

SKRIPSI

**PERBANDINGAN KINERJA JENIS-JENIS BASIS DATA
NOSQL DALAM MANAJEMEN DATA BIOMEDIK**

Disusun dan diajukan oleh:

**HERRYTS TIMISELA
D121171701**



**DEPARTEMEN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS HASANUDDIN
GOWA
2024**

LEMBAR PENGESAHAN SKRIPSI

**PERBANDINGAN KINERJA JENIS-JENIS BASIS DATA NOSQL
DALAM MANAJEMEN DATA BIOMEDIK**


Disusun dan diajukan oleh

**HERRYTS TIMISELA
D121171701**

Telah dipertahankan dihadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian
Studi Program Sarjana Program Studi Teknik Informatika
Fakultas Teknik Universitas Hasanuddin
Pada tanggal 31 Juli 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama,


Mukarramah Yusuf, B.Sc., M.Sc., Ph.D.
NIP. 198310082012122003

Ketua Program Studi,


Prof. Dr. H. Indrabayu, ST., MT., M.Bus.Sys., IPM, ASEAN. Eng.
NIP. 197507162002121004

PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama : Herryts Timisela

NIM : D121171701

Program Studi : Departemen Teknik Informatika

Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Perbandingan Kinerja Jenis-Jenis Basis Data Nosql Dalam Manajemen Data Biomedik

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 31 Juli 2024

Yang Menyatakan



Herryts Timisela

ABSTRAK

HERRYTS TIMISELA. *Perbandingan Kinerja Jenis-Jenis Basis Data Nosql Dalam Manajemen Data Biomedik* (dibimbing oleh Mukarramah Yusuf)

Penelitian ini bertujuan untuk menganalisis dan membandingkan kinerja *database* MongoDB dan *Elasticsearch* pada *framework* Laravel. Penelitian ini dimulai dari tahap analisis kebutuhan, pengukuran performa, hingga analisis performa. Berdasarkan data pengujian performa, MongoDB secara konsisten menunjukkan *response time* yang lebih cepat dibandingkan *Elasticsearch* dalam sebagian besar skenario, termasuk melihat data, menambah data, mengedit data, dan menghapus data. Sebagai contoh, untuk melihat 10 list data (TC-01), MongoDB memiliki rata-rata *response time* 0.0020 detik, sedangkan *Elasticsearch* memerlukan 0.0187 detik. Namun, dalam skenario *filter* data (TC-04), *Elasticsearch* unggul dengan rata-rata *response time* 0.019 detik dibandingkan dengan 3.1881 detik pada MongoDB. Hasil penelitian menunjukkan bahwa MongoDB lebih efisien untuk operasi data umum yang memerlukan *response time* cepat, sementara *Elasticsearch* lebih efisien dalam menangani operasi pencarian dan analisis data yang kompleks. Secara keseluruhan, MongoDB menunjukkan performa yang lebih baik dalam operasi *Create*, *Read*, *Update*, dan *Delete* data, sedangkan *Elasticsearch* lebih unggul dalam operasi pencarian dan analisis data yang memerlukan kompleksitas tinggi.

Kata Kunci: MongoDB, *Elasticsearch*, *Framework* Laravel, kinerja *database*, *response time*, operasi data, pencarian data, analisis data

ABSTRACT

HERRYTS TIMISELA. *Perbandingan Kinerja Jenis-Jenis Basis Data Nosql Dalam Manajemen Data Biomedik* (supervised by Mukarramah Yusuf)

This study aims to analyze and compare the performance of MongoDB and Elasticsearch databases on the Laravel framework. The research starts from the requirement analysis stage, performance measurement, to performance analysis. Based on performance test data, MongoDB consistently shows faster response times compared to Elasticsearch in most scenarios, including viewing data, adding data, editing data, and deleting data. For instance, to view a list of 10 data items (TC-01), MongoDB has an average response time of 0.0020 seconds, whereas Elasticsearch requires 0.0187 seconds. However, in the data filtering scenario (TC-04), Elasticsearch excels with an average response time of 0.019 seconds compared to 3.1881 seconds on MongoDB. The results indicate that MongoDB is more efficient for general data operations requiring fast response times, while Elasticsearch is more efficient in handling complex search and data analysis operations. Overall, MongoDB demonstrates better performance in Create, Read, Update, and Delete operations, whereas Elasticsearch excels in search and data analysis operations that require high complexity.

Keywords: MongoDB, Elasticsearch, framework Laravel, database performance, response time, data operations, data search, data analysis

DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI	i
PERNYATAAN KEASLIAN.....	ii
ABSTRAK	iii
ABSTRACT	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR	vi
DAFTAR TABEL	vii
DAFTAR SINGKATAN DAN ARTI SIMBOL	viii
DAFTAR LAMPIRAN.....	ix
KATA PENGANTAR	x
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian	4
1.4 Manfaat Penelitian	5
1.5 Ruang Lingkup.....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Basis Data (<i>Database</i>)	6
2.2 NoSQL	6
2.3 MongoDB.....	9
2.4 <i>Elasticsearch</i>	11
2.5 <i>Framework</i> Laravel.....	15
2.6 Docker	18
2.7 <i>Electroencephalography</i> (EEG).....	20
2.8 Penelitian Terkait	24
BAB III METODE PENELITIAN/PERANCANGAN	33
3.1 Waktu dan Lokasi Penelitian	33
3.2 Instrumen Penelitian	33
3.3 Tahapan Penelitian	33
3.4 Metode Pembuatan Web	35
3.4.1 Analisis Kebutuhan Sistem	37
3.4.2 <i>Use Case</i> Diagram.....	37
3.4.3 Implementasi Sistem	37
3.4.4 Karakteristik Data	38
3.4.5 Skenario Pengujian.....	40
BAB IV HASIL DAN PEMBAHASAN	50
4.1 Hasil	50
4.2 Pembahasan.....	65
BAB V KESIMPILAN DAN SARAN	68
5.1 Kesimpulan	68
5.2 Saran.....	68
DAFTAR PUSTAKA	70

DAFTAR GAMBAR

Gambar 1 Diagram Ilustrasi <i>Query</i> MongoDB	11
Gambar 2 Proses <i>Indexing Single Node</i>	14
Gambar 3 Proses <i>Query Single Node</i>	15
Gambar 4 Arsitektur Docker	18
Gambar 5 Menunjukkan Contoh Data EEG	20
Gambar 6 Tahapan Penelitian	33
Gambar 7 Diagram Model <i>Waterfall</i>	35
Gambar 8 <i>Use Case Diagram</i> Rancangan Sistem	37
Gambar 9 Halaman List Data MongoDB	50
Gambar 10 Halaman List Data <i>Elasticsearch</i>	51
Gambar 11 Halaman <i>Create Data</i>	52
Gambar 12 Halaman Lanjutan <i>Create Data</i>	53
Gambar 13 Halaman Lanjutan Berhasil <i>Create Data</i>	54
Gambar 14 Halaman Detail Data	54
Gambar 15 Halaman Berhasil <i>Delete Data</i>	55
Gambar 16 <i>Query Info</i>	56
Gambar 17 TC-01 (Melihat 10 List Data)	58
Gambar 18 TC-02 (Melihat 1000 List Data)	59
Gambar 19 TC-03 (Melihat 5000 List Data)	60
Gambar 20 TC-04 (Melakukan <i>Filter Data</i>)	61
Gambar 21 TC-05 (Melihat Detail Data)	62
Gambar 22 TC-06 (Menambah Data)	63
Gambar 23 TC-07 (Mengedit Data)	64
Gambar 24 TC-08 (Menghapus Data)	65

DAFTAR TABEL

Tabel 1 <i>Test Case</i> Untuk Pengujian <i>Query Response Time</i>	41
Tabel 2 <i>Query</i> Untuk Melihat 10 List Data	42
Tabel 3 <i>Query</i> Untuk Melihat 1000 List Data	43
Tabel 4 <i>Query</i> Untuk Melihat 5000 List Data	44
Tabel 5 <i>Query</i> Untuk Melakukan <i>Filter</i> Data	45
Tabel 6 <i>Query</i> Untuk Melihat Detail Data	46
Tabel 7 <i>Query</i> Untuk Menambah Data	47
Tabel 8 <i>Query</i> Untuk Mengedit Data	48
Tabel 9 <i>Query</i> Untuk Menghapus Data	49
Tabel 10 Hasil Pengujian <i>Response Time</i> Pada <i>Database</i> MongoDB dan <i>Elasticsearch</i>	57

DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
NoSQL	<i>Not Only SQL</i>
EEG	<i>Electroencephalography</i>
DBEEG	<i>Database Electroencephalography</i>
TC	<i>Test case</i>

DAFTAR LAMPIRAN

Lampiran 1 <i>Screenshot</i> perhitungan rata-rata	73
Lampiran 2 <i>Screenshot</i> data di <i>database</i> MongoDB	73
Lampiran 3 <i>Screenshot</i> data di <i>database Elasticsearch</i>	74
Lampiran 4 Daftar hadir dan berita acara seminar hasil	75
Lampiran 5 Daftar hadir dan berita acara ujian skripsi.....	78
Lampiran 6 Lembar perbaikan skripsi	81

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang oleh karena kasih dan karunia-Nya penulis boleh dapat menyelesaikan penulisan skripsi yang berjudul **“PERBANDINGAN KINERJA JENIS-JENIS BASIS DATA NOSQL DALAM MANAJEMEN DATA BIOMEDIK”** ini dapat diselesaikan dengan baik sebagai salah satu syarat dalam menyelesaikan jenjang Strata-1 pada Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari sepenuhnya bahwa skripsi ini masih jauh dari kesempurnaan karena menyadari segala keterbatasan yang ada. Dalam penulisan skripsi ini penulis menghadapi berbagai kendala dan masalah, namun karena usaha yang maksimal dan kemampuan yang Tuhan berikan kepada penulis serta bantuan dan dukungan dari berbagai pihak, maka penulisan skripsi ini dapat selesai. Oleh karena itu, pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada:

1. Tuhan Yang Maha Esa pencipta alam semesta yang senantiasa memberikan kasih karunia serta hikmatnya kepada penulis.
2. Kedua orang tua terkasih, Alm. Bapak J.C Timisela dan Almh. Ibu Elisabeth Titahena yang selalu menjadi pendoa kepada penulis.
3. Kakak terkasih, Melyssa Timisela yang selalu memberikan motivasi serta semangat kepada penulis.
4. Dosen pembimbing Ibu Mukarramah Yusuf, B.Sc., M.Sc., Ph.D. yang senantiasa menyediakan waktu, tenaga, pikiran, dan perhatian yang luar biasa dalam mengarahkan penulis dalam penyusunan tugas akhir ini.
5. Ketua Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin, Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng., atas ilmu, nasihat, wawasan dan pengalaman yang diberikan kepada penulis.
6. Segenap Dosen dan Staf Departemen Teknik Informatika Fakultas Teknik Universitas Hasanuddin yang telah banyak membantu semasa perkuliahan hingga penyelesaian tugas akhir penulis.

7. Saudara seperjuangan penulis Afiriasi Papua 2017 Unhas yang telah menemani dan mendukung perjalanan penulis sekaligus tempat berbagi keluh kesah selama menjadi mahasiswa di Universitas Hasanuddin.
8. Saudara seperjuangan penulis RECOGN17ER yang telah menemani dan mendukung perjalanan penulis sekaligus tempat berbagi keluh kesah selama menjadi mahasiswa teknik di Departemen Informatika Fakultas Teknik Universitas Hasanuddin.
9. Saudara saya Ikbal, Annas, Ghalib, Ridwan, Dwi, Uga, Kayzar dan Harfiansa yang hadir memberikan motivasi dan semangat kepada penulis selama menyusun tugas akhir.
10. Seluruh pihak yang tidak sempat disebutkan satu persatu yang telah banyak meluangkan tenaga, waktu, dan pikiran selama penyusunan tugas akhir ini.

Penulis berharap semoga Tuhan Yang Maha Esa berkenan membalas segala kebaikan yang telah diterima oleh penulis dari berbagai pihak yang telah membantu, mempermudah penulis dalam mengerjakan tugas akhir ini. Penulis menyadari bahwa masih banyak kekurangan dalam penulisan tugas akhir ini.

Oleh karena itu, penulis menerima segala bentuk masukan, kritik dan saran untuk kesempurnaan tugas akhir ini. Penulis berharap semoga tugas akhir ini dapat bermanfaat bagi semua pihak yang membacanya. Amin.

Gowa, 31 Juli 2024

Penulis,
Herryts Timisela

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam era informasi yang terus berkembang pesat, pengelolaan data biomedis menjadi semakin penting. Data biomedik, seperti data EEG (*Electroencephalography*), memiliki karakteristik yang unik dan kompleks. EEG adalah salah satu jenis data biomedis yang dicatat dari aktivitas listrik otak manusia dan digunakan dalam berbagai aplikasi, mulai dari diagnosis penyakit hingga penelitian neurosains. Namun, pengelolaan data biomedik seperti EEG sering kali menimbulkan tantangan tersendiri karena volume yang besar, kecepatan penyimpanan dan akses yang tinggi, serta kebutuhan akan struktur data yang fleksibel untuk analisis yang lebih baik.

Namun, dengan meningkatnya kompleksitas dan volume data, model basis data relasional tradisional seringkali tidak memadai untuk memenuhi kebutuhan pengelolaan data biomedik yang berskala besar dan dinamis. Sebagai alternatif, sistem basis data *non-relational*, juga dikenal sebagai *Not Only SQL* (NoSQL) menawarkan solusi yang menjanjikan untuk mengatasi tantangan ini. NoSQL memberikan fleksibilitas yang lebih besar dalam struktur data, kemampuan untuk menangani data semi-terstruktur dan tidak terstruktur, dan skalabilitas horizontal yang baik, seringkali diperlukan dalam lingkungan biomedik yang berubah dengan cepat.

Basis data NoSQL telah menjadi pilihan utama dalam berbagai aplikasi, termasuk dalam domain biomedik. Ada empat jenis utama basis data NoSQL, yaitu: basis data berorientasi dokumen, seperti MongoDB dan *Elasticsearch*. Basis data ini memungkinkan penyimpanan data yang fleksibel dalam bentuk dokumen, cocok untuk data biomedik yang sering kali memiliki struktur yang tidak teratur atau berubah-ubah. Selain itu, basis data berorientasi kolom seperti Apache Cassandra juga menjadi pilihan populer karena kemampuannya untuk menyimpan dan mengelola data kolom secara efisien, yang sesuai dengan kebutuhan analisis biomedik yang seringkali memerlukan pengambilan data berbasis kolom. Basis data graf, seperti Neo4j dan ArangoDB, juga menawarkan keunggulan dalam

mengelola dan menganalisis hubungan kompleks antara data, yang sangat penting dalam penelitian biomedik untuk memetakan jaringan interaksi biologis atau jalur penyakit. Basis data kunci-nilai seperti Redis dan Amazon DynamoDB menawarkan kinerja tinggi untuk operasi baca/tulis yang cepat, yang dapat sangat berguna dalam aplikasi biomedik yang memerlukan akses data waktu nyata seperti pemantauan pasien atau *respons* darurat.

Pada beberapa penelitian yang telah dilakukan pada kasus yang sama tentang database utamanya untuk mengelola, menyimpan, dan menganalisis data dalam jumlah yang sangat besar dan bervariasi dengan cepat dan efisien. Berdasarkan tinjauan penelitian mengenai perbandingan dari waktu respon *database* (Gunawan, 2018; Halimi & Sudarmanto, 2021), ternyata kinerja NoSQL lebih baik dibanding *database* relasional. Namun, penelitian yang membandingkan waktu respon antara aplikasi *database* NoSQL MongoDB dan aplikasi *database* NoSQL *Elasticsearch* masih minim dilakukan. Beberapa perusahaan besar yang menerapkan MongoDB dalam pengolahan datanya, seperti SEGA HARD light yang bergerak di bidang *mobile game development*, Medtronic yang merupakan perusahaan penyedia alat-alat medis di Amerika Serikat, EA yang merupakan *developer game* yang bertema olahraga (pada game EA FIFA Online 3 dikembangkan menggunakan MongoDB). Sedangkan beberapa perusahaan di Indonesia yang menerapkan *Elasticsearch* yaitu Tokopedia dan BCA, serta perusahaan enterprise seperti Facebook, Airbus, dan IEEE (Renaldi dkk, 2020). Penelitian ini membandingkan waktu respon pada operasi *Create, Read, Update, Delete* (CRUD) data dari aplikasi database NoSQL *Elasticsearch* dan MongoDB, sehingga dapat diketahui aplikasi database NoSQL manakah yang bekerja lebih cepat. Pengujian pada penelitian ini dilakukan sebanyak 3 kali untuk tiap operasi CRUD data dalam beragam record data (100, 1000, dan 2000), sedangkan waktu respon dari setiap operasi disajikan dalam bentuk tabel dan grafik.

