

**RANCANG BANGUN SISTEM OPTIMASI KINERJA DATA
DEFINITION LANGUAGE (DDL) MYSQL BERBASIS API GEMINI**

TUGAS AKHIR

Untuk Memenuhi Sebagian Persyaratan Mencapai Derajat Sarjana S-1
Program Studi Informatika



Disusun oleh:

MUHAMMAD ALFARIZI HABIBULLAH

22106050049

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
YOGYAKARTA**

2026

HALAMAN PENGESAHAN



KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN KALIJAGA
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Marsda Adisucipto Telp. (0274) 540971 Fax. (0274) 519739 Yogyakarta 55281

PENGESAHAN TUGAS AKHIR

Nomor : B-1418/Un.02/DST/PP.00.9/07/2026

Tugas Akhir dengan judul : RANCANG BANGUN SISTEM OPTIMASI KINERJA DATA DEFINITION LANGUAGE (DDL) MYSQL BERBASIS API GEMINI

yang dipersiapkan dan disusun oleh:

Nama : MUHAMMAD ALFARIZI HABIBULLAH
Nomor Induk Mahasiswa : 22106050049
Telah diujikan pada : Jumat, 05 Juni 2026
Nilai ujian Tugas Akhir : A-

dinyatakan telah diterima oleh Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta

TIM UJIAN TUGAS AKHIR



Ketua Sidang
Nurochman, S.Kom., M.Kom.
SIGNED

Valid ID: 6a46171977a4



Penguji I
Dr. Ir. Sumarsono, S.T., M.Kom.
SIGNED

Valid ID: 6a45e3ee947e



Penguji II
Dr. Fitri Wulandari, S.Si., M.Kom.
SIGNED

Valid ID: 6a45dc4eedc5



Yogyakarta, 05 Juni 2026
UIN Sunan Kalijaga
Dekan Fakultas Sains dan Teknologi
Prof. Dr. Dra. Hj. Khurul Wardati, M.Si.
SIGNED

Valid ID: 6a4744ce5524d

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Alfarizi Habibullah

NIM : 22106050049

Program Studi : Informatika

Fakultas : Sains dan Teknologi

Menyatakan dengan sesungguhnya, bahwa skripsi saya yang berjudul **“RANCANG BANGUN SISTEM OPTIMASI KINERJA DATA DEFINITION LANGUAGE (DDL) MYSQL BERBASIS API GEMINI”** merupakan penelitian saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi dan bukan plagiasi karya orang lain kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Sebagai salah satu syarat untuk memperoleh gelar Sarjana Program Studi Informatika pada Fakultas Sains dan Teknologi, Universitas Islam Negeri Sunan Kalijaga.

Yogyakarta, Mei 2026

Yang membuat pernyataan,



Muhammad Alfarizi Habibullah
22106050049

SURAT PERSETUJUAN TUGAS AKHIR



Universitas Islam Negeri Sunan Kalijaga

SURAT PERSETUJUAN TESIS / TUGAS AKHIR

Hal : Persetujuan Tugas Akhir

Lamp : -

Kepada

Yth.

Dekan Fakultas Sains dan Teknologi

UIN Sunan Kalijaga

DI Yogyakarta

Assalamu'alaikum wr. wb.

Setelah membaca, meneliti, memberikan petunjuk dan mengoreksi serta mengadakan perbaikan seperlunya, maka saya selaku pembimbing berpendapat bahwa skripsi Saudari:

Nama : Muhammad Alfarizi Habibullah
NIM : 22106050049
Judul Skripsi : RANCANG BANGUN SISTEM OPTIMASI KINERJA DATA
DEFINITION LANGUAGE (DDL) MYSQL BERBASIS API
GEMINI

Sudah dapat diajukan kepada Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu dalam Program Studi Informatika.

Dengan ini kami mengharap agar skripsi / tugas akhir Saudara dapat segera dimunaqosyahkan. Atas perhatiannya saya ucapkan terima kasih.

Wassalamu'alaikum wr. wb.

Yogyakarta, 25 Mei 2026

Dosen Pembimbing,

Nurochman, S.Kom., M.Kom
NIP. 198012232009011007

LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR

Tugas Akhir ini tidak dipublikasikan, tetapi tersedia di perpustakaan dalam lingkungan UIN Sunan Kalijaga Yogyakarta, diperkenankan dipakai sebagai referensi kepustakaan, tetapi pengutipan harus seizin penyusun, dan harus menyebutkan sumbernya sesuai dengan kebiasaan ilmiah. Dokumen Tugas Akhir ini merupakan hak milik UIN Sunan Kalijaga Yogyakarta.



INTISARI

Perancangan struktur database yang kurang tepat dapat menimbulkan inefisiensi dalam pengembangan perangkat lunak, khususnya pada pemilihan tipe data, penggunaan index, penerapan constraint, dan relasi antar tabel. Inefisiensi struktur tersebut berpotensi memengaruhi performa akses data dan efisiensi penggunaan ruang penyimpanan (size). Penelitian ini bertujuan mengembangkan sistem berbasis web untuk menganalisis struktur Data Definition Language (DDL) MySQL, menghasilkan rekomendasi optimasi menggunakan AI generatif melalui API Gemini, serta memvalidasi SQL hasil optimasi melalui Docker Sandbox. Rekomendasi optimasi diarahkan pada perbaikan struktur DDL MySQL dengan mempertimbangkan potensi dampak terhadap performa dan size sebagai indikator bantu, bukan sebagai hasil benchmark performa aktual pada database produksi. Metode pengembangan yang digunakan adalah Extreme Programming (XP) dengan tahapan planning, design, coding, testing, dan release. Sistem memiliki fitur upload DDL MySQL, analisis rekomendasi, review suggestion, finalisasi SQL, validasi sandbox, koreksi otomatis berbasis AI generatif, dan download file SQL optimized. Hasil pengujian end-to-end menunjukkan alur utama sistem berhasil dengan status PASS, final status job FINALIZED, dan failed steps 0. Pengujian koreksi otomatis berbasis AI generatif juga berhasil dengan status PASS, final status job FINALIZED, was self corrected bernilai true, dan self correction count bernilai 1. Dengan demikian, sistem berhasil membantu proses analisis, rekomendasi, validasi, koreksi otomatis, dan finalisasi SQL hasil optimasi DDL MySQL pada lingkungan local/development.

Kata Kunci: DDL MySQL, AI Generatif, API Gemini, Docker Sandbox, Performa, Size

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

ABSTRACT

An inappropriate database structure may cause inefficiency in software development, particularly in data type selection, index usage, constraint implementation, and relationships between tables. Such structural inefficiency may potentially affect data access performance and storage space efficiency (size). This study aims to develop a web-based system to analyze MySQL Data Definition Language (DDL) structures, generate optimization recommendations using generative AI through the Gemini API, and validate the optimized SQL through Docker Sandbox. The optimization recommendations are directed toward improving MySQL DDL structures by considering the potential impact on performance and size as supporting indicators, not as the result of actual performance benchmarking on a production database. The development method used is Extreme Programming (XP), which consists of planning, design, coding, testing, and release stages. The system provides MySQL DDL upload, recommendation analysis, review suggestion, SQL finalization, sandbox validation, generative AI-based automatic correction, and optimized SQL file download features. The end-to-end testing result shows that the main system flow succeeded with PASS status, FINALIZED final job status, and zero failed steps. The generative AI-based automatic correction test also succeeded with PASS status, FINALIZED final job status, true self-correction status, and one self-correction attempt. Therefore, the system successfully supports the analysis, recommendation, validation, automatic correction, and finalization process of optimized MySQL DDL SQL in a local/development environment.

Keywords: MySQL DDL, Generative AI, Gemini API, Docker Sandbox, Performance, Size.

