

BAB I

PENDAHULUAN

1.1 Latar Belakang

Research, Innovation & Sustainability (RIS) Universitas Multimedia Nusantara merupakan kanal digital yang memuat informasi penelitian, pengabdian kepada masyarakat, publikasi, inovasi, kolaborasi, agenda, berita, serta pemetaan kegiatan akademik terhadap Sustainable Development Goals (SDGs). Kanal ini berperan sebagai media diseminasi dan sistem informasi institusional yang membantu menunjukkan produktivitas akademik UMN.

Data publik yang ditampilkan pada halaman RIS UMN menunjukkan 1684 *Research & Community Service*, 19 *Publication*, dan 2 *Network*. Halaman tersebut juga menampilkan cakupan modul Penelitian, PKM, Inovasi, Tentang Kami, serta subfitur seperti Peneliti Kami, Info Hibah Penelitian, Penelitian per Area Studi, Ringkasan PKM, Peta Jalan, MBKM and KKN, Info PKM, Katalog Inovasi, Program Inovasi, dan Proyek Inovasi. Angka dan cakupan fitur ini memperlihatkan bahwa RIS mengelola informasi organisasi yang beragam, walaupun data jumlah pengguna aktif, jumlah transaksi internal, dan waktu respons aktual tidak tersedia secara publik.

Berdasarkan data internal yang diperoleh dalam bentuk rekap memo cek jurnal dan memo cek conference, aktivitas publikasi di lingkungan Universitas Multimedia Nusantara menunjukkan volume data yang cukup besar dan beragam. Data memo cek jurnal mencatat 1.469 data cek jurnal pada periode 2023–2025, yang terdiri atas 588 data pada tahun 2023, 444 data pada tahun 2024, dan 437 data pada tahun 2025. Selain itu, data memo cek conference mencatat 320 data konferensi atau prosiding yang melibatkan 160 dosen atau pengajar unik dari 25 program studi. Data tersebut menunjukkan bahwa pengelolaan informasi riset dan publikasi tidak hanya berkaitan dengan daftar publikasi, tetapi juga melibatkan data

dosen, program studi, jurnal, konferensi, indeks publikasi, ISSN, URL, serta status verifikasi publikasi.

Data RIS tidak hanya berkaitan dengan publikasi, tetapi juga penelitian, pengabdian masyarakat, peneliti, fakultas, program studi, kolaborasi, inovasi, berita, agenda, dan *SDGs*. Sebagian data tersebut memiliki relasi yang jelas, misalnya hubungan peneliti, program studi, publikasi, dan *SDGs*. Sebagian lainnya dapat bersifat lebih fleksibel, seperti metadata publikasi, kata kunci, konten berita, agenda, dan informasi kolaborasi.

Kompleksitas struktur data tersebut menyebabkan proses pemilihan DBMS untuk *Website* RIS UMN menjadi tidak sederhana. *Website* RIS tidak hanya berfungsi sebagai media penyimpanan data, tetapi juga harus mampu mendukung berbagai operasi yang umum dilakukan pada aplikasi *web*, seperti menampilkan daftar publikasi, membaca detail data, melakukan pencarian, *filtering* berdasarkan tahun dan *SDGs*, *pagination*, serta menampilkan data yang saling berelasi, seperti peneliti dan kategori *SDGs*. Setiap DBMS memiliki karakteristik yang berbeda dalam menangani relasi data, metadata, pencarian, *filtering*, *pagination*, maupun konsistensi transaksi. Oleh karena itu, pemilihan DBMS yang hanya didasarkan pada asumsi umum atau popularitas teknologi belum tentu menghasilkan solusi yang sesuai dengan karakteristik *workload Website* RIS. Kondisi tersebut berpotensi memengaruhi waktu respons, stabilitas akses, serta kelayakan implementasi sistem. Dengan demikian, diperlukan *benchmark* sebagai dasar evaluasi yang objektif untuk membandingkan performa beberapa DBMS pada skenario yang merepresentasikan kebutuhan *Website* RIS UMN [1], [2], [4].