Seperti yang ditunjukkan oleh penelitian dalam jurnal berjudul "*A Deep Learning Approach for Automatic Seizure Detection in Long-Term EEG Signals*" yang dilakukan oleh Abdelhameed & Bayoumi (2021), data EEG memiliki peran penting dalam diagnosis medis, terutama untuk deteksi kejang epilepsi. Penelitian ini memperkenalkan metode baru berbasis *deep learning* yang menggabungkan

convolutional neural networks (CNN) dan *long short-term memory* (LSTM) *networks* untuk deteksi kejang otomatis dalam sinyal EEG jangka panjang. Model yang diusulkan berhasil mencapai akurasi tinggi dalam mengidentifikasi kejadian kejang dalam rekaman EEG kontinu, menunjukkan peningkatan signifikan dalam hal sensitivitas dan spesifisitas dibandingkan dengan model *machine learning* tradisional. Pendekatan ini tidak hanya meningkatkan akurasi diagnosis tetapi juga memungkinkan pemantauan terus-menerus yang dapat memberikan peringatan dini kepada pasien dan penyedia layanan kesehatan.

Kemudian penelitian lain dalam jurnal berjudul "*EEG-Based Brain-Computer Interfaces: A Comprehensive Review of Signal Processing Methods and Clinical Applications*" yang dilakukan oleh Abiri dkk (2019), memberikan tinjauan luas tentang teknik pengolahan sinyal EEG dan aplikasinya dalam konteks klinis. Ulasan ini mencakup berbagai metode untuk ekstraksi fitur, seleksi fitur, dan klasifikasi sinyal EEG, dengan fokus pada implementasinya dalam diagnostik klinis dan neurorehabilitasi. Penelitian ini menyoroti bahwa teknik pengolahan sinyal EEG yang canggih dapat mengubah data mentah menjadi informasi yang berguna untuk aplikasi antarmuka otak-komputer (BCI), termasuk pemulihan pasien stroke dan kontrol perangkat prostetik. Kemajuan dalam pengolahan sinyal EEG ini memfasilitasi aplikasi BCI yang akurat dan real-time, yang berpotensi meningkatkan kualitas hidup pasien dengan gangguan neurologis.

Penelitian berjudul "*Machine Learning Techniques for the Classification of Epileptic Seizures Using EEG Signals*" yang dilakukan oleh Savadkoohi dkk (2020), mengeksplorasi efektivitas berbagai algoritma *machine learning* dalam menganalisis data EEG untuk klasifikasi kejang epilepsi. Studi ini mengevaluasi berbagai algoritma, termasuk *support vector machines* (SVM), *k-nearest neighbors* (k-NN), dan *random forests*, dalam hal kemampuannya untuk secara akurat mengklasifikasikan kejadian kejang dan non-kejang. Temuan penelitian ini menunjukkan bahwa dengan memilih algoritma yang tepat, analisis data EEG dapat dioptimalkan untuk memberikan diagnosis epilepsi yang lebih akurat dan andal. Penggunaan *machine learning* ini membuka peluang besar untuk meningkatkan kemampuan diagnosis dan pemantauan kondisi neurologis melalui analisis data EEG.

Beberapa penelitian diatas menunjukkan betapa pentingnya analisis data EEG dalam berbagai aplikasi medis. Dengan menggunakan metode deep learning, ulasan komprehensif tentang teknik pengolahan sinyal, dan eksplorasi algoritma machine learning, penelitian-penelitian ini menekankan bagaimana data EEG dapat diolah dan dianalisis untuk meningkatkan diagnosis, pemantauan, dan intervensi klinis. Pendekatan berbasis NoSQL untuk penyimpanan dan manajemen data EEG juga didukung oleh fleksibilitas dan kemampuan skalabilitasnya, yang memungkinkan penanganan data EEG yang besar dan kompleks secara efisien.

Berdasarkan latar belakang yang telah diuraikan, penelitian ini bertujuan untuk membandingkan kinerja jenis-jenis basis data NoSQL dalam manajemen data biomedik, dengan fokus pada penggunaan data EEG. Melalui pemahaman yang lebih baik tentang kelebihan dan kekurangan masing-masing platform, diharapkan dapat membuat keputusan yang lebih tepat dalam memilih basis data yang sesuai dengan kebutuhan aplikasi biomedik dan memberikan kontribusi yang berarti dalam pengembangan sistem manajemen data biomedik yang lebih efisien dan efektif.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah pada penelitian ini adalah:

1. Bagaimana membangun dan mengimplementasikan jenis-jenis basis data NoSQL terhadap data biomedik?
2. Bagaimana menguji kinerja *query* jenis-jenis basis data NoSQL terhadap data biomedik?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Untuk membangun dan mengimplementasikan jenis-jenis basis data NoSQL pada sebuah data biomedik.
2. Untuk mengetahui perbandingan kinerja *query* jenis-jenis basis data NoSQL dalam mengelola data biomedik.

1.4 Manfaat Penelitian

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Memberikan wawasan yang lebih mengenai ilmu pengetahuan berupa data hasil kinerja jenis-jenis basis data NoSQL dalam mengelola data biomedik.
2. Hasil perbandingan kinerja dapat dijadikan sebagai referensi dalam proses pemilihan jenis-jenis basis data NoSQL dalam mengelola data biomedik.

1.5 Ruang Lingkup

Ruang lingkup dalam penelitian ini ialah sebagai berikut:

1. Sistem dibangun menggunakan *framework* Laravel
2. Dataset yang digunakan data *Electroencephalography*.
3. MongoDB dan *Elasticsearch* untuk database *no-relational*.

BAB II TINJAUAN PUSTAKA

2.1 Basis Data (*Database*)

Database adalah suatu sistem hardware dan software yang digunakan dalam menyimpan, menghapus, mengolah dan mengalirkan data. Data dapat menjadi sebuah informasi dengan pengelolaan dari suatu program komputer. Pada *database*, data disimpan dengan cara terstruktur. Struktur data memiliki skema khusus dalam menyimpan data. Skema menggambarkan objek yang diwakili suatu *database*, serta hubungan diantara objek tersebut (Heryanto dkk, 2019).

Menurut Winarno dkk (2014), dalam buku Pemrograman Web Berbasis HTML5, PHP, dan JavaScript mengatakan bahwa basis data merupakan sebuah tempat penyimpanan data yang jenisnya beraneka ragam.

Sedangkan menurut Pamungkas (2017), mengemukakan bahwa basis data merupakan suatu kumpulan data yang saling berhubungan disimpan secara bersama-sama pada suatu media, yang mana diorganisasikan berdasarkan sebuah skema atau struktur tertentu, dan menggunakan software untuk melakukan proses manipulasi untuk suatu kegunaan tertentu.

2.2 NoSQL

Teknologi NoSQL pertama kali dikembangkan oleh Carlo Strozzi pada tahun 1998. Dan nama NoSQL mulai terdengar kembali pada tahun 2009 oleh Erick Evans dikarenakan banyak bermunculan situs-situs yang mengalami peningkatan jumlah pengakses data sehingga membebani server basis data. Basis data NoSQL adalah basis data *non-relasional* yang menyediakan mekanisme untuk penyimpanan dan pengambilan data menggunakan model selain hubungan tabular tradisional yang digunakan dalam basis data relasional (Hasibuan dkk, 2023).

Basis data NoSQL tidak menggunakan model data rasional, dapat menyimpan data dalam ukuran besar dan juga memperbolehkan data untuk disimpan didalam record yang tidak mempunyai skema yang sudah tertentu (Suliyanti, 2019).

