memiliki keterbatasan dalam menangani struktur semantik dan variasi ekspresi SQL. Untuk analisis yang lebih mendalam dan akurat, terutama dalam konteks pendidikan di mana kesamaan semantik dan logika query sangat penting, metode alternatif seperti *Tree-based Similarity* atau *Graph Matching* dapat dipertimbangkan meskipun dengan biaya komputasi yang lebih tinggi. Menggabungkan *Cosine Similarity* dengan metode lain mungkin juga menjadi solusi yang efektif untuk mengatasi keterbatasannya.

C. Pengujian Sistem

Pengujian sistem dilakukan dengan perhitungan nilai error pada nilai input dosen pengampu Basis Data dan nilai hasil *cosine similarity*. Kedua nilai tersebut dilakukan uji validitas dengan menghitung nilai MAE dan RMSE dari data pada tabel XV dengan perhitungan berikut.

$$MAE = \frac{(0,7217 - 0,6250) + (0,9354 - 0,8750) + (0,9354 - 0,8750)}{3}$$

$$MAE = \frac{(0,0967) + (0,0604) + (0,0604)}{3} = \frac{0,2175}{3} = \mathbf{0,0725}$$

$$RMSE = \sqrt{\frac{(0,7217 - 0,6250)^2 + (0,9354 - 0,8750)^2 + (0,9354 - 0,8750)^2}{3}}$$

$$RMSE = \sqrt{\frac{0,00935089 + 0,00364816 + 0,00364816}{3}} = \sqrt{\frac{0,01664721}{3}}$$

$$RMSE = \mathbf{0,07449208}$$

Pengujian juga dilakukan pada pengukuran kecepatan sistem dalam memproses query SQL dengan SQL Parser dan perhitungan cosine similarity. Pada sistem ini memberikan informasi kecepatan pemrosesan pada Tabel XVI.

TABEL XVI
 HASIL EKSKUSI PENILAIAN PADA SISTEM

No	Inisial	Time Execute (detik)
1.	Jawaban Mahasiswa 1	0,0504
2.	Jawaban Mahasiswa 2	0.0510
3.	Jawaban Mahasiswa 3	0.0741

Pada Tabel XVI merupakan waktu yang dihasilkan oleh sistem dalam memproses algoritma SQL Parser dan perhitungan Cosine Similarity pada setiap soal. Rata-rata waktu eksekusi sistem adalah sebesar 0,0585 detik. Pada kasus ini sistem dapat menangani hasil kemiripan query dan telah menampilkan respon data dari database server. Data yang dihasilkan sesuai dengan perintah query yang telah dikirim saat proses pengoreksian jawaban.

Pengujian sistem juga dilakukan menggunakan pengujian Black Box untuk pengujian fungsionalitas sistem. Pengujian ini melibatkan survei mahasiswa dan dosen pengampu Basis Data untuk uji coba asesmen. Uji coba meliputi halaman: 1) login, 2) register, 3) membuat kelas bagi dosen, 4) mendaftar kelas bagi mahasiswa, 5) membuat database, 6) import file SQL, 7) Kelola asesmen, 8) Mengikuti Asesmen, 9) Koreksi, 10) Run Query, dan 11) Logout, Berikut hasil pengujian fungsionalitas dari sistem asesmen pada Tabel XVII.

TABEL XVII
 HASIL PENGUJIAN BLACK BOX

No	Fitur	Aksi	Output	Pengguna	Hasil
1	Login	Memasukkan npm/npt dan password.	Sistem menerima dan masuk ke halaman dashboard.	Dosen & Mahasiswa	Berhasil
2	Register	Memasukkan data register.	Data register tersimpan, menuju ke dashboard.	Dosen & Mahasiswa	Berhasil
3	Pembuatan Kelas	Menekan tombol tambah kelas lalu, Memasukkan data kelas baru.	Beralih ke halaman tambah kelas Menyimpan data kelas dan kembali ke halaman daftar kelas.	Dosen	Berhasil
4	Mendaftar Kelas	Memilih kelas, memasukkan kode kelas.	Memvalidasi kode dan menampilkan daftar kelas diambil	Mahasiswa	Berhasil
5	Membuat Database	Menekan tombol tambah, mengisi data nama database dan akses database	Beralih ke halaman tambah database, menyimpan data hasil. Membuat database baru pada sistem.	Dosen	Berhasil
6	Import file SQL	Memilih database tersedia, mengirim file SQL.	Eksekusi file ke database yang dipilih.	Dosen	Berhasil
7	Kelola Asesmen	Menekan tombol tambah asesmen, Membuat agenda ujian, membuat soal dan kunci jawaban.	Beralih halaman, menyimpan data agenda ujian, soal dan kunci jawaban.	Dosen	Berhasil
8	Mengikuti Asesmen	Memilih menu ujian/asesmen, mengisi soal hingga selesai.	Menyimpan jawaban ujian.	Mahasiswa	Berhasil