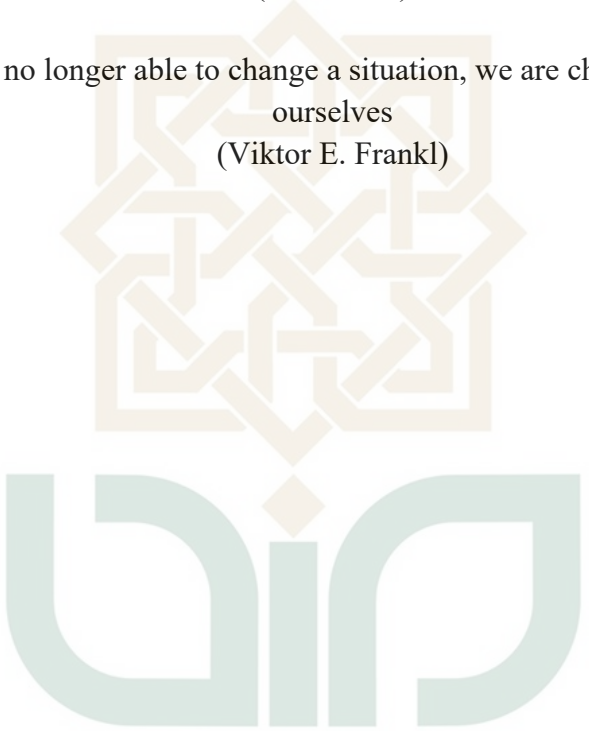
STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN MOTTO

وَأَنْ لَّيْسَ لِلْإِنْسَانِ إِلَّا مَا سَعَىٰ
(Q.S. An-Najm: 39)

“Be stubborn on vision, but flexible on details”
(Jeff Bezos)

When we are no longer able to change a situation, we are challenged to change
ourselves
(Viktor E. Frankl)



STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

HALAMAN PERSEMBAHAN

Dengan penuh rasa syukur dan terima kasih kepada Allah yang Maha Pengasih dan Maha Penyayang, saya persembahkan Tugas Akhir ini sebagai tanda penghormatan dan rasa cinta yang mendalam. Semoga karya ini dapat bermanfaat dan menjadi amal yang baik. Tugas Akhir ini saya persembahkan dengan penuh cinta dan hormat kepada:

1. Ibu, yang telah memberikan cinta dan kasih sayang, doa, dan dukungan yang tiada henti. sumber inspirasi dan motivasi dalam setiap langkah hidupku. Doa dan pengorbananmu adalah cahaya yang terus menyinari perjalanan hidupku.
2. Bapak, yang telah menjadi pilar penopang dalam hidupku. Dengan kebijaksanaan dan kesabaranmu, yang membimbingku melewati segala rintangan dan meraih keberhasilan. Doa dan pengorbananmu adalah cahaya yang terus menyinari perjalanan hidupku.
3. Kakak dan Adik, Muhammad Quraish Zulhamsyah dan Khairunisa Hernita Putri, yang selalu menjadi sumber semangat dan motivasi bagi saya dalam menyelesaikan tugas akhir ini.
4. Kepada diri sendiri yang telah bertahan dan berjuang melewati proses perjalanan ini. Terima kasih yang tak terhingga atas segala dukungan yang telah kalian berikan.

Semoga Allah senantiasa melimpahkan rahmat dan berkah-Nya, serta memberikan balasan terbaik untuk semua kebaikan yang telah diberikan.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

KATA PENGANTAR

Alhamdulillah, segala puja dan puji kepada Allah Yang Maha Pengasih lagi Maha Penyayang yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan Tugas Akhir ini. Selawat serta salam teruntuk Baginda Nabi Muhammad saw. semoga umatnya mendapatkan syafa'at beliau di yaumul akhir kelak. Amin Ya Rabbal 'Alamin.

Adapun tujuan utama dalam pembuatan laporan ini adalah sebagai Tugas Akhir dengan judul **“RANCANG BANGUN SISTEM OPTIMASI KINERJA DATA DEFINITION LANGUAGE (DDL) MYSQL BERBASIS API GEMINI”**.

Tugas akhir ini membutuhkan banyak dedikasi di setiap tahapnya. Keberhasilan ini tidak mungkin dicapai tanpa bantuan dan dukungan dari berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Prof. Noorhaidi Hasan S.Ag., M.A., M.Phil., Ph.D. selaku Rektor UIN Sunan Kalijaga Yogyakarta.
2. Prof. Dr. Dra. Khurul Wardati, M.Si selaku Dekan Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
3. Bapak Dr. Muhammad Mustakim, S.T. M.T. selaku Ketua Program Studi Informatika Fakultas Sains dan Teknologi UIN Sunan Kalijaga Yogyakarta.
4. Bapak Ir. Muhammad Didik Rohmad Wahyudi, ST., MT. selaku dosen penasihat akademik penulis selama kuliah di UIN Sunan Kalijaga.
5. Bapak Nurochman, S.Kom., M.Kom. selaku Dosen Pembimbing. Terima kasih atas arahan, waktu, dan ilmu yang sangat berharga selama proses penyusunan tugas akhir ini.
6. Seluruh Bapak dan Ibu Dosen serta staf akademik UIN Sunan Kalijaga Yogyakarta yang telah memberikan bekal ilmu pengetahuan dan fasilitas selama masa perkuliahan.
7. 295 Solution yang bersedia menjadi mitra penelitian dan sangat kooperatif selama proses pengembangan sistem.
8. Bapak dan Ibuku tercinta. Terima kasih atas segala doa, kasih sayang, dan pengorbanan tanpa batas yang telah diberikan agar penulis dapat menempuh pendidikan tinggi dengan lancar.
9. Kakak dan adikku, Muhammad Quraish Zulhamsyah dan Khairunisa Hernita Putri, yang selalu menjadi motivasi dan memberikan semangat bagi penulis di rumah.
10. Keluarga besar penulis yang senantiasa memberikan doa dan dukungan moral selama proses penyelesaian tugas akhir.

11. Untuk diri sendiri, Muhammad Alfarizi Habibullah. Terima kasih telah berjuang, bertahan melewati keraguan, dan menyelesaikan tanggung jawab ini dengan baik.
12. Maryam Faizah Rosyadi yang selalu setia mendampingi, memberikan semangat, dan menjadi pendengar yang baik dalam setiap proses penyusunan tugas akhir.
13. Ahmad Mushthofa Kamal dan Zhafran Pradistyatama Kuncoro yang selalu berkompetisi secara sehat dan positif serta selalu mendukung atas segala usaha yang berakhir baik untuk diri penulis.
14. Teman-teman Kost Imelda yang menjadi tempat keluh kesah seperjuangan perantauan untuk menimba ilmu di Yogyakarta
15. Teman-teman RPL serta Matdis & Musang yang selalu memberikan dukungan emosional, tawa, dan solidaritas di tengah penatnya perkuliahan.
16. Sahabat-sahabat di luar kampus yang selalu menemani dan berbagi cerita.
17. Rekan-rekan KKN Karangnongko atas kebersamaan dan kenangan berharga selama masa pengabdian masyarakat.
18. Semua pihak yang tidak dapat disebutkan satu per satu namun turut membantu kelancaran tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kata sempurna dan memiliki banyak ruang untuk perbaikan. Oleh karena itu, penulis sangat terbuka terhadap kritik dan saran yang membangun demi penyempurnaan karya ini. Akhir kata, semoga Tugas Akhir ini dapat memberikan manfaat serta bisa dikembangkan lebih baik lagi di masa depan.

STATE ISLAMIC UNIVERSITY
SUNAN KALIJAGA
YOGYAKARTA

DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
LEMBAR PERNYATAAN KEASLIAN	iii
SURAT PERSETUJUAN TUGAS AKHIR	iv
LEMBAR PEDOMAN PENGGUNAAN TUGAS AKHIR	v
INTISARI.....	vi
ABSTRACT.....	vii
HALAMAN MOTTO	viii
HALAMAN PERSEMBAHAN	ix
KATA PENGANTAR	x
DAFTAR ISI.....	xii
DAFTAR TABEL.....	xvi
DAFTAR GAMBAR	xix
DAFTAR LAMPIRAN.....	xx
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	7
1.3 Batasan Masalah	7
1.4 Tujuan	9
1.5 Manfaat	10
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI.....	12
2.1 Tinjauan Pustaka	12
2.2 Landasan Teori.....	15
2.2.1 Rancang Bangun Sistem.....	15
2.2.2 Data Definition Language (DDL) dan MySQL.....	16
2.2.3 Optimasi Kinerja DDL MySQL	17
2.2.4 API Gemini dan Large Language Model	18
2.2.5 Metodologi Pengembangan Sistem Menggunakan Extreme Programming.....	19
BAB III METODE PENGEMBANGAN SISTEM	23
3.1 Metode Pengembangan Sistem	23