Pemilihan DBMS untuk *Website* RIS tidak dapat dilakukan hanya berdasarkan klaim umum bahwa suatu DBMS memiliki performa yang lebih baik dibandingkan DBMS lainnya. Penelitian terdahulu mengenai database benchmarking menunjukkan bahwa performa DBMS dipengaruhi oleh berbagai faktor, seperti karakteristik *workload*, ukuran dataset, konfigurasi sistem, indeks, pola *query*, metrik evaluasi, serta lingkungan pengujian [1], [6], [17]. Selain itu, kajian mengenai SQL dan NoSQL menunjukkan bahwa masing-masing model

basis data memiliki karakteristik dan keunggulan yang berbeda. DBMS relasional umumnya lebih sesuai untuk mengelola data yang memiliki relasi terstruktur, transaksi, *constraint*, dan *query* yang kompleks, sedangkan *document database* lebih sesuai untuk data yang bersifat semi-terstruktur karena menawarkan fleksibilitas skema dan pemodelan data [2], [10], [12], [15]. Oleh karena itu, pemilihan DBMS untuk Website RIS UMN perlu didasarkan pada hasil *benchmark* yang sesuai dengan karakteristik workload sistem, bukan semata-mata mengacu pada hasil penelitian sebelumnya [1], [2], [5], [6].

Pada aplikasi berbasis *web*, pengguna tidak berinteraksi secara langsung dengan *database*, melainkan melalui lapisan aplikasi. Dengan demikian, *direct database benchmark* yang hanya mengukur performa *query* atau koneksi langsung ke DBMS belum sepenuhnya merepresentasikan mekanisme akses pada Website RIS. Atas dasar tersebut, penelitian ini menggunakan pendekatan *API-layer benchmark* karena proses pengujian dilakukan melalui *endpoint* HTTP sehingga lebih mendekati pola akses aplikasi *web*. Pengujian dilaksanakan menggunakan Grafana K6 sebagai *load generator*, *Backend API* berbasis Node.js/TypeScript sebagai lapisan aplikasi, serta *database adapter layer* yang bertugas meneruskan operasi yang sama ke masing-masing DBMS. Melalui arsitektur tersebut, setiap DBMS dievaluasi menggunakan *endpoint* dan skenario pengujian yang setara, sedangkan perbedaan implementasi *query* maupun mekanisme akses data ditangani oleh *database adapter layer*.

Pendekatan *API-layer benchmark* dipilih karena permasalahan dalam pemilihan DBMS pada Website RIS UMN tidak hanya berkaitan dengan kompleksitas data yang dikelola, tetapi juga belum adanya dasar empiris yang menunjukkan DBMS mana yang paling sesuai untuk mendukung karakteristik kebutuhan sistem. Website RIS UMN memiliki kebutuhan pengelolaan data publikasi, peneliti, fakultas, program studi, SDGs, berita, agenda, dan kolaborasi, serta memerlukan operasi seperti CRUD, pencarian, *filter*, *pagination*, dan *related data*. Pola kebutuhan tersebut dapat memberikan beban yang berbeda pada setiap DBMS karena masing-masing memiliki model penyimpanan, mekanisme *indexing*, relasi data, dan strategi *query* yang berbeda.

Apabila pemilihan DBMS dilakukan hanya berdasarkan popularitas atau asumsi umum, terdapat risiko bahwa DBMS yang dipilih tidak sesuai dengan karakteristik *workload Website RIS UMN*. DBMS relasional dapat lebih kuat dalam menjaga struktur relasi dan konsistensi data, tetapi beberapa skenario pembacaan dapat melibatkan join antar tabel. Sebaliknya, *document database* dapat lebih fleksibel dalam menyimpan data bertingkat, tetapi dapat menimbulkan duplikasi data dan membutuhkan strategi pembaruan yang berbeda. Oleh karena itu, diperlukan *benchmark* yang mampu merepresentasikan pola akses tersebut agar perbandingan performa antar-DBMS dilakukan pada kondisi yang sesuai dengan karakteristik Website RIS UMN.

Lima DBMS yang digunakan dalam penelitian ini dipilih untuk mewakili dua model basis data yang relevan dengan karakteristik data Website RIS, yaitu *relational database* dan *document database*. Microsoft SQL Server, PostgreSQL, dan MariaDB dipilih sebagai representasi DBMS relasional karena sesuai untuk mengelola data yang memiliki relasi eksplisit, seperti publikasi, peneliti, program studi, fakultas, dan SDGs. Selain itu, PostgreSQL juga banyak digunakan sebagai pembanding pada berbagai penelitian mengenai performa DBMS relasional maupun perbandingan dengan MongoDB [11], [12], [15]. Sementara itu, MongoDB dan Firebase Firestore Emulator dipilih sebagai representasi *document database* karena model dokumen lebih sesuai untuk mengelola data yang memiliki struktur fleksibel, seperti metadata, kata kunci, berita, agenda, kolaborasi, dan dokumen [10], [12], [15]. Pemilihan kelima DBMS tersebut tidak dimaksudkan untuk membuktikan bahwa salah satu model basis data selalu lebih unggul dibandingkan model lainnya, melainkan untuk membandingkan performanya pada *workload* yang merepresentasikan karakteristik Website RIS UMN.