Menurut (Hasibuan dkk, 2023), Basis data NoSQL telah menjadi populer di berbagai domain, termasuk aplikasi web, e-niaga, media sosial, IoT, dan analitik

real-time. Mereka menawarkan keuntungan dalam skenario di mana model data berkembang, konsumsi data cepat diperlukan, dan skalabilitas horizontal sangat penting. Secara keseluruhan, database NoSQL memberikan pendekatan alternatif untuk penyimpanan dan pengambilan data, menekankan skalabilitas, fleksibilitas, dan kinerja untuk menangani kumpulan data berskala besar dan tidak terstruktur.

a. Model Data

Basis data NoSQL menggunakan berbagai model data, seperti nilai kunci, dokumen, kolom, dan grafik. Model ini umumnya lebih fleksibel dan tanpa skema, memungkinkan data yang dinamis dan tidak terstruktur.

b. Skalabilitas

Basis data NoSQL dirancang untuk skalabilitas horizontal. Mereka dapat mendistribusikan data ke beberapa server dan menangani kumpulan data berskala besar dengan menambahkan perangkat keras komoditas atau menggunakan infrastruktur berbasis *cloud*.

c. Fleksibilitas

Basis data NoSQL lebih cocok untuk menangani data yang tidak terstruktur atau semi-terstruktur. Mereka menawarkan fleksibilitas dalam desain skema dan dapat menangani kebutuhan data yang berkembang. Namun, *database* NoSQL sering mengorbankan beberapa tingkat konsistensi untuk skalabilitas dan kinerja.

d. Membuat Kueri dan Analitik

Basis data NoSQL bervariasi dalam kemampuan permintaannya. Beberapa menyediakan bahasa kueri yang mirip dengan SQL (misalnya, CQL Cassandra), sementara yang lain menawarkan pencarian nilai kunci sederhana atau kueri berbasis dokumen. Basis data NoSQL mungkin tidak memiliki fungsi analitik tingkat lanjut, sehingga pemrosesan data harus dilakukan secara eksternal.

NoSQL dirancang untuk menangani volume besar data tidak terstruktur atau semi-terstruktur dan memberikan skalabilitas dan kinerja tinggi. *Database* NoSQL menawarkan model data yang lebih fleksibel. Fleksibilitas ini memungkinkan database NoSQL beradaptasi dengan perubahan kebutuhan data dan menangani struktur data dinamis yang kompleks.

Ada tiga komponen penting pada aplikasi *database* NoSQL (Sinaga dkk., 2023), yaitu :

- a. *Database*, merupakan wadah yang memiliki struktur penyimpanan yang disebut *collection*.
- b. *Collection*, merupakan kumpulan dokumen-dokumen (*collection* ini setara dengan tabel pada sistem *database* relasional).
- c. *Document*, merupakan unit data satuan terkecil dalam MongoDB yang berisikan baris-baris data yang berupa struktur pasangan *key-value* yang berfungsi saling memasangkan informasi. *Document* dapat dianalogikan seperti *record* pada *database* relasional.

2.2.1 Jenis-jenis *Database* NoSQL

Terdapat 4 jenis *Database* NoSQL yaitu:

1. Basis data *Key-value* adalah kombinasi antara *key* dan *value* yang merupakan basis data inti dari semua basis data NoSQL. *Key-value database* ini lebih sesuai untuk penyimpanan data dalam jumlah besar yang tidak perlu kueri yang rumit untuk mengambilnya. Solusi paling terkenal yang mengadopsi sistem berorientasi nilai kunci adalah: Redis, Memcached, Amazon DynamoDB, Riak, Voldemort Ehcache, Hazelcast (digunakan oleh LinkedIn), OrientDB, Berkeley DB, Oracle NoSQL, dll.
2. Basis data *Document Store* adalah basis data yang menggunakan *record* sebagai dokumen. *Record* ini menyimpan dokumen tidak terstruktur (*unstructured*) atau semi terstruktur (semi *structured documents*). Dalam *document database* menyimpan data dalam dokumen yang mirip dengan objek JSON (JavaScript Object Notation). Jenis *database* ini sangat cocok digunakan untuk *database* yang bertujuan umum. Selain itu, *document database* juga mampu mengakomodasi volume data yang besar. Basis data NoSQL Berorientasi Dokumen utama adalah: MongoDB (10gen), Elasticsearch, CouchDB, CouchBase, Amazon DynamoDB, MarkLogic, RavenDB, Cloudant, OrientDB, GemFire, RethinkDB, Datameer, Microsoft Azure DocumentDB, ArangoDB, PouchDB, dll

3. Basis data berbentuk Kolom adalah basis data yang berorientasi kolom. *Column-based database* memberikan banyak fleksibilitas dari pada *database* yang relasional karena setiap baris tidak diharuskan memiliki kolom yang sama. Setiap kolom dibuat secara terpisah dan nilai dalam *database* kolom tunggal disimpan secara berdekatan. Jenis *database* ini memberikan kinerja tinggi pada *aggregation queries* seperti SUM, Count, AVG, hingga MIN karena datanya sudah tersedia di kolom. *Column-based* ini banyak digunakan untuk mengelola *data warehouse, business intelligence*, hingga CRM. Solusi utama NoSQL Berorientasi Kolom adalah: Cassandra (Apache), HBase (Apache), Bigtable (Google), Accumulo (Apache), Hypertable, dll.
4. Basis data Grafis terdiri dari node, properties (karakteristik) dan edge. *Graph database* menyimpan data dalam node dan edge. Node biasanya menyimpan informasi tentang orang, tempat, dan benda-benda. Sementara itu, edge menyimpan informasi tentang hubungan antar node. Jenis *database* yang satu ini lebih unggul dalam penggunaan di mana pengguna perlu mencari tahu hubungan atau pola. Misalnya untuk *social network*, deteksi penipuan, logistik hingga rekomendasi. Solusi paling terkenal yang telah mengadopsi NoSQL Berorientasi Grafik adalah: Neo4j, Orient DB, Titan, ArangoDB, Giraph, InfiniteGraph, Sqrrl, Sparksee, InfoGrid, HyperGraphDB, FlockDB, VelocityGraph, GlobalsDB, GraphDB, dll.

2.3 MongoDB

MongoDB merupakan salah satu basis data NoSQL. Metode yang digunakan oleh MongoDB adalah *document-store/document-oriented* dimana penyimpanan data dimasukkan kedalam dokumen seperti Javascript Object Notation (JSON) (Akhmad dkk, 2016).

Berbeda dengan *database* SQL yang menyimpan data dengan relasi tabel, MongoDB menyimpan data dalam format BSON (Binary JSON). Ini adalah *database* yang paling umum digunakan dan dirancang untuk ketersediaan dan

skalabilitas tinggi, menyediakan berbagai otomatis dan replikasi bawaan (Frankie dkk, 2023).

MongoDB dapat digunakan untuk membangun aplikasi analitik. MongoDB menyediakan fleksibilitas model data, skalabilitas elastis, bersama dengan kinerja dan ketahanan basis data NoSQL. Akibatnya, pengembang dapat terus meningkatkan aplikasi, dan mengirimkannya pada skala yang hampir tak terbatas dimanapun mereka memilih untuk menjalankannya. MongoDB memberikan fleksibilitas yang baik dalam sistem *database*. Data dengan kompleksitas yang berbeda dapat dimasukkan ke dalam yang dapat digunakan dalam situasi kebutuhan data yang sangat besar (Hamid dkk, 2015).

MongoDB mempunyai fitur-fitur sebagai berikut (Silalahi dkk, 2018):

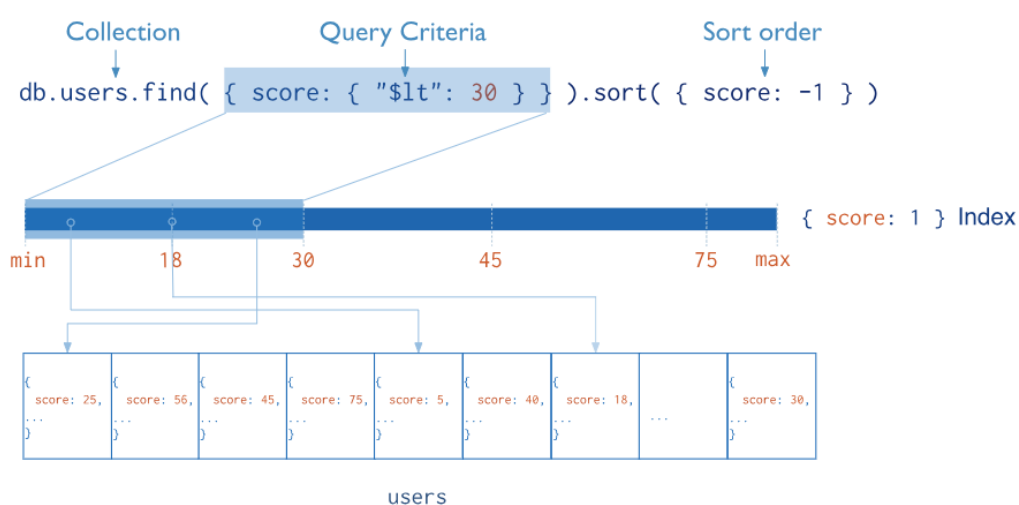
- a. *Document Oriented*, MongoDB tidak mengambil dan memecah entitas menjadi beberapa struktur relasional, tetapi MongoDB menyimpannya dalam jumlah dokumen yang minimal.
- b. *Ad Hoc Queries*, MongoDB mendukung pencarian berdasarkan *field*, *range queries*, dan *regular expression*. Hasil dari *query* dapat berupa *field-field* tertentu dari dokumen, termasuk penggunaan fungsi-fungsi JavaScript.
- c. *Indexing*, Setiap *field* di dokumen MongoDB dapat diberi indeks yang menyediakan efisiensi dalam pencarian data.
- d. *Replication*, MongoDB memberikan ketersediaan yang tinggi untuk kumpulan-kumpulan replika. Sebuah replika berisi dua atau lebih salinan data.
- e. *Load Balancing*, Skalabilitas MongoDB bersifat horizontal menggunakan *sharding*. Pengguna memilih sebuah kunci *shard*, untuk menentukan bagaimana data dalam sebuah koleksi akan didistribusikan.