9	Koreksi Asesmen	Memilih agenda ujian, memilih mahasiswa dan soal yang akan dikoreksi.	Menampilkan data ujian, data jawaban, dan kunci jawaban, lalu menghitung tingkat kemiripan jawaban mahasiswa dan kunci jawaban	Dosen	Berhasil
10	Run Query SQL	Pada halaman Kelola ujian dan koreksi ujian. Menekan run query.	Mengirim string query SQL dari editor, lalu sistem mengeksekusi query, dan menampilkan data hasil query.	Dosen & Mahasiswa	Berhasil
11	Sign out	Menekan tombol sign out.	Keluar sistem.	Dosen & Mahasiswa	Berhasil

Pada Tabel XV merupakan hasil pengujian fungsionalitas pada keseluruhan sistem yang telah dibangun, Dari hasil validasi pengujian dapat dipastikan sistem berjalan dengan baik dan dapat digunakan sebagai sistem pembelajaran SQL.

Sistem asesmen berbasis web yang menggunakan SQL Parser dan algoritma Cosine Similarity ini memiliki potensi besar untuk dikembangkan lebih lanjut guna meningkatkan fungsionalitas dan fleksibilitasnya. Salah satu arah pengembangan adalah penyempurnaan analisis semantik, yang meliputi pengenalan sinonim dan ekspresi alternatif, serta deteksi kesalahan logika dalam query. Ini akan membantu sistem untuk lebih memahami variasi penulisan query SQL yang berbeda namun memiliki makna yang sama.

Selain itu, fitur penilaian berbasis rubrik bisa ditambahkan untuk memungkinkan dosen membuat rubrik penilaian yang lebih terstruktur dan detail, termasuk aspek-aspek seperti efisiensi query dan performa eksekusi, menyediakan mode pembelajaran dan latihan otomatis dengan tingkat kesulitan yang bervariasi, menambahkan dukungan untuk query NoSQL dan query SPARQL akan memperluas kegunaan sistem dalam berbagai jenis basis data, menyediakan API untuk ekstensi juga akan memungkinkan pengembang pihak ketiga menambah fungsionalitas sistem, seperti modul penilaian untuk proyek akhir atau tugas besar. Dengan penambahan fitur-fitur tersebut, sistem asesmen ini dapat menjadi alat yang sangat berharga dalam pendidikan komputer dan pemrograman, memberikan manfaat tambahan baik bagi pengajar maupun mahasiswa.

Penelitian ini mengembangkan sistem asesmen berbasis web menggunakan SQL Parser dan algoritma *Cosine Similarity* untuk penilaian query SQL. Untuk memberikan konteks lebih lanjut dan menunjukkan kontribusi penelitian ini, berikut adalah perbandingan dengan hasil penelitian terkait yang telah dilakukan sebelumnya.

Bastian [5] mengembangkan sistem penilaian jawaban esai menggunakan metode Cosine Similarity dan algoritma Nazief dan Adriani. Penelitian tersebut menunjukkan bahwa penggunaan Cosine Similarity dapat mengurangi waktu yang dibutuhkan oleh dosen dalam mengoreksi dan memberikan nilai terhadap jawaban mahasiswa. Sistem ini berhasil dalam konteks penilaian teks esai, namun tidak secara khusus dirancang untuk query SQL. Al-Salmi [11] mengembangkan SQL Formulation Editor (SQL-FE) untuk membantu mahasiswa dalam merumuskan query SQL dengan tepat dan meningkatkan kinerja mereka dalam memahami SQL. Sistem ini menggunakan antarmuka yang intuitif dan memberikan umpan balik langsung kepada pengguna, namun tidak mencakup fitur penilaian otomatis berdasarkan kemiripan query dengan kunci jawaban.