Berdasarkan uraian tersebut, penelitian terdahulu telah memberikan landasan penting dalam kajian perbandingan performa DBMS. Taipalus menjelaskan bahwa hasil *benchmark* DBMS sangat dipengaruhi oleh *workload*, *dataset*, konfigurasi, metrik, dan lingkungan pengujian. Oleh karena itu, hasil *benchmark* tidak dapat digeneralisasi tanpa mempertimbangkan konteks pelaksanaannya [1]. Kajian mengenai SQL dan NoSQL juga menunjukkan bahwa

model basis data relasional dan NoSQL memiliki perbedaan dalam aspek arsitektur, skalabilitas, konsistensi, dan kesesuaian terhadap karakteristik data [2]. Selain itu, sejumlah penelitian telah mengevaluasi performa MongoDB dan basis data dokumen lainnya menggunakan *workload benchmark* tertentu [10], serta membandingkan PostgreSQL dan MySQL melalui skenario operasi *select* dan *insert* [11].

Penelitian lain yang membandingkan MongoDB dan PostgreSQL menunjukkan bahwa perbedaan model data dan jenis operasi dapat memengaruhi performa antara *relational database* dan *document database* [12], [15]. Di sisi lain, prinsip evaluasi performa yang reproduktibel menekankan pentingnya penjelasan yang rinci mengenai *workload*, konfigurasi eksperimen, metrik, dan lingkungan pengujian agar hasil *benchmark* dapat dipahami dan ditafsirkan secara tepat [5], [6], [17].

Meskipun demikian, penelitian-penelitian terdahulu belum secara langsung menjawab kebutuhan pemilihan DBMS untuk Website RIS UMN. Sebagian penelitian masih menggunakan *workload* umum atau melakukan pengujian secara langsung pada basis data, sehingga belum merepresentasikan pola akses aplikasi web yang berlangsung melalui *endpoint* API. Pada aplikasi berbasis REST API, proses pengujian juga perlu mempertimbangkan lapisan HTTP, *backend*, serta mekanisme pemrosesan *request* dan *response* [7], [8].

Selain itu, penelitian terdahulu belum secara khusus mengaitkan evaluasi DBMS dengan karakteristik data pada Website RIS UMN. Data tersebut mencakup publikasi, peneliti, fakultas, program studi, SDGs, berita, agenda, dan kolaborasi, serta melibatkan berbagai kebutuhan operasi, seperti CRUD, pencarian, penyaringan, *pagination*, dan pengambilan data yang saling berelasi. Hasil *benchmark* terdahulu juga belum dapat digunakan secara langsung sebagai dasar pemilihan DBMS untuk Website RIS UMN karena adanya perbedaan objek penelitian, *workload*, *dataset*, konfigurasi, dan lingkungan pengujian [1], [17].

Berdasarkan kondisi tersebut, kesenjangan penelitian terletak pada belum tersedianya evaluasi empiris yang membandingkan beberapa DBMS menggunakan

workload yang menyerupai pola akses RIS melalui pendekatan *API-layer benchmark*. Evaluasi tersebut diperlukan untuk mendukung proses pemilihan DBMS pada pengembangan Website RIS UMN. Pemilihan DBMS tidak dapat hanya didasarkan pada popularitas teknologi maupun hasil *benchmark* yang bersifat umum. Website RIS UMN memiliki karakteristik data yang memadukan relasi terstruktur dan *metadata* semi-terstruktur, sehingga setiap DBMS berpotensi menunjukkan performa yang berbeda ketika menjalankan operasi penambahan data, pembacaan daftar, pembacaan detail, pembaruan data, penghapusan data, penyaringan berdasarkan tahun, penyaringan berdasarkan SDGs, pencarian kata kunci, pembacaan data terkait, dan *pagination*.