3.2 Penerapan Tahapan Extreme Programming pada Tugas Akhir	25
3.2.1 <i>Planning</i> (Perencanaan).....	27
3.2.2 <i>Design</i> (Perancangan).....	28
3.2.3 <i>Coding</i> (Pengkodean)	29
3.2.4 <i>Testing</i> (Pengujian).....	30
3.2.5 <i>Release</i> (Rilis)	31
3.3 Tahapan Iterasi Pengembangan	32
3.3.1 Iterasi 1: Fondasi Aplikasi dan Input DDL	34
3.3.2 Iterasi 2: Pipeline Analisis AI dan Rekomendasi Terstruktur	35
3.3.3 Iterasi 3: Peninjauan Rekomendasi, SQL Diff, Estimasi Dampak, dan ERD	36
3.3.4 Iterasi 4: Finalisasi, Validasi Docker Sandbox, Koreksi Otomatis, dan Output SQL Optimized	37
3.4 Arsitektur Sistem Optimasi DDL Berbasis API Gemini	38
3.5 Spesifikasi Kebutuhan Sistem.....	41
3.5.1 Kebutuhan Fungsional.....	41
3.5.2 Kebutuhan Non-Fungsional.....	42
3.6 Data, Skenario, dan Instrumen Evaluasi Sistem	44
3.6.1 Data Input Evaluasi Sistem	44
3.6.2 Skenario Evaluasi Sistem	46
3.6.3 Instrumen Evaluasi Sistem	47
3.6.4 Batasan Evaluasi Sistem.....	49
BAB IV HASIL DAN PEMBAHASAN	52
4.1 Implementasi Iterasi Pertama.....	53
4.1.1 <i>Planning</i>	53
4.1.2 <i>Design</i>	55
4.1.3 <i>Coding</i>	58
4.1.4 <i>Testing</i>	59
4.1.5 Evaluasi Increment Pertama dan Fokus Pengembangan Berikutnya ...	59
4.2 Implementasi Iterasi Kedua	60
4.2.1 <i>Planning</i>	61
4.2.2 <i>Design</i>	63
4.2.3 <i>Coding</i>	65
4.2.4 <i>Testing</i>	66
4.2.5 Evaluasi Increment Kedua dan Fokus Pengembangan Berikutnya	67

4.3 Implementasi Iterasi Ketiga	68
4.3.1 Planning	68
4.3.2 Design	70
4.3.3 Coding	73
4.3.4 Testing	74
4.3.5 Evaluasi Increment Ketiga dan Fokus Pengembangan Berikutnya	75
4.4 Implementasi Iterasi Keempat	76
4.4.1 Planning	77
4.4.2 Design	79
4.4.3 Coding	81
4.4.4 Testing	82
4.4.5 Evaluasi Increment Keempat dan Hasil Akhir	83
4.5 Release (Rilis)	84
4.6 Data Masukan dan Hasil Pemrosesan Sistem	90
4.6.1 Data Masukan Sistem	91
4.6.2 Hasil Pemrosesan Awal dan Struktur Rekomendasi	92
4.6.3 Interpretasi Confidence, Estimasi Dampak, dan Output Sistem	94
4.7 Pembahasan Hasil Implementasi	96
4.7.1 Perkembangan Sistem pada Setiap Iterasi	96
4.7.2 Peran Evaluasi Increment dan Feedback Stakeholder dalam Pengembangan Sistem	98
4.7.3 Keterkaitan Implementasi dengan Tujuan Tugas Akhir	100
4.7.4 Pembahasan Integrasi API Gemini	101
4.7.5 Pembahasan Peninjauan Rekomendasi dan Visualisasi	102
4.7.6 Pembahasan Validasi Docker Sandbox	103
4.7.7 Pembahasan Koreksi Otomatis Berbasis AI Generatif	104
4.7.8 Pembahasan Hasil Rilis Internal	105
4.7.9 Ringkasan Pembahasan	105
4.8 Hasil Pengujian dan Evaluasi Sistem	106
4.8.1 Pengujian Fungsional Keseluruhan Sistem	107
4.8.2 Pengujian Teknis Backend dan Frontend	110
4.8.3 Pengujian Validasi SQL Optimized	111
4.8.4 Pengujian Koreksi Otomatis Berbasis AI Generatif	112
4.8.5 Evaluasi Kelayakan Sistem oleh Stakeholder	115

4.8.6 Ringkasan Hasil Pengujian dan Evaluasi	117
4.9 Keunggulan Sistem dan Keterbatasan.....	118
4.9.1 Keunggulan Sistem.....	119
4.9.2 Keterbatasan Sistem	121
4.9.3 Ringkasan Keunggulan dan Keterbatasan	124
BAB V PENUTUP.....	126
5.1 Kesimpulan	126
5.2 Saran	127
DAFTAR PUSTAKA	131



DAFTAR TABEL

Tabel 2. 1. Penelitian Terdahulu	14
Tabel 3. 1. Ringkasan Tahapan Iterasi Pengembangan.....	33
Tabel 3. 2. Kebutuhan Fungsional	41
Tabel 3. 3. Kebutuhan Non-Fungsional	43
Tabel 3. 4. Skenario Evaluasi Sistem.....	46
Tabel 3. 5. Instrumen Evaluasi Sistem.....	48
Tabel 4. 1. Kebutuhan Awal Iterasi Pertama.....	54
Tabel 4. 2. User Story Iterasi Pertama	54
Tabel 4. 3. Acceptance Criteria Iterasi Pertama.....	54
Tabel 4. 4. Komponen Utama Iterasi Pertama	56
Tabel 4. 5. Pengujian Iterasi Pertama.....	59
Tabel 4. 6. Evaluasi Stakeholder Iterasi Pertama.....	60
Tabel 4. 7. Kebutuhan Pengembangan Iterasi Kedua	61
Tabel 4. 8. User Story Iterasi Kedua.....	62
Tabel 4. 9. Acceptance Criteria Iterasi Kedua	62
Tabel 4. 10. Komponen Utama Iterasi Kedua.....	63
Tabel 4. 11. Pengujian Iterasi Kedua	66
Tabel 4. 12. Evaluasi Stakeholder Iterasi Kedua	67
Tabel 4. 13. Kebutuhan Pengembangan Iterasi Ketiga.....	69
Tabel 4. 14. User Story Iterasi Ketiga.....	69
Tabel 4. 15. Acceptance Criteria Iterasi Ketiga	70
Tabel 4. 16. Komponen Utama Iterasi Ketiga.....	71
Tabel 4. 17. Pengujian Iterasi Ketiga	74

Tabel 4. 18. Evaluasi Stakeholder Iterasi Ketiga	75
Tabel 4. 19. Kebutuhan Pengembangan Iterasi Keempat	77
Tabel 4. 20. User Story Iterasi Keempat	77
Tabel 4. 21. Acceptance Criteria Iterasi Keempat	78
Tabel 4. 22. Komponen Utama Iterasi Keempat.....	79
Tabel 4. 23. Pengujian Iterasi Keempat	82
Tabel 4. 24. Evaluasi Stakeholder Iterasi Keempat	83
Tabel 4. 25. Status Rilis Sistem	85
Tabel 4. 26. Lingkungan dan Komponen Rilis	86
Tabel 4. 27. Fitur pada Rilis Internal Sistem	86
Tabel 4. 28. Kriteria Rilis Sistem.....	88
Tabel 4. 29. Batasan Rilis Sistem.....	89
Tabel 4. 30. Ringkasan Data Masukan Sistem.....	91
Tabel 4. 31. Hasil Pemrosesan Awal DDL MySQL	92
Tabel 4. 32. Struktur Output Rekomendasi AI	93
Tabel 4. 33. Interpretasi Indikator Hasil Analisis Sistem	95
Tabel 4. 34. Perkembangan Sistem pada Setiap Iterasi	97
Tabel 4. 35. Hubungan Evaluasi Increment dan Feedback Stakeholder dengan Pengembangan Sistem	98
Tabel 4. 36. Keterkaitan Tujuan Tugas Akhir dengan Hasil Implementasi.....	100
Tabel 4. 37. Pengujian Fungsional Keseluruhan Sistem.....	107
Tabel 4. 38. Ringkasan Pengujian End-to-End Alur Utama	109
Tabel 4. 39. Ringkasan Pengujian Teknis Backend dan Frontend.....	110

Tabel 4. 40. Pengujian Validasi SQL Optimized.....	111
Tabel 4. 41. Contoh Alur Koreksi SQL Berdasarkan Error Log	113
Tabel 4. 42. Pengujian Alur Koreksi Otomatis Berbasis AI Generatif.....	113
Tabel 4. 43. Instrumen Evaluasi Stakeholder.....	115
Tabel 4. 44. Evaluasi Kelayakan Sistem oleh Stakeholder.....	116
Tabel 4. 45. Ringkasan Hasil Pengujian dan Evaluasi Sistem.....	117
Tabel 4. 46. Keunggulan Sistem	119
Tabel 4. 47. Keterbatasan Sistem.....	121
Tabel 4. 48. Ringkasan Keunggulan dan Keterbatasan Sistem.....	124