Penelitian ini memperluas penggunaan Cosine Similarity dari teks esai ke penilaian query SQL, memberikan kontribusi unik dalam konteks pembelajaran basis data. Tidak seperti penelitian Bastian yang berfokus pada teks esai, penelitian ini mengintegrasikan SQL Parser untuk menganalisis sintaksis dan semantik query SQL. SQL Parser memungkinkan pemecahan query menjadi token-token yang relevan dan membentuk struktur pohon, yang memberikan analisis lebih mendalam dibandingkan hanya menggunakan metode berbasis teks.

Selain itu, penelitian ini menambahkan fitur penilaian otomatis yang tidak hanya menghemat waktu dosen tetapi juga memberikan umpan balik instan kepada mahasiswa. Ini sejalan dengan temuan Al-Salmi tentang pentingnya umpan balik langsung dalam pembelajaran SQL, namun dengan penekanan tambahan pada otomatisasi penilaian.

Dibandingkan dengan penelitian Aditya, yang menunjukkan keunggulan Cosine Similarity dalam sistem rekomendasi, penelitian ini menunjukkan aplikasi praktis Cosine Similarity dalam konteks pendidikan dan penilaian query SQL. Hasil penelitian ini mendukung temuan bahwa Cosine Similarity adalah metode yang efektif untuk mengukur kemiripan, namun dalam konteks yang lebih spesifik dan teknis yaitu penilaian query SQL.

IV. KESIMPULAN

Berdasarkan penelitian di atas, sistem asesmen pembelajaran SQL berbasis website menggunakan SQL Parser dan algoritma Cosine Similarity terbukti efektif dalam menilai jawaban query SQL secara otomatis. Sistem ini mampu menangani banyak query secara simultan dengan efisiensi tinggi, memberikan umpan balik instan kepada pengguna baik dosen dan mahasiswa, dan menghemat waktu dosen dalam proses penilaian. Hasil pengujian

menunjukkan bahwa sistem berfungsi dengan baik dalam berbagai fitur dan skenario penggunaan, serta memiliki potensi besar untuk dikembangkan lebih lanjut guna meningkatkan fungsionalitas dan fleksibilitasnya dalam konteks pendidikan komputer dan pemrograman.