Oleh karena itu, penelitian ini dilakukan untuk menyediakan dasar empiris dalam membandingkan Microsoft SQL Server, PostgreSQL, MariaDB, MongoDB, dan Firebase Firestore Emulator melalui skenario *API-layer benchmark* yang sama. Penelitian ini tidak bertujuan mengembangkan teori atau metode baru dalam bidang *database benchmarking*, melainkan menerapkan prinsip *benchmark* empiris untuk menghasilkan rekomendasi pemilihan DBMS yang sesuai dengan konteks Website RIS UMN.

Hasil pengujian dianalisis menggunakan metrik *average response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate*. Hasil tersebut kemudian diinterpretasikan dengan mempertimbangkan konteks Website RIS UMN yang menggunakan Laravel dan memiliki struktur data yang dominan relasional.

Kontribusi penelitian ini tidak diarahkan pada pengembangan teori baru dalam bidang *database benchmarking*, melainkan pada penerapan metode *benchmark* yang disesuaikan dengan karakteristik dan kebutuhan Website RIS UMN. Penelitian ini menyusun *workload* berbasis *endpoint API* yang menyerupai pola akses pada RIS, sehingga pengujian tidak hanya dilakukan melalui *query* langsung ke basis data, tetapi juga mencakup alur pemrosesan melalui *backend* dan *database adapter* sebagaimana terjadi pada aplikasi web. Selain itu, karakteristik data RIS yang terdiri atas data relasional terstruktur dan *metadata* semi-terstruktur dipetakan ke dalam lima DBMS yang mewakili dua model basis data, yaitu basis

data relasional dan basis data dokumen. Hasil pengujian selanjutnya digunakan untuk menyusun rekomendasi yang mempertimbangkan tidak hanya performa masing-masing DBMS, tetapi juga kelayakan implementasinya pada Website RIS UMN.

Untuk menjawab *gap* tersebut, penelitian ini merancang *API-layer benchmark* yang terdiri atas sepuluh skenario pengujian yang merepresentasikan operasi utama pada Website RIS, yaitu *insert data*, *read list*, *read detail*, *update data*, *delete data*, *filter by year*, *filter by SDGs*, *search keyword*, *read related data*, dan *pagination*. Hasil benchmark dianalisis menggunakan metrik *average response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate*. Selanjutnya, hasil pengujian diinterpretasikan dalam konteks Website RIS UMN yang menggunakan Laravel dan memiliki struktur data relasional. Interpretasi tersebut digunakan sebagai konteks pembahasan rekomendasi DBMS, bukan sebagai faktor kuantitatif dalam penentuan ranking.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, rumusan masalah dalam penelitian ini difokuskan pada performa API-layer benchmark, perbandingan hasil pengujian antar-DBMS, dan penentuan kandidat DBMS yang paling sesuai untuk pengembangan Website RIS UMN berdasarkan metrik yang diuji dalam penelitian ini.

1. Bagaimana performa Microsoft *SQL Server*, *PostgreSQL*, *MariaDB*, *MongoDB*, dan *Firebase Firestore Emulator* pada skenario *API-layer benchmark* yang merepresentasikan kebutuhan Website RIS UMN?
2. Bagaimana perbandingan hasil *benchmark* antar-DBMS berdasarkan *average response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate*?
3. DBMS mana yang menjadi kandidat paling sesuai untuk pengembangan Website RIS UMN berdasarkan hasil API-layer benchmark, khususnya *average response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate*, dengan tetap memperhatikan batasan bahwa aspek biaya,

keamanan, maintainability, skalabilitas, dan kesiapan produksi tidak diuji secara kuantitatif dalam penelitian ini?

1.3 Batasan Masalah

Agar penelitian lebih terarah dan hasil pengujian dapat dianalisis secara konsisten, batasan masalah dalam penelitian ini adalah sebagai berikut:

1. Penelitian ini hanya membandingkan lima *Database Management System* (DBMS), yaitu Microsoft SQL Server, PostgreSQL, MariaDB, MongoDB, dan Firebase Firestore Emulator.
2. Pengujian Firebase Firestore pada penelitian ini dilakukan menggunakan Firestore Emulator lokal, bukan Firebase Firestore Cloud pada lingkungan produksi. Oleh karena itu, hasil pengujian Firestore tidak digunakan untuk menyimpulkan performa layanan Firestore Cloud secara langsung.
3. *Benchmark* dilakukan melalui *endpoint API* pada *backend* Node.js/TypeScript, bukan melalui koneksi langsung dari Grafana K6 ke masing-masing *database*. Dengan demikian, hasil penelitian merepresentasikan performa end-to-end *API* pada implementasi *backend* dan *database adapter* yang digunakan.
4. Hasil *benchmark* pada penelitian ini tidak diposisikan sebagai pengukuran murni terhadap mesin *database*, karena waktu respons yang diperoleh juga mencakup overhead dari proses *backend*, routing, serialisasi data, validasi respons, serta implementasi *adapter* pada masing-masing DBMS.
5. Konfigurasi *benchmark* utama dibatasi pada 10 *virtual users*, durasi pengujian 30 detik, 3 iterasi, 10 skenario *benchmark*, serta dataset sintesis yang dirancang untuk merepresentasikan kebutuhan data *website* RIS UMN.
6. Dataset yang digunakan dalam penelitian ini merupakan dataset sintesis atau dummy, bukan data produksi dari *website* RIS UMN. Dataset sintesis digunakan untuk menjaga keamanan data, memudahkan replikasi pengujian, serta memastikan kondisi pengujian dapat dikontrol secara konsisten.

7. Skenario *benchmark* yang digunakan dalam penelitian ini mencakup operasi insert data, read list, read detail by ID, update data, delete data, filter by year, filter by SDGs, search keyword, read related data, dan pagination. Oleh karena itu, hasil penelitian tidak dimaksudkan untuk menggambarkan seluruh kemungkinan *workload* pada setiap *database*.
8. Metrik evaluasi utama yang digunakan dalam penelitian ini mencakup *average response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate* berdasarkan metrik HTTP dari Grafana K6.
9. Kolom *db_time_avg_ms* dan *db_time_p95_ms* tersedia pada format CSV hasil *benchmark*, tetapi nilai pada data akhir yang dianalisis tidak terisi. Oleh karena itu, penelitian ini tidak menganalisis waktu eksekusi internal *database* secara langsung dan hanya berfokus pada metrik HTTP level yang dihasilkan oleh Grafana K6.
10. Penelitian ini tidak membahas aspek biaya operasional, keamanan tingkat lanjut, high availability, backup and recovery, resource utilization secara mendalam, maupun deployment pada lingkungan produksi.
11. Hasil penelitian ini bersifat kontekstual terhadap konfigurasi pengujian, dataset sintetis, implementasi *API layer*, *database adapter*, perangkat lokal, serta versi perangkat lunak yang digunakan. Dengan demikian, hasil *benchmark* tidak dimaksudkan sebagai ranking universal untuk seluruh penggunaan Microsoft SQL Server, PostgreSQL, MariaDB, MongoDB, dan Firebase Firestore.

1.4 Tujuan & Manfaat Penelitian

1.4.1 Tujuan Penelitian

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengidentifikasi karakteristik kebutuhan data dan pola akses utama Website RIS UMN sebagai dasar penyusunan skenario *benchmark*.
2. Merancang *API-layer benchmark* yang merepresentasikan operasi utama Website RIS UMN, yaitu *insert data*, *read list*, *read detail*, *update data*,

delete data, filter by year, filter by SDGs, search keyword, read related data, dan pagination.

3. Mengukur performa Microsoft SQL Server, PostgreSQL, MariaDB, MongoDB, dan Firebase Firestore Emulator pada skenario *API-layer benchmark* yang sama berdasarkan *average response time, P95 latency, throughput, error rate, dan checks rate.*
4. Membandingkan hasil *benchmark* antar-DBMS untuk mengetahui perbedaan performa pada *workload* yang merepresentasikan kebutuhan *Website RIS UMN.*
5. Menentukan kandidat DBMS yang paling sesuai untuk kebutuhan pengembangan *Website RIS UMN* berdasarkan hasil *API-layer benchmark*, dengan tetap membatasi kesimpulan pada metrik yang benar-benar diuji dalam penelitian.

1.4.2 Manfaat Penelitian

1.4.2.1 Manfaat Akademis

Secara akademis, penelitian ini memperkaya kajian benchmark DBMS untuk sistem informasi institusional berbasis web yang menggabungkan data relasional dan semi-terstruktur. Kontribusi utama penelitian bukan hanya pada perbandingan DBMS, tetapi juga pada penggunaan API-layer benchmark sebagai pendekatan yang lebih dekat dengan alur akses aplikasi web dibandingkan benchmark query langsung.

Penelitian ini juga menyediakan contoh pengukuran yang transparan dengan skenario, dataset sintetis, konfigurasi K6, output CSV, dan visualisasi yang terdokumentasi. Dengan demikian, penelitian dapat menjadi rujukan bagi studi berikutnya yang ingin mengevaluasi DBMS pada konteks sistem informasi kampus, repository penelitian, atau kanal publikasi institusional.