DAFTAR GAMBAR

Gambar 3. 1 Rancangan Model Penelitian Sistem Optimasi DDL MySQL Berbasis API Gemini (Sumber: Hasil perancangan penulis).....	25
Gambar 3. 2. Penerapan Siklus Extreme Programming pada Pengembangan Sistem (Sumber: Diadaptasi dari[46], [48] dan disesuaikan oleh penulis).....	26
Gambar 3. 3. Arsitektur Sistem Optimasi DDL Berbasis API Gemini (Sumber: Hasil perancangan penulis)	38
Gambar 4. 1. Halaman Login dan Registrasi.....	56
Gambar 4. 2. Dashboard Project dan Modal Pembuatan Project.....	57
Gambar 4. 3. Form Upload SQL dan Status Job Awal	57
Gambar 4. 4. Pilihan Workload dan Status Proses Analisis	64
Gambar 4. 5. Tampilan Status Job Analisis	64
Gambar 4. 6. Tampilan Hasil Rekomendasi AI	65
Gambar 4. 7. Tampilan Review Suggestion dan Checkbox.....	71
Gambar 4. 8. Tampilan SQL Diff Preview	72
Gambar 4. 9. Tampilan Estimasi Dampak Performa	72
Gambar 4. 10. Tampilan Visualisasi ERD	73
Gambar 4. 11. Tampilan Proses Finalisasi Suggestion.....	80
Gambar 4. 12. Tampilan Ringkasan Validasi Docker Sandbox.....	80
Gambar 4. 13. Tampilan Download SQL optimized	81

DAFTAR LAMPIRAN

Lampiran 1. Data Uji DDL MySQL	137
Lampiran 2. Skenario Pengujian End-To-End Sistem	145
Lampiran 3. Bukti Pengujian End-to-End Sistem	148
Lampiran 4. Artefak Hasil Optimasi SQL	153
Lampiran 5. Bukti Validasi Docker Sandbox dan Self-Correction	154



BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam era transformasi digital, sistem perangkat lunak tidak dapat dipisahkan dari peran basis data sebagai fondasi penyimpanan, pengelolaan, dan pengaksesan informasi. Basis data yang dirancang dengan baik memungkinkan aplikasi menyimpan data secara terstruktur, menjaga konsistensi informasi, serta mendukung proses pengambilan data secara efisien. Seiring meningkatnya volume data dan kompleksitas aplikasi, kinerja Database Management System (DBMS) menjadi aspek penting dalam menentukan kualitas layanan sistem. Taipalus menjelaskan bahwa perbandingan kinerja DBMS masih menjadi perhatian penting dalam penelitian dan praktik industri karena performa database berpengaruh terhadap efisiensi sistem dan pemilihan teknologi yang sesuai kebutuhan [1]. Dalam konteks pengembangan aplikasi, MySQL juga banyak digunakan karena mampu mendukung proses penyimpanan dan pengelolaan data secara fleksibel, efisien, dan terstruktur [2]. Selain itu, pemilihan DBMS yang tepat menjadi bagian penting dalam pengembangan sistem karena setiap DBMS memiliki karakteristik yang berbeda sesuai kebutuhan aplikasi [3]. Penelitian lain mengenai perancangan basis data relasional juga menunjukkan bahwa desain database yang terstruktur dapat membantu meningkatkan efisiensi pengelolaan data, mengurangi kesalahan, dan mendukung operasional sistem secara lebih baik [4]. Oleh karena itu, kualitas rancangan basis data perlu diperhatikan sejak tahap awal pengembangan perangkat lunak karena database bukan hanya tempat penyimpanan data, tetapi juga komponen yang memengaruhi keandalan dan efisiensi sistem secara keseluruhan.

Salah satu aspek penting dalam pengelolaan basis data relasional adalah rancangan skema database yang direpresentasikan melalui Data Definition Language (DDL). Dalam MySQL, DDL digunakan untuk mendefinisikan dan

mengubah struktur objek database, seperti tabel, kolom, tipe data, index, constraint, dan relasi antar tabel. Dokumentasi resmi MySQL menjelaskan bahwa CREATE TABLE digunakan untuk membuat struktur tabel, sedangkan ALTER TABLE digunakan untuk mengubah struktur tabel, seperti menambah atau menghapus kolom, membuat atau menghapus index, mengubah tipe data, serta mengubah karakteristik tabel [5], [6]. Selain itu, index dapat membantu meningkatkan performa pencarian data karena index berperan sebagai penunjuk terhadap baris yang sesuai dengan kondisi query [7]. Normalisasi database juga menjadi bagian penting dalam rancangan skema karena berperan dalam mengurangi redundansi dan menjaga integritas data, meskipun tetap perlu mempertimbangkan kebutuhan performa sistem [8]. Dengan demikian, DDL MySQL menjadi bagian penting yang perlu diperhatikan karena kualitas rancangan struktur skema dapat memengaruhi konsistensi data, efisiensi akses, serta kemudahan pemeliharaan dan pengembangan sistem.

Dalam praktik pengembangan perangkat lunak, permasalahan desain database sering muncul karena pengembang lebih berfokus pada penyelesaian fitur aplikasi dibandingkan pada kualitas struktur database yang digunakan. Kondisi ini dapat terjadi terutama pada pengembang pemula yang belum memiliki pengalaman cukup dalam merancang skema relasional, menentukan normalisasi, memilih tipe data, menyusun index, dan menerapkan constraint secara tepat. Presler-Marshall dkk. menunjukkan bahwa mahasiswa atau programmer pemula masih sering melakukan kesalahan dalam penulisan dan pemahaman SQL, sehingga alat bantu untuk mengidentifikasi dan memperbaiki kesalahan SQL menjadi relevan dalam proses pembelajaran maupun pengembangan sistem [9]. Selain itu, penerapan normalisasi database juga penting karena dapat mengurangi redundansi dan meningkatkan efisiensi penyimpanan data [10]. Namun, desain logis database tetap perlu dipertimbangkan secara hati-hati karena dapat memengaruhi ukuran database, kompleksitas query, performa, dan efisiensi sistem [11]. Apabila keputusan desain database dilakukan secara terburu-buru atau tanpa evaluasi yang memadai, kesalahan tersebut dapat berkembang menjadi technical debt yang

berdampak pada pemeliharaan dan biaya perbaikan sistem di kemudian hari [12]. Oleh karena itu, diperlukan alat bantu yang dapat membantu pengembang dalam mengevaluasi struktur DDL MySQL secara lebih sistematis sejak tahap awal pengembangan.

Evaluasi dan optimasi database pada umumnya dapat dilakukan melalui tinjauan manual oleh pengembang atau Database Administrator (DBA). Namun, pendekatan manual memiliki keterbatasan karena membutuhkan pengalaman teknis, waktu analisis, serta pemahaman yang baik terhadap struktur database dan pola penggunaan data. Seiring meningkatnya volume data, kompleksitas workload, dan kebutuhan aplikasi, proses tuning database menjadi semakin sulit dilakukan secara manual. Siddiqui dan Wu menjelaskan bahwa automated index tuning dapat membantu merekomendasikan index untuk mempercepat eksekusi query, tetapi masih menghadapi tantangan seperti skalabilitas workload, pemilihan kandidat index, dan risiko penurunan performa pada lingkungan produksi [13]. Kajian mengenai self-tuning database systems juga menunjukkan bahwa otomatisasi desain skema dan tuning database masih menjadi topik penting karena proses desain database melibatkan representasi yang kompleks, baik pada basis data SQL maupun NoSQL [14]. Selain itu, automatic configuration tuning pada cloud database menunjukkan bahwa optimasi database tidak hanya berkaitan dengan struktur skema, tetapi juga dipengaruhi oleh parameter konfigurasi, karakteristik workload, pemilihan fitur, dan evaluasi tuning yang sistematis [15]. Beberapa pendekatan terbaru bahkan mulai memanfaatkan LLM untuk membantu database tuning, tetapi fokusnya masih lebih banyak pada konfigurasi sistem atau parameter tuning, bukan pada mekanisme terpadu untuk membaca DDL, menghasilkan rekomendasi refactoring skema, memberikan kontrol pengguna, dan memvalidasi SQL hasil optimasi sebelum digunakan [16]. Oleh karena itu, masih diperlukan sistem yang dapat menggabungkan analisis DDL, rekomendasi berbasis AI, kontrol pengguna, serta validasi SQL dalam satu alur yang terintegrasi.