2.3.1 Proses *Indexing*

Indeks mendukung eksekusi *query* yang efisien di MongoDB. Tanpa indeks, MongoDB harus melakukan pemindaian setiap dokumen dalam *collection*, untuk memilih dokumen yang cocok dengan pernyataan *query*. Jika ada indeks yang sesuai untuk *query*, MongoDB dapat menggunakan indeks untuk membatasi jumlah

dokumen yang harus diperiksa. Indeks adalah struktur data khusus yang menyimpan sebagian kecil dari kumpulan data *collection* dalam bentuk yang sederhana. Indeks menyimpan nilai bidang tertentu atau sekumpulan bidang, diurutkan berdasarkan nilai bidang (MongoDB, 2023).

Pengurutan *entry* indeks mendukung pencocokan kesetaraan yang efisien dan operasi kueri berbasis rentang. Selain itu, MongoDB dapat mengembalikan hasil yang diurutkan dengan menggunakan pengurutan di indeks. Diagram berikut mengilustrasikan *query* yang memilih dan mengurutkan dokumen yang cocok menggunakan indeks:



Gambar 1 Diagram Ilustrasi *Query* MongoDB

(Sumber: MongoDB, 2023)

Pada dasarnya, indeks di MongoDB mirip dengan indeks di sistem *database* lain. MongoDB mendefinisikan indeks di tingkat koleksi dan mendukung indeks di setiap bidang atau sub-bidang dokumen dalam koleksi MongoDB (MongoDB, 2023).

2.4 *Elasticsearch*

Elasticsearch adalah sistem pencarian terbuka, gratis, terdistribusi, dan *analytics engine* untuk seluruh tipe data, yaitu: tekstual, numerikal, geospasial, data terstruktur, dan data tidak terstruktur. *Elasticsearch* dibangun di atas sistem *apache lucene* dan pertama kali dikeluarkan pada tahun 2010. Cara kerja *elasticsearch* yaitu data mentah diindeks ke dalam basis data *elasticsearch* dari sumber sistem

utama. Data yang akan di indeks ke dalam *elasticsearch* sebelumnya penguraian dan normalisasi. Setelah data di indeks, pengguna dapat menjalankan *query* yang kompleks dan dapat melakukan agregasi serta pengelompokan data.

Elasticsearch indeks adalah koleksi dari dokumen-dokumen yang saling terhubung satu sama lain. *Elasticsearch* menyimpan data dalam bentuk dokumen JSON. Tiap dokumen berkorelasi pada sekumpulan *property keys* serta value dari properti tersebut. *Elasticsearch* menggunakan struktur data yang disebut *inverted index*, yaitu struktur data yang dirancang untuk melakukan pencarian dengan cepat. Sebuah *inverted index* terdiri dari sekumpulan kata yang unik yang muncul pada dokumen dan mengidentifikasi seluruh dokumen tempat setiap kata yang muncul. Berdasarkan penelitian yang dilaksanakan oleh Rahmat Firman (2021) berjudul *Perancangan Sistem Pencarian Berbasis Full-Text Search Engine dengan Elasticsearch Studi Kasus: Web Digital Library Fakultas Teknik Universitas Hasanuddin*, terdapat beberapa fitur dalam sistem *elasticsearch*, yaitu:

1. Indeks

Dalam *elasticsearch*, indeks dapat dianggap sebagai basis data. Indeks diatur dengan mengirim *query* ke *elasticsearch* melalui akses API. Pada indeks ini akan diatur jumlah *shard* yang akan digunakan dan jumlah *replica* yang akan dibuat.

2. Mapping

Mapping adalah bagian dimana pengguna mengatur bagaimana data akan disimpan dalam *elasticsearch*. Di bagian inilah kolom-kolom dokumen diatur beserta tipe datanya. Pengguna juga dapat mengatur bagaimana teks-teks yang akan di indeks di tokenisasi, misalnya *tags* HTML akan dibersihkan sebelum diinputkan ke dalam indeks. Dalam satu indeks pengguna dapat menambahkan beberapa *mapping*. *Mapping* dapat dianggap sebagai tabel *elasticsearch*.

3. Dokumen

Dokumen dalam *elasticsearch* merupakan *entity*, terdiri dari beberapa kolom dan setiap kolomnya memiliki penamaan dan dapat menyimpan satu atau lebih nilai dalam bentuk *array*. Dokumen dalam *elasticsearch* akan mempunyai kolom yang berbeda dan dokumen yang

disimpan akan memiliki format JSON. Dokumen dapat dianggap sebagai kolom dalam *elasticsearch*.

4. *Type*

Setiap dokumen yang ada dalam *elasticsearch* memiliki *type*. *Type* dalam dokumen dikelompokkan ke dalam beberapa grup logika untuk mempermudah proses pencarian.

5. *Node*

Node adalah *instance* dari sebuah *elasticsearch*. *Elasticsearch* memerlukan minimal satu *node* untuk dapat bekerja. Jika lebih dari satu *node*, maka penamaan *node* biasanya menggunakan urutan angka. *Node-node* ini akan saling berbagi data dan beban menyelesaikan suatu tugas yang diberikan. Satu *node* akan bertindak sebagai *node master* yang akan mengatur *node* yang lain seperti penambahan atau penghapusan *node*.

6. *Cluster*

Kumpulan dari beberapa *node* dapat disebut *cluster*. Konsep penggunaan *cluster* adalah untuk mengurangi beban dari sistem dengan membagi tugas-tugas menjadi beberapa bagian. Kelebihan dari sistem *cluster* adalah ketika sistem mengalami kegagalan pada *node* tertentu maka sistem akan tetap beroperasi dengan mengalihkan kinerja *node* yang gagal ke *node* yang masih aktif, sementara *node* yang gagal akan melakukan *restart* agar dapat beroperasi lagi.

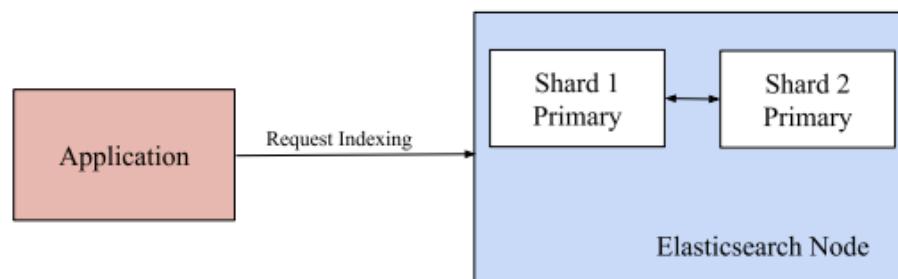
7. *Shard*

Data yang ada dalam *elasticsearch* akan dibagi menjadi beberapa bagian atau *chunks* kemudian disimpan ke dalam *shard*. *Shard* merupakan partisi dari suatu indeks. Ketika indeks pertama kali dibuat, pengguna dapat mengatur jumlah *shard* yang akan digunakan. Ketika pengguna memasukkan data ke dalam *elasticsearch* (*indexing*), maka *elasticsearch* akan secara otomatis melakukan *sharding* ke dalam *shard*. Pengguna tidak dapat mengetahui data yang diambil dari *elasticsearch* berasal dari *shard* mana.