1.4.2.2 Manfaat Praktis

Secara praktis, hasil penelitian ini dapat digunakan oleh tim pengembang dan pengelola Website RIS UMN sebagai bahan pertimbangan teknis dalam memilih DBMS yang sesuai dengan kebutuhan sistem. Tim pengembang dapat

memanfaatkan hasil benchmark untuk memahami konsekuensi pemilihan DBMS terhadap *response time*, *P95 latency*, *throughput*, *error rate*, dan *checks rate* pada skenario *API-layer* yang merepresentasikan operasi utama *Website RIS*. Selain itu, hasil penelitian ini juga dapat menjadi masukan bagi pihak yang terlibat dalam pengambilan keputusan teknologi, seperti pengelola sistem atau unit terkait di UMN, sebelum menentukan apakah DBMS tertentu layak dipertimbangkan untuk pengembangan, optimasi, atau evaluasi lanjutan *Website RIS*. Namun, hasil *benchmark* ini tetap perlu dibaca bersama faktor implementasi nyata, seperti kompatibilitas Laravel, struktur SQL yang sudah ada, kebutuhan migrasi data, biaya, keamanan, skalabilitas, dan kemampuan pemeliharaan

Manfaat praktis lain adalah tersedianya rancangan benchmark yang dapat digunakan kembali oleh pengembang sistem atau peneliti berikutnya. Skenario dan pipeline yang digunakan dalam penelitian ini dapat diperluas untuk pengujian dataset lebih besar, VUs lebih tinggi, durasi lebih panjang, resource monitoring, Firestore Cloud, instrumentasi database time, dan integrasi langsung dengan Laravel sebelum keputusan produksi dibuat.

1.5 Sistematika Penulisan

Penulisan laporan penelitian ini dibagi menjadi beberapa bab sebagai berikut:

- 1 BAB I membahas pendahuluan penelitian yang meliputi latar belakang RIS sebagai sistem informasi institusional, karakteristik data, masalah pemilihan DBMS, relevansi *API-layer benchmark*, gap penelitian, rumusan masalah, batasan masalah, tujuan, manfaat, definisi operasional basis data optimal, dan sistematika penulisan.
- 2 BAB II menyajikan landasan teori yang mencakup penelitian terdahulu, DBMS, RDBMS, NoSQL, SQL dan NoSQL pada sistem informasi web, karakteristik SQL Server, PostgreSQL, MariaDB, MongoDB, Firebase Firestore dan Firestore Emulator, Laravel dan *backend API*, *performance testing*, *load testing*, *API benchmarking*, Grafana K6, Apache JMeter, Locust, metrik evaluasi, *synthetic* dataset, research gap, dan kerangka pemikiran.

- 3 BAB III menjelaskan metodologi penelitian, mulai dari tahapan penelitian, objek penelitian, pendekatan penelitian, arsitektur *benchmark*, rancangan dataset sintetis, pemetaan model data, perancangan *Backend API* dan *database adapter*, skenario *benchmark*, konfigurasi K6, lingkungan pengujian, prosedur *health check*, *reset*, *seed*, eksekusi *benchmark*, format output, pengolahan CSV, pembuatan grafik, teknik pengumpulan data, teknik analisis data, validitas, dan keterbatasan metodologi.
- 4 BAB IV menyajikan analisis dan hasil penelitian. Bab ini memuat analisa masalah, analisa proses bisnis, user requirements, perancangan sistem, perancangan *database*, dokumentasi testing dan implementasi, bukti screenshot, hasil ranking DBMS, analisis *average response time*, *P95 latency*, *throughput*, *error rate*, *checks rate*, analisis S1 sampai S10, perbandingan SQL dan NoSQL, implikasi terhadap RIS aktual berbasis Laravel dan SQL, keterbatasan hasil, serta rekomendasi DBMS.
- 5 BAB V berisi simpulan dan saran. Simpulan menjawab rumusan masalah, sedangkan saran berisi arahan pengembangan penelitian berikutnya seperti dataset lebih besar, VUs lebih tinggi, durasi lebih panjang, production-like environment, Firestore Cloud, instrumentasi waktu *database* internal, resource monitoring, Laravel integration testing, indexing, cost, security, backup, high availability, dan maintainability evaluation sistem.