Keterbatasan evaluasi DDL secara manual menunjukkan perlunya alat bantu yang mampu membaca struktur database dan memberikan rekomendasi awal secara lebih terarah. Dalam konteks tersebut, Large Language Model (LLM) menjadi relevan untuk dibahas karena memiliki kemampuan memahami konteks teknis, struktur kode, serta pola relasi pada data. Perkembangan Large Language Model (LLM) membuka peluang baru dalam proses analisis dan optimasi database karena model AI memiliki kemampuan untuk memahami konteks teknis, struktur kode, serta pola relasi pada data. Dalam konteks pengembangan sistem, Gemini API dapat digunakan untuk menghasilkan respons melalui endpoint `generateContent`, sehingga dapat dimanfaatkan untuk menganalisis input teknis seperti struktur DDL MySQL dan menghasilkan rekomendasi berbasis konteks [17]. Gemini API juga mendukung structured output, yaitu kemampuan menghasilkan respons dalam format terstruktur berdasarkan skema tertentu, sehingga keluaran model dapat diproses lebih mudah oleh sistem dan tidak hanya ditampilkan sebagai teks bebas [18]. Potensi LLM dalam ranah database juga ditunjukkan oleh penelitian Database Normalization via Dual-LLM Self-Refinement yang memanfaatkan arsitektur dual-model untuk membantu proses normalisasi skema database secara otomatis [19]. Selain itu, penelitian Making Databases Faster with LLM Evolutionary Sampling menunjukkan bahwa LLM dapat memanfaatkan pengetahuan semantik untuk menemukan optimasi yang tidak selalu dapat ditangkap oleh heuristik konvensional [20]. Namun, penelitian-penelitian tersebut belum secara khusus membahas sistem yang berfokus pada analisis DDL MySQL yang sudah ada, menghasilkan rekomendasi refactoring skema, memberikan kontrol kepada pengguna melalui review suggestion, serta memvalidasi SQL hasil optimasi sebelum diterima pengguna. Oleh karena itu, tugas akhir ini diarahkan untuk mengisi celah tersebut melalui pengembangan sistem optimasi DDL MySQL berbasis API Gemini dengan dukungan validasi sandbox.

Meskipun LLM memiliki kemampuan untuk menghasilkan rekomendasi dan kode, hasil keluaran AI tetap memiliki risiko berupa halusinasi, kesalahan

sintaks, atau kode yang terlihat benar tetapi belum tentu dapat dijalankan. Risiko ini penting diperhatikan dalam konteks optimasi DDL MySQL karena SQL hasil rekomendasi berpotensi mengubah struktur database. Penelitian CodeHalu menunjukkan bahwa LLM dapat menghasilkan kode yang secara sintaks tampak benar dan masuk akal, tetapi tidak selalu berjalan sesuai kebutuhan ketika diverifikasi melalui eksekusi [21]. SecCodeBench-V2 juga menegaskan bahwa evaluasi kode yang dihasilkan LLM perlu memperhatikan aspek keamanan dan validitas karena kode generatif dapat mengandung kelemahan yang tidak cukup dinilai melalui pemeriksaan statis saja [22]. Oleh karena itu, SQL hasil optimasi dari API Gemini tidak boleh langsung diterima sebagai hasil akhir tanpa proses validasi. Docker dapat digunakan untuk menjalankan aplikasi di dalam container, yaitu lingkungan yang terisolasi dan berisi kebutuhan aplikasi agar dapat berjalan secara konsisten [23]. Dalam tugas akhir ini, Docker Sandbox digunakan sebagai mekanisme validasi SQL hasil optimasi, sehingga SQL dapat diuji terlebih dahulu pada lingkungan terpisah sebelum disediakan kepada pengguna. Pendekatan ini sejalan dengan penggunaan sandbox pada evaluasi kode LLM, yaitu memisahkan proses eksekusi dari sistem utama untuk mengurangi risiko terhadap lingkungan host [22], [24].

Kebutuhan terhadap alur analisis, rekomendasi, dan validasi SQL yang terstruktur juga relevan pada konteks pengembangan sistem di 295 Solution. 295 Solution bergerak pada bidang sistem gate parking dengan kebutuhan bisnis seperti cashless parking, cash dan cashless, manless parking, IPL, serta ANPR. Sistem tersebut memiliki ketergantungan terhadap database MySQL untuk menyimpan data operasional, seperti transaksi masuk dan keluar, data member kartu, data kendaraan, data gate, data pembayaran, serta log akses. Berdasarkan observasi awal dan diskusi dengan stakeholder 295 Solution, permasalahan stabilitas pada sistem gate parking dapat dipengaruhi oleh beberapa lapisan, seperti hardware, software, dan database. Pada sisi hardware, kondisi perangkat dinilai relatif sesuai dengan kebutuhan operasional. Namun, pada sisi software dan database masih diperlukan evaluasi lebih lanjut, salah

satunya berkaitan dengan race condition pada software dan kebutuhan perapihan struktur database.

Dalam tugas akhir ini, fokus evaluasi diarahkan pada lapisan database, khususnya struktur DDL MySQL. Struktur DDL perlu diperhatikan karena menjadi representasi rancangan database yang menentukan tabel, kolom, tipe data, index, constraint, dan relasi antar tabel. Struktur database yang kurang rapi dapat menyulitkan proses pemeliharaan, memperbesar risiko inkonsistensi data, dan membuat proses pengembangan sistem menjadi kurang terarah. Tugas akhir ini tidak berfokus untuk membuktikan bahwa struktur DDL menjadi penyebab utama permasalahan stabilitas pada sistem gate parking, melainkan membatasi ruang lingkup pada pengembangan alat bantu untuk menganalisis dan mengoptimasi struktur DDL MySQL. Oleh karena itu, sistem yang dikembangkan diarahkan untuk membaca struktur DDL MySQL, menghasilkan rekomendasi optimasi menggunakan API Gemini, memberikan kontrol kepada pengguna melalui review suggestion, serta memvalidasi SQL hasil optimasi menggunakan Docker Sandbox sebelum diterima sebagai output akhir.

Urgensi evaluasi DDL pada konteks ini terletak pada perannya sebagai representasi struktur dasar database. Melalui DDL, struktur tabel, kolom, tipe data, index, constraint, dan relasi antar tabel dapat dianalisis tanpa harus menggunakan data operasional asli. Hal ini penting karena sistem gate parking menyimpan data yang terus bertambah, seperti transaksi parkir, data member, kendaraan, gate, pembayaran, dan log akses. Struktur database yang kurang tertata dapat menyulitkan pemeliharaan, memperbesar risiko inkonsistensi data, dan menghambat proses pengembangan fitur lanjutan.

Dalam tugas akhir ini, optimasi kinerja DDL MySQL dimaknai sebagai upaya menganalisis dan memperbaiki rancangan struktur skema database yang direpresentasikan dalam perintah DDL, bukan mengukur kecepatan eksekusi perintah DDL itu sendiri maupun membuktikan peningkatan performa runtime pada database produksi. Fokus optimasi diarahkan pada evaluasi elemen skema seperti kesesuaian tipe data, penggunaan index, penerapan constraint,

normalisasi, dan hubungan antar tabel agar struktur database lebih efisien, konsisten, dan mudah dikembangkan.

Berdasarkan uraian tersebut, tugas akhir ini mengangkat judul “Optimasi Kinerja Data Definition Language (DDL) MySQL Berbasis API Gemini” sebagai upaya untuk membangun sistem yang mampu membantu proses analisis, rekomendasi, dan validasi SQL hasil optimasi secara lebih terstruktur.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam rancang bangun sistem ini adalah sebagai berikut:

1. Bagaimana merancang dan mengimplementasikan sistem berbasis web yang mengintegrasikan AI generatif melalui API Gemini untuk menganalisis skema DDL MySQL serta menghasilkan rekomendasi refactoring terhadap potensi inefisiensi struktur database?
2. Bagaimana merancang dan mengimplementasikan mekanisme validasi berbasis Docker Sandbox serta koreksi otomatis berbasis AI generatif untuk memastikan SQL hasil optimasi valid secara sintaksis dan dapat dieksekusi sebelum diterima oleh pengguna?

1.3 Batasan Masalah

Agar penelitian lebih terarah dan sesuai dengan rumusan masalah yang telah ditetapkan, maka ruang lingkup tugas akhir ini dibatasi pada beberapa hal berikut:

1. Objek Penelitian: Studi kasus dilakukan pada kebutuhan evaluasi struktur DDL MySQL pada sistem gate parking 295 Solution, khususnya pada database yang digunakan untuk menyimpan data operasional seperti transaksi parkir, data member, data kendaraan, data gate, dan log akses.