2.4.1 Indexing Elasticsearch

Menurut Rafał Kuć dkk (2013), dalam bukunya yang berjudul *Mastering Elasticsearch*, cara untuk melakukan proses *indexing* yaitu dengan menggunakan index API yang telah disediakan *elasticsearch*. Dimana dokumen yang akan di indeks dikirim menggunakan curl tool, berikut adalah contoh perintah untuk melakukan *indexing*: `curl -XPUT http://localhost:9200/blog/article/1 -d '{"title": "New version of Elasticsearch released!", "content": "...", "tags": ["announce", "elasticsearch", "release"]}'`

Berikut adalah alur proses *indexing* pada sistem pencarian *elasticsearch*:



Gambar 2 Proses *Indexing Single Node*

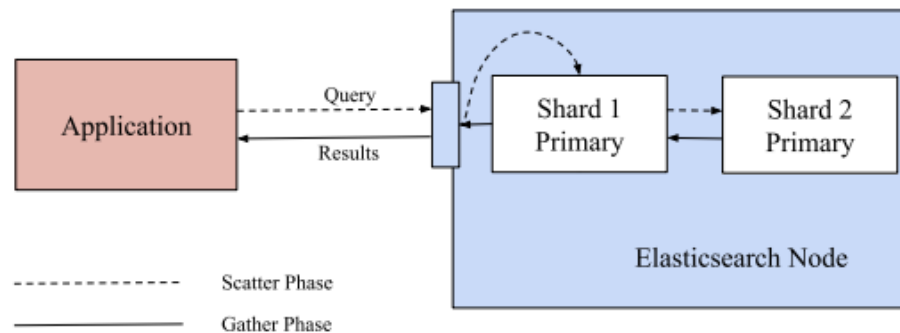
(Sumber: *Mastering Elasticsearch*, 2013)

Proses *indexing* terjadi saat pengguna mengirimkan *request* untuk melakukan pemasukan data ke dalam sistem pencarian *elasticsearch*. Sebelum melakukan *indexing* data, nama *node*, *cluster*, dan *replica* telah dilakukan konfigurasi terlebih dahulu untuk menghindari tumpang tindih saat proses *indexing*. Data yang akan di indeks dilakukan pemisahan agar dapat disimpan ke dalam *shard* yang sesuai secara otomatis. Pada Gambar 2, hanya dilakukan instansiasi 1 *node* saja maka dapat dipastikan *request* akan dikirim ke *node* master. Pada kasus yang khusus, terdapat lebih dari satu instansiasi *node* jika terjadi kesalahan indeks aplikasi mengirimkan *request* ke *node* yang memiliki *shard replica*, maka data yang diindeks akan diteruskan ke *node* yang memiliki *shard primer*.

2.4.2 Querying Data

Menurut Rafał Kuć dkk (2013), dalam bukunya yang berjudul *Mastering Elasticsearch*, *query* API dalam sistem pencarian *elasticsearch* menggunakan *dsl query* (berbasiskan JSON untuk mendefinisikan *query* yang kompleks).

Berikut adalah alur proses yang *query* dalam *elasticsearch*:



Gambar 3 Proses *Query Single Node*

(Sumber: *Mastering Elasticsearch*, 2013)

Fase *scatter* terjadi saat pengguna mengirimkan *request* ke dalam sistem yang akan dikirimkan ke seluruh *shard* dalam *node*. *Shard* tersebut akan mengeluarkan data-data yang relevan berdasarkan *query dsl* yang dikirim melalui *request*. Setelah fase *scatter* selesai, maka dilanjutkan pada fase *gather*. Pada fase ini, terjadi proses penggabungan, penyortiran, dan pengolahan data yang telah dikeluarkan untuk dikirim kembali dalam bentuk *response* pada aplikasi utama.

2.5 Framework Laravel

Menurut Naista (2017), mengemukakan bahwa *framework* adalah suatu struktur konseptual dasar digunakan untuk memecahkan atau menangani suatu masalah yang bersifat kompleks. Singkatnya, *framework* merupakan suatu kerangka kerja dari sebuah *website* yang akan dibangun. Dengan menggunakan kerangka tersebut, waktu yang diperlukan dalam membangun sebuah *website* menjadi lebih singkat dan memudahkan dalam proses perbaikan.

Sedangkan definisi Laravel menurut Naista (2017), mengatakan bahwa Laravel merupakan salah satu *framework* berbasis *PHP* bersifat *open source* (terbuka), dan menggunakan konsep MVC (*model – view – controller*). Laravel berada di bawah lisensi MIT *License* dengan menggunakan Github sebagai tempat berbagi *code* menjalankannya.

Menurut Abdulloh (2017), mengatakan bahwa terdapat beberapa keunggulan yang dimiliki Laravel yaitu sebagai berikut.

1. Laravel memiliki banyak fitur yang tidak dimiliki oleh *framework* lain.
2. Laravel merupakan *framework PHP* yang ekspresif, artinya sintaks pada Laravel menggunakan bahasa yang mudah dimengerti sehingga *programmer* pemula sekalipun akan mudah memahami kegunaan dari suatu sintaks meskipun *programmer* tersebut belum mempelajarinya.
3. Laravel memiliki dokumentasi yang cukup lengkap, bahkan setiap versinya memiliki dokumentasi tersendiri mulai dari cara instalasi hingga penggunaan fitur-fiturnya.
4. Laravel digunakan oleh banyak *programmer* sehingga banyak *library* yang mendukung Laravel yang diciptakan para *programmer* pecinta Laravel.
5. Laravel didukung oleh *Composer* sehingga *library-library* diperoleh dengan mudah dari internet menggunakan *Composer*.
6. Laravel memiliki *template engine* tersendiri yang diberi nama *blade* yang memudahkan dalam menampilkan data pada *template* HTML.

Adapun fitur-fitur yang dimiliki Laravel antara lain sebagai berikut (Aminudin, 2015).

- a) *Bundles* yaitu sebuah fitur dengan sistem pengemasan modular dan berbagai *bundle* telah tersedia untuk digunakan dalam aplikasi.
- b) *Eloquent ORM* merupakan penerapan PHP lanjutan dari pola *active record* menyediakan metode internal untuk mengatasi kendala hubungan antara objek *database*.
- c) *Application Logic* merupakan bagian dari aplikasi yang dikembangkan, baik menggunakan *controllers* maupun sebagai bagian dari deklarasi

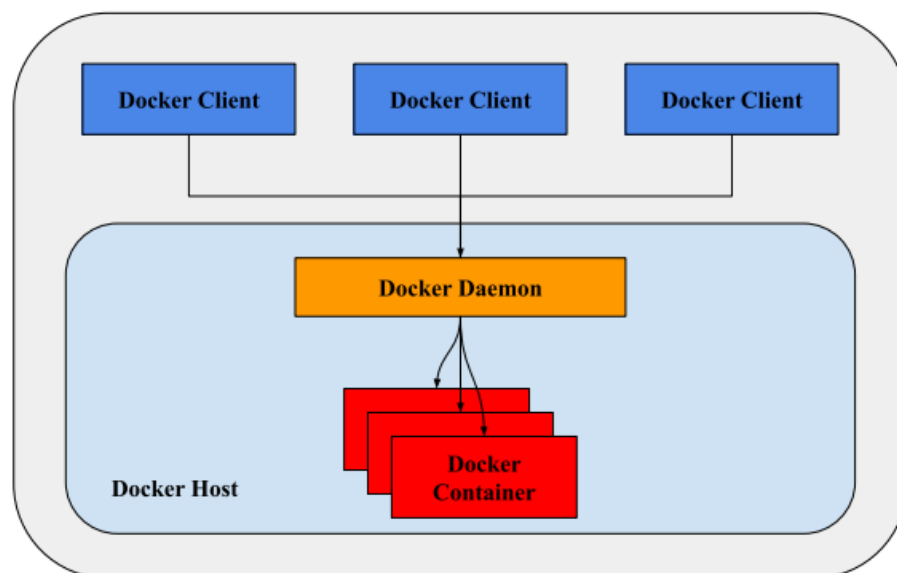
route. Sintaks yang digunakan untuk mendefinisikannya mirip dengan yang digunakan oleh *framework* Sinatra.