DAFTAR PUSTAKA

- [1] D. A. Rismayadi, J. F. Rusdi, A. Prinayanti, P. D. Akbar, and R. D. Andriani, "Sistem Informasi Akademik, Keuangan Dan Ujian Online Berbasis Website Dan Android (Studi Kasus Smk Negeri 2 Cimahi)," *SENSITIf Semin. Nas. Sist. Inf. dan Teknol. Inf.*, pp. 1379–1390, 2019, [Online]. Available: <http://ejurnal.diponegara.ac.id/index.php/sensitif/article/view/582>
- [2] M. D. Payana and M. B. Wibawa, "Sistem Ujian Online Tes Masuk Universitas Ubudiyah Indonesia Bagi Calon Mahasiswa Baru Berbasis Web Menggunakan CI (Code Igniter)," *J. Informatics Comput. Sci.*, vol. 6, no. 2, pp. 91–103, 2020, [Online]. Available: <https://doi.org/http://jurnal.uui.ac.id/index.php/jics/article/view/1248/64%0A7>
- [3] F. Fataruba, "Penerapan Metode Cosine Similarity Untuk Pengecekan Kemiripan Jawaban Ujian Siswa," *J. Mhs. Tek. Inform.*, vol. 2, no. 2, pp. 88–95, 2018.
- [4] D. Puspitasari, P. P. Arhandi, P. Y. Saputra, Y. W. Syaifudin, H. A. Himawan, and P. A. Sholihah, "Online judge mysql for learning process of database practice course," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 523, no. 1, 2019, doi: 10.1088/1757-899X/523/1/012046.
- [5] A. Bastian, H. Sujadi, and P. A. Sukmana, "Rancang Bangun Aplikasi Penilaian Ujian Essay Dengan Menggunakan Algoritma Nazief & Andriani Dan Metode Cosine Similarity," pp. 62–68, 2018.
- [6] R. Y. MF, *Pemrograman SQL dan PL/pgSQL di Database PostgreSQL*. in Pemrograman Database. Native Enterprise, 2020. [Online]. Available: <https://books.google.co.id/books?id=14HgDwAAQBAJ>
- [7] F. A. Akbar, F. Muttaqin, and E. P. Mandyartha, "An approach for refactoring in model layer on MVC based web application," *Proceeding - 6th Inf. Technol. Int. Semin. ITIS 2020*, pp. 178–182, 2020, doi: 10.1109/ITIS50118.2020.9320998.
- [8] Phpmyadmin, "phpmyadmin_sql-parser_ A validating SQL lexer and parser with a focus on MySQL dialect." Accessed: Mar. 06, 2024. [Online]. Available: <https://github.com/phpmyadmin/sql-parser>
- [9] J. Sarazyn, "SQL QUERY PARSING ANDTRANSLATION," vol. 1, no. 19, p. 33, 2014, [Online]. Available: <https://patents.google.com/patent/US20140244680A1/en>
- [10] R. Williams and D. Garcia, "Codekey - An online code editor to study code patterns and enhance student performance in cs courses," *SIGCSE 2020 - Proc. 51st ACM Tech. Symp. Comput. Sci. Educ.*, p. 1357, 2020, doi: 10.1145/3328778.3372680.
- [11] A. AL-Salmi, "A web-based semi-automatic assessment tool for formulating basic SQL statements: Point-and-click interaction method," *CSEDU 2018 - Proc. 10th Int. Conf. Comput. Support. Educ.*, vol. 1, no. Csedu 2018, pp. 191–198, 2018, doi: 10.5220/0006671501910198.
- [12] M. Mustamiin, E. Ismantohadi, A. L. Ghozali, Darsih, and L. N. Inara, "Rancangan Bangun Sistem Manajemen Soal Dan Ujian Berbasis Website Menggunakan Framework Laravel," *J. IKRA-ITH Inform.*, vol. 4, no. 1, pp. 58–63, 2020.
- [13] S. V. Putratama, *Pemrograman Web dengan Menggunakan PHP dan Framework Codeigniter*. Deepublish, 2018. [Online]. Available: <https://books.google.co.id/books?id=7SIIDwAAQBAJ>
- [14] C. L. Phang, *Mastering Front-End Web Development (HTML, Bootstrap, CSS, SEO, Cordova, SVG, ECMAScript, JavaScript, WebGL, Web Design and many more.): 14 Books in 1. Introducing 200+ Extensions. An Advanced Guide*. Amazon Digital Services LLC - Kdp, 2020. [Online]. Available: <https://books.google.co.id/books?id=Y-UJEAAAQBAJ>
- [15] Fawwaz Ali Akbar, Eka Prakarsa Mandyartha, and Hendra Maulana, "An Approach for Automatic Generating RESTful API Code based on SQL DDL Code," *Tech. Rom. J. Appl. Sci. Technol.*, vol. 16, no. 3, pp. 118–123, 2023, doi: 10.47577/technium.v16i.9969.
- [16] A. D. B. A. Mutlu, and P. Karagoz, "Cosine Similarity-Based Pruning," vol. 1, pp. 90–96, 2016, doi: 10.1007/978-3-319-47217-1.
- [17] M. A. Ma'ruf and A. Qoiriah, "Perbandingan Algoritma Cosine Similarity dan Euclidean Distance pada Sistem Rekomendasi Film dengan Metode Item-Based Collaborative Filtering," *J. Informatics Comput. Sci.*, vol. 04, pp. 160–168, 2022, doi: 10.26740/jinacs.v4n02.p160-168.
- [18] Microsoft, "Monaco Editor - The Editor of the Web," Microsoft - Monaco Editor. Accessed: Jan. 17, 2024. [Online]. Available: <https://microsoft.github.io/monaco-editor/>
- [19] S. Yang, Z. Wei, G. L. Herman, and A. Alawini, "Analyzing Patterns in Student SQL Solutions via Levenshtein Edit Distance," *L@S 2021 - Proc. 8th ACM Conf. Learn. @ Scale*, pp. 323–326, 2021, doi: 10.1145/3430895.3460979.
- [20] M. V. Sadaphule, "SQL Query Parser : An Automated Tool for Translating the Queries Into Spreadsheets," vol. 14, no. 8, pp. 23–28, 2016.