2. Ruang Lingkup Sistem: Sistem yang dikembangkan berbentuk aplikasi berbasis web yang digunakan untuk membantu proses analisis DDL MySQL, menghasilkan rekomendasi optimasi menggunakan AI generatif melalui API Gemini, menyediakan fitur review suggestion, memvalidasi SQL hasil optimasi, menjalankan koreksi otomatis berbasis AI generatif apabila validasi gagal pada skenario tertentu, dan menghasilkan file SQL optimized.
3. Ruang Lingkup Optimasi: Optimasi dibatasi pada level struktur skema DDL MySQL, meliputi normalisasi, efisiensi tipe data, strategi indexing, constraint, dan relasi antar tabel. Tugas akhir ini tidak membahas optimasi query DML, stored procedure, trigger, view, maupun konfigurasi server database secara mendalam.
4. Mekanisme Rekomendasi: Rekomendasi optimasi dihasilkan oleh AI generatif melalui API Gemini berdasarkan analisis terhadap input DDL MySQL. Rekomendasi yang dihasilkan tidak langsung diterapkan secara otomatis, tetapi harus ditinjau terlebih dahulu oleh pengguna melalui fitur review suggestion.
5. Mekanisme Validasi: Validasi SQL hasil optimasi dilakukan menggunakan Docker Sandbox. Apabila validasi menghasilkan error pada skenario tertentu, sistem dapat menggunakan error log sebagai konteks koreksi otomatis berbasis AI generatif. Validasi ini hanya bertujuan untuk memastikan bahwa SQL hasil optimasi valid secara sintaksis dan dapat dieksekusi pada lingkungan pengujian terpisah. Validasi tidak mencakup

load testing, stress testing, maupun benchmark performa aktual pada database produksi.

6. Batasan Hasil Optimasi: Sistem hanya menghasilkan rekomendasi dan file SQL optimized yang dapat diunduh oleh pengguna. Sistem tidak melakukan perubahan langsung pada database operasional 295 Solution.
7. Batasan Permasalahan Studi Kasus: Tugas akhir ini tidak berfokus untuk menyelesaikan seluruh permasalahan teknis pada software gate parking lama, seperti blue screen, race condition, atau kesalahan logic bisnis. Fokus penelitian hanya berada pada analisis dan optimasi struktur DDL MySQL sebagai salah satu bagian dari upaya evaluasi dan standardisasi sistem.
8. Batasan Infrastruktur: Sistem diimplementasikan dan diuji pada lingkungan local/development menggunakan Docker Compose. Tugas akhir ini belum mencakup deployment ke server produksi, server on-premise 295 Solution, cloud infrastructure, CI/CD production, maupun hardening keamanan untuk penggunaan operasional penuh.

1.4 Tujuan

Berdasarkan rumusan masalah yang telah ditetapkan, tujuan dari tugas akhir ini adalah sebagai berikut:

1. Merancang dan mengimplementasikan sistem berbasis web yang mengintegrasikan AI generatif melalui API Gemini untuk menganalisis skema DDL MySQL serta menghasilkan rekomendasi refactoring terhadap potensi inefisiensi struktur database.
2. Merancang dan mengimplementasikan mekanisme validasi berbasis Docker Sandbox serta koreksi otomatis berbasis AI generatif untuk memastikan

SQL hasil optimasi valid secara sintaksis dan dapat dieksekusi pada lingkungan pengujian terpisah sebelum diterima oleh pengguna.

1.5 Manfaat

Tugas akhir ini diharapkan dapat memberikan manfaat baik secara praktis maupun akademik sebagai berikut:

1. Manfaat Praktis

- a. Bagi 295 Solution: Sistem yang dikembangkan dapat menjadi alat bantu dalam proses evaluasi struktur DDL MySQL pada sistem gate parking. Sistem ini membantu membaca struktur database, menghasilkan rekomendasi optimasi, menyediakan fitur review suggestion, serta memvalidasi SQL hasil optimasi sebelum diterima sebagai hasil akhir.
- b. Bagi Pengembang atau Stakeholder Teknis: Sistem ini dapat membantu pengembang dalam menganalisis struktur DDL MySQL secara lebih terarah. Dengan adanya rekomendasi berbasis AI generatif melalui API Gemini, pengembang dapat memperoleh masukan terkait potensi perbaikan struktur database, seperti penggunaan index, tipe data, constraint, normalisasi, dan relasi antar tabel.
- c. Bagi Proses Evaluasi dan Standardisasi Database: Sistem ini dapat mendukung proses evaluasi dan standardisasi struktur database, khususnya pada sistem yang sebelumnya dikembangkan oleh beberapa pihak berbeda. Dengan adanya alur kerja yang terstruktur, proses analisis DDL tidak hanya dilakukan secara manual, tetapi

dibantu oleh sistem yang menyediakan rekomendasi, kontrol pengguna, validasi sandbox, dan output SQL optimized.

- d. Bagi Keamanan Proses Optimasi SQL: Sistem ini membantu mengurangi risiko penggunaan SQL hasil optimasi yang belum tervalidasi. Melalui Docker Sandbox, SQL hasil optimasi dapat diuji terlebih dahulu pada lingkungan terpisah sehingga tidak langsung dijalankan pada database operasional.

2. Manfaat Akademik

- a. Menambah Referensi pada Bidang Rekayasa Perangkat Lunak dan Database Engineering: Tugas akhir ini dapat menjadi referensi mengenai pemanfaatan AI generatif dan Large Language Model, khususnya API Gemini, dalam rancang bangun sistem berbasis web untuk membantu analisis dan optimasi struktur DDL MySQL.
- b. Memberikan Pendekatan pada Pengembangan Sistem Optimasi DDL: Tugas akhir ini memberikan pendekatan yang berbeda dari sistem berbasis LLM yang lebih banyak berfokus pada Text-to-SQL, interaksi database berbasis chat, atau tuning parameter database. Sistem yang dikembangkan berfokus pada analisis DDL MySQL yang sudah ada, pemberian rekomendasi refactoring, kontrol pengguna melalui review suggestion, serta validasi SQL hasil optimasi.
- c. Memberikan Contoh Penerapan Validasi Sandbox pada Hasil AI: Tugas akhir ini menunjukkan penerapan Docker Sandbox untuk

memvalidasi SQL hasil optimasi yang dihasilkan dengan bantuan AI generatif melalui API Gemini. Pendekatan ini dapat menjadi contoh penerapan sistem yang menggabungkan AI generatif dengan proses validasi eksekusi agar hasil yang diberikan lebih terkontrol.



BAB V

PENUTUP

Bab ini berisi kesimpulan dan saran berdasarkan hasil pengembangan sistem yang telah dilakukan. Kesimpulan disusun berdasarkan rumusan masalah, tujuan tugas akhir, hasil implementasi sistem, serta hasil pengujian yang telah dibahas pada Bab IV. Sementara itu, saran diberikan sebagai masukan untuk pengembangan sistem pada penelitian atau implementasi berikutnya.

5.1 Kesimpulan

Berdasarkan hasil pengembangan dan pengujian sistem optimasi kinerja Data Definition Language (DDL) MySQL berbasis API Gemini, diperoleh beberapa kesimpulan sebagai berikut:

1. Sistem berbasis web yang mengintegrasikan AI generatif melalui API Gemini berhasil dirancang dan diimplementasikan untuk membantu proses analisis skema DDL MySQL serta menghasilkan rekomendasi refactoring terhadap potensi inefisiensi struktur database. Sistem menerima input file DDL MySQL, melakukan parsing dan sanitasi, menyusun konteks skema database, kemudian mengirimkan konteks tersebut ke API Gemini untuk menghasilkan rekomendasi optimasi. Rekomendasi yang dihasilkan mencakup aspek struktur skema fisik database, seperti strategi indexing, tipe data, constraint, dan perbaikan struktur tabel. Hasil rekomendasi juga disajikan secara terstruktur dan dilengkapi dengan fitur review suggestion agar pengguna tetap memiliki kontrol dalam menerima atau menolak rekomendasi sebelum dibentuk menjadi SQL hasil optimasi.
2. Mekanisme validasi berbasis Docker Sandbox serta koreksi otomatis berbasis AI generatif berhasil dirancang dan diimplementasikan

untuk memastikan SQL hasil optimasi valid secara sintaksis dan dapat dieksekusi sebelum diterima oleh pengguna. SQL hasil optimasi tidak langsung diberikan sebagai hasil akhir, tetapi terlebih dahulu diuji pada lingkungan Docker Sandbox yang terpisah dari database operasional. Apabila SQL hasil optimasi gagal divalidasi, sistem dapat menggunakan error log dari Docker Sandbox sebagai konteks koreksi otomatis melalui API Gemini. Berdasarkan hasil pengujian, sistem mampu melakukan koreksi SQL, menjalankan validasi ulang, dan menghasilkan file SQL optimized yang dapat diunduh dalam format .sql.