- d) *Reverse Routing*, mendefinisikan hubungan antara *link* dan *route*. Sehingga jika suatu saat ada perubahan pada *route* secara otomatis akan tersambung dengan *link* yang relevan. Ketika *link* yang dibuat dengan menggunakan nama-nama dari *route* yang ada, secara otomatis Laravel akan membuat URI yang sesuai.
- e) *Restful Controller*, memberikan sebuah pilihan untuk memisahkan logika dalam melayani HTTP *GET* dan permintaan *POST*.
- f) *Class Auto Loading*, menyediakan otomatis *loading* untuk kelas-kelas PHP, tanpa membutuhkan pemeriksaan manual terhadap jalur masuknya. Fitur ini mencegah *loading* yang tidak perlu.
- g) *View Composers* adalah kode unit *logical* yang dapat dijalankan ketika sebuah *view* di *load*.
- h) *IoC Container* memungkinkan untuk objek baru yang dihasilkan dengan mengikuti prinsip *control* pembalik, dengan pilihan contoh dan referensi dari objek baru sebagai Singletons.
- i) *Migrations* menyediakan versi sistem *control* untuk skema *database*, sehingga memungkinkan untuk menghubungkan perubahan adalah basis kode aplikasi dan keperluan yang dibutuhkan dalam merubah tata letak *database*, mempermudah dalam penempatan dan memperbarui aplikasi.
- j) *Unit Testing* mempunyai peran penting dalam *framework* Laravel, dimana *unit testing* ini mempunyai banyak tes untuk mendeteksi dan mencegah regresi. *Unit testing* dapat dijalankan melalui fitur “*artisan command-line*”.
- k) *Automatic Pagination* menyederhanakan tugas dari penerapan halaman, menggantikan penerapan yang manual dengan metode otomatis yang terintegrasi ke Laravel.

2.6 Docker

Docker adalah *open source platform* yang dapat menjalankan aplikasi dan membuat proses pengembangan dan distribusi menjadi lebih mudah. Aplikasi yang dibangun dalam docker mengemas semua *dependencies* pendukung aplikasi di dalam sebuah *container*. *Container* berjalan dalam sebuah lingkungan yang terisolasi di atas sistem operasi. Terdapat fasilitas yang disediakan oleh docker, yaitu untuk membuat dan melakukan kontrol dengan *container*. Dengan melakukan virtualisasi aplikasi, pengembang dengan mudah menjalankan aplikasi di mana saja tanpa perlu melakukan banyak konfigurasi sistem. Terlebih docker dapat dengan mudah di *deploy* ke lingkungan *cloud computing*. Terdapat 4 komponen internal, yaitu:

1. Docker Client and Server



Gambar 4 Arsitektur Docker

Docker dapat dijelaskan sebagai aplikasi yang berbasis klien dan server seperti pada Gambar 4. Docker server mendapatkan *request* dari docker *client* kemudian memprosesnya. Docker server dan docker klien dapat dijalankan pada server yang sama atau docker klien dapat dihubungkan dengan sebuah *remote* server yang berjalan pada server yang berbeda.

2. Docker Images

Docker images adalah *tools* atau kumpulan *file* yang menunjang sebuah aplikasi untuk membangun sebuah *container*.

3. Docker Registries

Docker images disimpan di dalam *docker registries*. Bertugas menyimpan *source code repository* dimana pengguna dapat melakukan *pull* dan *push image* dari sebuah sumber tunggal. Terdapat dua tipe *registry*, yaitu *public* dan *private*. *Docker hub* disebut sebagai *public registry*, yaitu tempat bagi pengguna untuk melakukan *pull* pada *image* yang tersedia dan melakukan *push* terhadap *image* yang dibuat. *Images* dapat didistribusikan baik secara publik dan privat melalui fitur *docker hub*.

4. Docker Containers

Docker containers adalah wadah untuk mengemas dan menjalankan aplikasi. Wadah ini mencakup kode, *runtime*, *system tools*, dan pengaturan. *Container* hanya bisa mengakses *resource* yang telah ditentukan dalam *Docker image*.

Docker memiliki banyak kelebihan dibandingkan *virtual machine* yaitu:

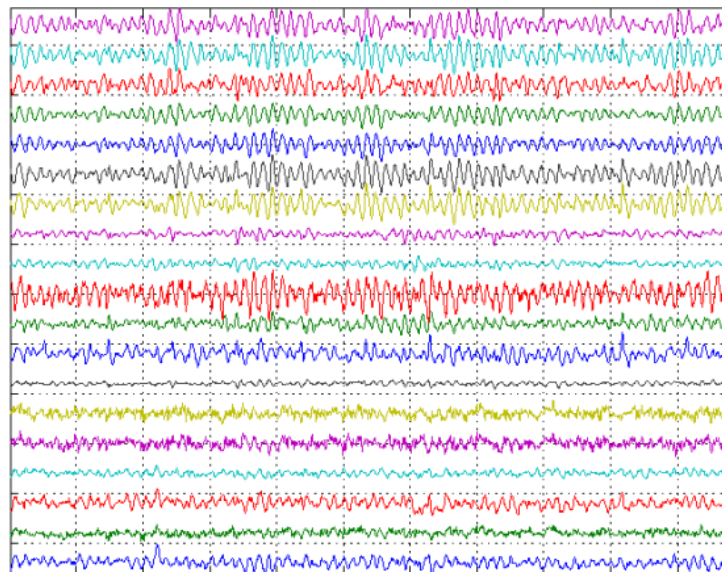
- a) Docker dapat menjalankan sebuah sistem dalam waktu yang sangat cepat dibandingkan *virtual machine*.
- b) *Container* lebih ringan dibandingkan *virtual machine* karena *container* hanya menjalankan library dan aplikasi yang dijalankan saja. Sedangkan *virtual machine* harus menjalankan seluruh komponen seperti perangkat keras, kernel, sistem operasi, dan aplikasi.
- c) Aplikasi yang dijalankan dengan *docker* sangat mudah dipindahkan dari satu server ke server lain karena hanya perlu menjalankan satu *script* konfigurasi saja.
- d) Alokasi *resource hard disk* dan RAM pada *container* dibagi oleh *host server*, *container* akan mengambil alokasi *resource* yang ada di *hardware* sesuai dengan yang *container* butuhkan.

2.7 Electroencephalography (EEG)

Electroencephalography (EEG) merupakan metode untuk merekam aktivitas listrik otak pada permukaan kulit kepala. Penggunaan EEG pada manusia pertama kali dilakukan oleh Hans Berger pada tahun 1924, yang kemudian mulai menuai perhatian pada tahun 1930. EEG menjadi salah satu alat utama dalam mendiagnosis epilepsi, selain itu juga dapat digunakan untuk mendeteksi kelainan yang berkaitan dengan fungsi serebral.

Pengukuran respons otak terhadap stimulus melalui EEG menjadi salah satu metode yang paling umum digunakan dalam bidang neurosains kognitif yang mengasosiasikan antara aktivitas fisiologis dengan pemrosesan informasi, sensori, perseptual, hingga aktivitas kognitif.

Data EEG merupakan salah satu jenis data *time series* multivariat yang dihasilkan dari beberapa saluran. Setiap saluran berhubungan dengan satu rangkaian waktu univariat.



Gambar 5 Menunjukkan Contoh Data EEG

(Sumber: Cao dkk, 2019)

Untuk data EEG, beberapa sinyal dikenali berdasarkan bentuknya, distribusi spasial di kulit kepala, dan sifat simetri. Pola ritme dan periodik dalam data EEG “*ictal-interictal contin-uum*” (IIC) sering dikaitkan dengan kejang (Cao dkk, 2019).

Sel otak pada dasarnya hanya menghasilkan potensial listrik yang sangat rendah, sehingga metode pengukuran dengan EEG hanya dapat merekam aktivitas

elektris yang dihasilkan oleh kumpulan sel yang tersinkronisasi sekaligus, dan hanya terbatas pada area korteks (bagian otak besar). Sebagian besar aktivitas elektrik yang dapat direkam oleh EEG bersumber dari neuron piramidal. Sel-sel tersebut cenderung memiliki orientasi yang sama. Secara teknis EEG tidak merekam potensial aksi antar sel otak, namun agregasi dari potensial listrik yang dihasilkan dari potensial post-sinaptik.

Otak manusia terdiri atas setidaknya 86 milyar sel/neuron. Sel-sel otak tersebut memproses informasi dalam bentuk potensial listrik dan diteruskan ke sel lain melalui sambungan yang disebut sinapsis.