Dengan demikian, sistem yang dikembangkan telah menjawab rumusan masalah pada tugas akhir ini. Sistem mampu mengintegrasikan API Gemini untuk mendeteksi dan memberikan rekomendasi refactoring terhadap potensi inefisiensi skema DDL MySQL, serta menyediakan mekanisme validasi Docker Sandbox dan koreksi otomatis untuk memastikan SQL hasil optimasi valid dan executable sebelum diterima pengguna. Sistem ini telah diuji melalui alur end-to-end pada lingkungan local/development dan dinyatakan layak sebagai rilis internal atau prototype final tugas akhir. Namun, sistem belum diklaim sebagai sistem produksi penuh karena belum dilakukan deployment pada server produksi maupun pengujian performa aktual pada database operasional.

5.2 Saran

Berdasarkan hasil pengembangan sistem yang telah dilakukan, sistem optimasi kinerja Data Definition Language (DDL) MySQL berbasis API Gemini masih memiliki beberapa peluang pengembangan yang dapat dilakukan pada tahap berikutnya. Sistem yang dikembangkan pada tugas akhir ini telah berhasil berjalan sebagai rilis internal pada lingkungan local/development, namun belum diterapkan pada server produksi atau lingkungan operasional penuh. Oleh karena itu, pengembangan selanjutnya

dapat diarahkan pada proses deployment ke server internal atau server produksi dengan memperhatikan aspek keamanan, monitoring, backup, penggunaan HTTPS, pengelolaan akses pengguna, serta konfigurasi infrastruktur yang lebih stabil.

Selain itu, pengujian sistem pada tugas akhir ini masih berfokus pada validasi sintaks dan keterlaksanaan SQL melalui Docker Sandbox. Validasi tersebut sudah mampu membuktikan bahwa SQL hasil optimasi dapat dieksekusi pada lingkungan terpisah, tetapi belum mengukur peningkatan performa secara empiris pada database dengan volume data besar. Pengembangan berikutnya dapat menambahkan pengujian performa aktual pada lingkungan pengujian terkontrol, seperti perbandingan waktu eksekusi query sebelum dan sesudah optimasi, penggunaan index, beban memori, ukuran tabel, serta dampak perubahan struktur DDL terhadap kinerja database. Pengujian tersebut perlu dilakukan dengan dataset dan workload yang representatif agar dampak optimasi dapat diukur secara lebih empiris tanpa mencampurkannya dengan klaim hasil pada tugas akhir ini.

Pengembangan berikutnya juga dapat menggunakan variasi DDL MySQL yang lebih beragam dari sistem gate parking 295 Solution, seperti skema untuk transaksi parkir, data member, kendaraan, gate, log akses, IPL, dan ANPR. Hal ini diperlukan agar kemampuan analisis sistem dapat diuji pada struktur database yang lebih representatif dan sesuai dengan kebutuhan operasional 295 Solution. Dengan penggunaan variasi DDL yang lebih luas, sistem dapat dievaluasi tidak hanya pada satu bentuk skema, tetapi juga pada beberapa karakteristik database yang berbeda sesuai kebutuhan sistem gate parking.

Pengembangan sistem juga dapat difokuskan pada peningkatan kemampuan parsing DDL MySQL. Sistem perlu dibuat lebih akurat dalam membaca struktur database yang kompleks, seperti foreign key, index, constraint, trigger, view, dan variasi sintaks DDL lainnya. Dengan kemampuan parsing yang lebih lengkap, konteks yang dikirimkan ke API Gemini akan

menjadi lebih baik sehingga rekomendasi optimasi yang dihasilkan dapat semakin relevan dan sesuai dengan kondisi struktur database yang dianalisis.

Dari sisi fitur, sistem dapat dikembangkan dengan menyempurnakan tampilan perbandingan antara DDL awal dan DDL hasil optimasi. Pengembangan tersebut dapat mencakup penyajian perubahan SQL yang lebih rinci berdasarkan setiap rekomendasi, pengelompokan perubahan berdasarkan tabel atau kategori optimasi, serta penyediaan riwayat keputusan pengguna terhadap rekomendasi yang diterima atau ditolak. Selain itu, fitur review suggestion juga dapat diperluas dengan menambahkan catatan pengguna atau alasan penerimaan dan penolakan rekomendasi agar proses evaluasi rekomendasi menjadi lebih terdokumentasi.

Mekanisme koreksi otomatis berbasis AI generatif juga masih dapat dikembangkan lebih lanjut. Pengujian pada tugas akhir ini telah menunjukkan bahwa sistem dapat menggunakan error log dari Docker Sandbox sebagai konteks koreksi pada skenario tertentu. Namun, pengembangan berikutnya dapat memperluas variasi skenario error, batas percobaan koreksi, mekanisme fallback ketika koreksi otomatis gagal, serta penyajian riwayat koreksi agar pengguna dapat mengetahui perubahan yang dilakukan oleh sistem sebelum SQL hasil koreksi diterima sebagai output akhir.

Dari sisi evaluasi rekomendasi, penelitian berikutnya dapat membandingkan hasil rekomendasi API Gemini dengan hasil analisis Database Administrator (DBA), aturan optimasi database konvensional, atau model AI lain. Perbandingan tersebut dapat digunakan untuk menilai kesesuaian, konsistensi, dan kualitas rekomendasi optimasi yang dihasilkan sistem. Dengan demikian, pengembangan sistem tidak hanya berfokus pada keberhasilan alur teknis, tetapi juga pada kualitas rekomendasi yang diberikan kepada pengguna.

Terakhir, apabila sistem akan digunakan secara operasional, diperlukan penguatan aspek keamanan dan tata kelola sistem. Pengembangan tersebut dapat mencakup manajemen API key yang lebih aman, pembatasan akses berbasis role, audit log aktivitas pengguna, validasi input yang lebih ketat, pengujian keamanan aplikasi, serta mekanisme backup dan recovery. Dengan

pengembangan tersebut, sistem dapat ditingkatkan dari prototype final pada lingkungan local/development menuju sistem yang lebih siap digunakan pada lingkungan operasional.



DAFTAR PUSTAKA

- [1] T. Taipalus, "Database management system performance comparisons: A systematic literature review," *Journal of Systems and Software*, vol. 208, Feb. 2024, doi: 10.1016/j.jss.2023.111872.
- [2] "Pengembangan database Management system menggunakan My SQL".
- [3] K. Y. Santika, E. Triandini, P. Desiana, and W. Ayu, "Systematic Literature Review untuk Identifikasi Pemilihan Database Management System dalam Pengembangan Sistem." [Online]. Available: <https://scholar.google.com>,
- [4] M. Azuddin *et al.*, "Design and Implementation of a Relational Database for an Academic Information System," *International Transactions on Artificial Intelligence (ITALIC)*, vol. 3, no. 2, pp. 171–180, May 2025, doi: 10.33050/italic.v3i2.801.
- [5] Oracle Corporation, "MySQL 8.4 Reference Manual: 15.1.20 CREATE TABLE Statement," Oracle Corporation. Accessed: May 20, 2026. [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/create-table.html>
- [6] Oracle Corporation, "MySQL 8.4 Reference Manual: ALTER TABLE Statement," Oracle Corporation. Accessed: May 20, 2026. [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/alter-table.html>
- [7] Oracle Corporation, "MySQL 8.4 Reference Manual: Optimization and Indexes," Oracle Corporation. Accessed: May 20, 2026. [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/optimization-indexes.html>

- [8] M. Fotache, M. I. Cluci, T. Taipalus, and G. Talaba, “The effects of database normalization on decision support system performance,” *Inf. Syst.*, vol. 136, Feb. 2026, doi: 10.1016/j.is.2025.102636.
- [9] K. Presler-Marshall, S. Heckman, and K. T. Stolee, “SQLRepair: Identifying and Repairing Mistakes in Student-Authored SQL Queries,” Feb. 2021, [Online]. Available: <http://arxiv.org/abs/2102.05729>
- [10] M. Hardini, V. Agarwal, D. Apriani, I. A. Widjaya, E. Setiawaty, and N. Nurasiah, “Application of Database Normalization in Increasing Data Storage Efficiency,” *International Transactions on Artificial Intelligence (ITALIC)*, vol. 3, no. 2, pp. 201–211, May 2025, doi: 10.33050/italic.v3i2.799.
- [11] T. Taipalus, “On the effects of logical database design on database size, query complexity, query performance, and energy consumption,” Jan. 2025, [Online]. Available: <http://arxiv.org/abs/2501.07449>
- [12] V. Lenarduzzi, T. Besker, D. Taibi, A. Martini, and F. Arcelli Fontana, “A systematic literature review on Technical Debt prioritization: Strategies, processes, factors, and tools,” *Journal of Systems and Software*, vol. 171, Jan. 2021, doi: 10.1016/j.jss.2020.110827.
- [13] T. Siddiqui and W. Wu, “ML-Powered Index Tuning: An Overview of Recent Progress and Open Challenges,” Aug. 2023, [Online]. Available: <http://arxiv.org/abs/2308.13641>
- [14] M. Mozaffari, A. Dignös, J. Gamper, and U. Störl, “Self-tuning Database Systems: A Systematic Literature Review of Automatic Database Schema

- Design and Tuning,” *ACM Comput. Surv.*, vol. 56, no. 11, Jun. 2024, doi: 10.1145/3665323.
- [15] L. Zhang and M. A. Babar, “Automatic Configuration Tuning on Cloud Database: A Survey,” Apr. 2024, [Online]. Available: <http://arxiv.org/abs/2404.06043>
- [16] J. Lao *et al.*, “GPTuner: A Manual-Reading Database Tuning System via GPT-Guided Bayesian Optimization,” in *Proceedings of the VLDB Endowment*, VLDB Endowment, 2024, pp. 1939–1952. doi: 10.14778/3659437.3659449.
- [17] Google AI for Developers, “Text generation - generateContent API,” Google AI for Developers. Accessed: May 20, 2026. [Online]. Available: <https://ai.google.dev/gemini-api/docs/text-generation>
- [18] Google AI for Developers, “Structured outputs - generateContent API,” Google AI for Developers. Accessed: May 20, 2026. [Online]. Available: <https://ai.google.dev/gemini-api/docs/structured-output?example=recipe>
- [19] E. Jo, N. Lee, and G. Kim, “Database Normalization via Dual-LLM Self-Refinement,” *ACM Conference Proceedings*, vol. 2657, pp. 1–9, Aug. 2025, [Online]. Available: <http://arxiv.org/abs/2508.17693>
- [20] M. H. Erol, X. Hao, F. Bianchi, C. Greco, J. Tagliabue, and J. Zou, “Making Databases Faster with LLM Evolutionary Sampling,” Feb. 2026, [Online]. Available: <http://arxiv.org/abs/2602.10387>

- [21] Y. Tian *et al.*, “CodeHalu: Investigating Code Hallucinations in LLMs via Execution-based Verification,” Jan. 2025, [Online]. Available: <http://arxiv.org/abs/2405.00253>
- [22] L. Chen *et al.*, “SecCodeBench-V2 Technical Report,” Feb. 2026, [Online]. Available: <http://arxiv.org/abs/2602.15485>
- [23] Docker Inc, “Docker overview,” Docker Inc. Accessed: May 20, 2026. [Online]. Available: <https://docs.docker.com/get-started/docker-overview>
- [24] Docker Inc, “Enhanced Container Isolation,” Docker Inc. Accessed: May 20, 2026. [Online]. Available: <https://docs.docker.com/enterprise/security/hardened-desktop/enhanced-container-isolation>
- [25] A. Baihaqi, H. R. Wicaksono, M. Al-Habsie, M. R. Habibie, and Wahyunengsih, “DATABASE MANAGEMENT SYSTEM (DBMS) UNTUK KINERJA TRANSAKSI BISNIS DALAM E-COMMERCE BAGI PENGGUNA DI INDONESIA,” *JURSISTEKNI (Jurnal Sistem Informasi dan Teknologi Informasi)*, vol. 6, no. 3, pp. 568–579, Sep. 2024.
- [26] S. Suryadi, “IMPLEMENTASI NORMALISASI DALAM PERANCANGAN DATABASE RELATIONAL,” *Jurnal Teknik Informatika*, vol. 3, no. 2, 2019, Accessed: Feb. 24, 2026. [Online]. Available: <https://ejurnal.univalabuhanbatu.ac.id/index.php/unet/article/download/7/5>

- [27] M. Pourreza and D. Rafiei, “DIN-SQL: Decomposed In-Context Learning of Text-to-SQL with Self-Correction,” Nov. 2023, [Online]. Available: <http://arxiv.org/abs/2304.11015>
- [28] S. Xue *et al.*, “DB-GPT: Empowering Database Interactions with Private Large Language Models,” Jan. 2024, [Online]. Available: <http://arxiv.org/abs/2312.17449>
- [29] S. Dwiyatno, E. Rakhmat, and O. Gustiawan, “IMPLEMENTASI VIRTUALISASI SERVER BERBASIS DOCKER CONTAINER,” *Jurnal PROSISKO*, vol. 7, no. 2, pp. 165–175, 2020.
- [30] W. Pratama and F. A. Ahda, “Inovasi Agen AI Dalam Sistem Pencatatan Struk Digital Otomatis Berbasis n8n,” *JURNAL FASILKOM*, vol. 15, no. 3, pp. 551–561, Dec. 2025.
- [31] Y. Xu, F. Lin, J. Yang, Tse-Hsun, Chen, and N. Tsantalis, “MANTRA: Enhancing Automated Method-Level Refactoring with Contextual RAG and Multi-Agent LLM Collaboration,” *Proceedings of ACM Conference*, vol. 2657, pp. 1–12, Mar. 2025, [Online]. Available: <http://arxiv.org/abs/2503.14340>
- [32] IEEE Computer Society, *Guide to the Software Engineering Body of Knowledge v4.0a*. 2025.
- [33] A. ALazzawi, Q. M. Yas, and B. Rahmatullah, “A Comprehensive Review of Software Development Life Cycle methodologies: Pros, Cons, and Future Directions,” *Iraqi Journal for Computer Science and Mathematics*, vol. 4, no. 4, pp. 173–190, 2023, doi: 10.52866/ijcsm.2023.04.04.014.

- [34] M. Souppaya, K. Scarfone, and D. Dodson, “Secure Software Development Framework (SSDF) version 1.1 ;,” Feb. 2022. doi: 10.6028/NIST.SP.800-218.
- [35] SEBoK Editorial Board, *Guide to the Systems Engineering Body of Knowledge (SEBoK)*, Version 2.14. Hoboken, NJ, USA: The Trustees of the Stevens Institute of Technology, 2026.
- [36] A. Watt and A. Nelson Eng Bccampus Victoria, “Database Design-2nd Edition,” 2014.
- [37] Oracle Corporation, “MySQL 8.4 Reference Manual Including MySQL NDB Cluster 8.4.”
- [38] Oracle Corporation, “MySQL 8.4 Reference Manual: Alternative Storage Engines,” Oracle. Accessed: May 25, 2026. [Online]. Available: <https://dev.mysql.com/doc/refman/8.4/en/storage-engines.html>
- [39] A. Watt and N. Eng, “Database Design-2nd Edition.” [Online]. Available: <http://open.bccampus.ca>
- [40] D. R. Yunianto, Yudistira Eka Putra, and Cahya Rahmad, “Comparison of Relational Database Modeling Performance based on Number of Normalized Entities,” *SMARTICS Journal*, vol. 9, no. 1, pp. 42–48, Apr. 2023, doi: 10.21067/smartics.v9i1.8390.
- [41] S. Maesaroh, H. Gunawan, A. Lestari, M. S. A. Tsaurie, and M. Fauji, “Query Optimization in MySQL Database Using Index,” 2022. [Online]. Available: <https://iiast-journal.org/ijcitsm/index.php/IJCITSM/article/view/84>

- [42] W. X. Zhao *et al.*, “A Survey of Large Language Models,” Mar. 2026, [Online]. Available: <http://arxiv.org/abs/2303.18223>
- [43] Gemini Team *et al.*, “Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context,” Dec. 2024, [Online]. Available: <http://arxiv.org/abs/2403.05530>
- [44] Google, “Gemini API Documentation,” Google. Accessed: May 25, 2026. [Online]. Available: <https://ai.google.dev/gemini-api/docs>
- [45] Google, “Google AI for Developers, ‘Structured Outputs,’” Google. Accessed: May 25, 2026. [Online]. Available: <https://ai.google.dev/gemini-api/docs/structured-output?example=recipe>
- [46] Kent Beck and Cynthia Andres, “Praise for Extreme Programming Explained, Second Edition,” Boston, 2005.
- [47] J. Erickson, K. Lyytinen, and K. Siau, “Agile modeling, agile software development, and extreme programming: The state of research,” 2005. doi: 10.4018/jdm.2005100105.
- [48] Dwi Yana Ayu Andini, Fahlul Rizki, and Aviv Fitria Yulia, “Agile Digital Transformation in Local Government: An Extreme Programming Approach to Public Service Mall Applications,” Jun. 2025, doi: 10.51519/journalisi.v7i2.1083.