Aktivitas sel-sel otak menghasilkan osilasi, atau yang kerap diasosiasikan dengan gelombang otak, dengan frekuensi (jumlah gelombang per detik), amplitudo (kekuatan gelombang) dan fase yang berbeda-beda. Otak manusia menghasilkan 5 jenis gelombang utama yang diklasifikasikan berdasarkan frekuensinya: gelombang alpha (8–13 Hertz), theta (4–8 Hertz), beta (14–26 Hertz), delta (0.5–4.0 Hertz), gamma (di atas 30 Hertz) dan mu (8–13 Hertz). Frekuensi gelombang alpha dan mu menunjukkan rentang frekuensi yang sama, perbedaan di antara keduanya adalah lokasi dari osilasi tersebut: gelombang mu ditemukan pada korteks motorik, sementara gelombang alpha umumnya ditemukan pada korteks visual (Khakim dkk, 2021).

Gelombang yang terekam oleh EEG merupakan hasil dari percampuran beberapa osilasi yang muncul bersamaan pada frekuensi yang berbeda. Untuk dapat melihat amplitudo dari tiap gelombang (*band power*), dilakukan proses kalkulasi matematis analisis transformasi Fourier, konsep ini akan diperdalam pada bagian ekstraksi fitur (Khakim dkk, 2021).

Dengan menggunakan analisis fourier, sinyal dapat didekomposisi menjadi spektrum frekuensi dalam rentang kontinu. Setelah dihaluskan menjadi menghilangkan kebisingan, representasi domain frekuensi yang dihasilkan dari sinyal disebut spektrumnya. Dengan menggunakan jendela geser dan menghitung serangkaian spektrum dari waktu ke waktu, dan menyusun spektrum ini ke dalam matriks frekuensi waktu, yang disebut spektogram, dimungkinkan untuk memberikan tampilan “terkompresi” atau “diperkecil” dari EEG. Sedangkan dokter biasanya melihat sinyal EEG “mentah” pada skala 10-15 detik / layar, spektogram

biasanya dilihat dalam skala besar 1-4 jam, memberikan konteks temporal. Spektrogram membantu para ahli mengenali pola seperti kejang, dan meninjau EEG yang panjang file secara efisien (Cao dkk, 2019).

2.7.1 Perangkat Keras EEG

Umumnya terdapat 4 proses yang berjalan dalam pengukuran aktivitas otak dengan EEG yaitu: (1) transduksi dari sinyal bioelektris (potensial listrik) menjadi arus listrik oleh elektroda, (2) transmisi sinyal dari sensor elektroda menuju amplifier, (3) amplifikasi sinyal EEG oleh amplifier, (4) konversi dari sinyal analog menjadi digital (Khakim dkk, 2021).

a. Elektroda

Elektroda Proses pertama dalam menangkap sinyal bioelektris tersebut dipengaruhi oleh kualitas dari sensor atau elektroda. Pada umumnya elektroda terbuat dari perak klorida (AgCl), aluminium, perunggu, perak, maupun emas yang ditempelkan pada permukaan kulit kepala dengan tambahan gel elektrolit untuk meningkatkan konduktivitas potensial listrik. Elektroda tersebut merupakan jenis elektroda basah. Elektroda basah merupakan gold standard dalam teknik perekaman data EEG.

Elektroda basah memberikan kualitas data terbaik dibandingkan elektroda jenis lain, namun kelemahannya terletak pada proses pemasangannya yang lebih sulit dan memakan lebih banyak waktu. Berbeda dengan elektroda basah, elektroda kering umumnya tidak menggunakan gel elektrolit dalam aplikasi pada kulit kepala. Kualitas data yang dihasilkan dari elektroda kering umumnya tidak sebaik dengan hasil data dengan elektroda basah, meskipun demikian elektroda kering sudah cukup memadai untuk digunakan dalam konteks penelitian.

Elektroda kering memiliki kelebihan dalam kemudahan pemasangannya. Selain itu, penggunaan elektroda kering juga mengurangi risiko munculnya “salt bridge” yang muncul ketika elektroda basah terlalu berdekatan dan/atau gel elektrolit tersambung

antar elektroda. Salt bridge mengakibatkan dua elektroda secara tidak langsung tersambung sehingga merekam data dengan voltase yang sama dan seolah menjadi satu elektroda. Selain terbagi atas basah dan kering, elektroda juga ada yang bersifat pasif dan aktif. Pada elektroda aktif, amplifikasi sinyal langsung dilakukan pada elektroda itu sendiri, sehingga mengurangi noise/artefak dari frekuensi listrik (artefak akan dibahas lebih lanjut). Elektroda aktif menunjukkan performa lebih baik dalam kondisi impedansi (perihal impedansi akan dibahas lebih lanjut pada bagian artefak) elektroda yang cukup tinggi ($> 2 \text{ k}\Omega$), namun dalam kondisi yang ideal elektroda pasif tetap memproduksi data yang lebih baik.

b. Amplifier

Data yang tertangkap oleh elektroda selanjutnya perlu dilakukan amplifikasi mengingat kecilnya sinyal yang terdeteksi. Proses ini dilakukan dengan perangkat amplifier. Amplifier merupakan perangkat keras terpenting dalam merekam data EEG. Amplifier memiliki sampling rate, atau jumlah data yang terekam dalam satu waktu tertentu. Sampling rate direpresentasikan dalam frekuensi Hertz (Hz), jumlah data yang terekam dalam satu detik. Amplifier dengan sampling rate 250 Hz berarti merekam 1 sampel setiap 4 milidetik, atau 250 sampel setiap detik. Semakin tinggi sampling rate dari amplifier tersebut, semakin akurat representasi data digital EEG dengan data aslinya yang berbentuk analog/ gelombang. Rendahnya sampling rate akan berisiko oleh aliasing error, yaitu eror yang terjadi ketika frekuensi yang terdeteksi berbeda dengan aslinya.

Pemilihan amplifier idealnya mengikuti tujuan penelitian. Semakin tinggi resolusi temporal (sampling rate) amplifier semakin baik, namun hal tersebut juga meningkatkan cost penelitian dari segi rasio antara harga dengan hasil yang didapatkan. Rekomendasi umum dalam memilih amplifier adalah yang memiliki sampling rate 3-4 kali lebih tinggi dari frekuensi tertinggi yang ingin direkam untuk menghindari aliasing error.

2.7.2 Kelebihan dan kekurangan dari EEG

1. Kelebihan

Kelebihan utama dari EEG adalah karena memiliki resolusi temporal yang sangat baik dibandingkan brain imaging lainnya, EEG dapat merekam data dalam rentang hingga milidetik (akan dibahas lebih lanjut di bagian amplifier) memungkinkan untuk meneliti proses simultan yang terjadi segera setelah kemunculan stimulus. Sementara itu resolusi temporal dari pengukuran hemodynamic membutuhkan waktu setidaknya beberapa ratus milidetik mengingat lambatnya respons aliran darah. Pengukuran menggunakan EEG relatif lebih murah dan fleksibel ketimbang metode brain imaging lainnya sehingga memungkinkan aplikasi praktis yang lebih luas dalam konteks braincomputer interface (BCI).

2. Kekurangan

Lemahnya resolusi spasial dari EEG tersebut menjadikan metode pengukuran ini tidak tepat digunakan dalam konteks studi untuk mengasosiasikan area otak spesifik terhadap perilaku. signifikansinya dalam mengevaluasi keadaan koma pada otak dibatasi oleh fakta bahwa hasilnya seringkali tidak ditentukan oleh kondisi otak itu sendiri. Dalam hal ini, penggunaan brain imaging lainnya akan memberikan kesimpulan yang lebih akurat, misalnya pengukuran yang berdasarkan hemodynamic (berdasarkan aliran darah, misalnya fMRI dan PET).

2.8 Penelitian Terkait

Berikut ini merupakan beberapa penelitian terkait dengan penelitian yang dilakukan:

1. Analisis Spektral Daya Dan Koherensi EEG Pada Anak Penderita *Autism Spectrum Disorders* (ASD)

Penelitian yang dilakukan oleh Handayani dkk (2017), menggunakan teknik *Quantitative Electroencephalography* (QEEG) untuk memahami perbedaan karakteristik fungsionalitas otak antara anak autis dan anak sehat. Data EEG direkam menggunakan Emotiv Epoc dengan 14 channel

dan 2 channel referensi pada kondisi rileks dan mata tertutup. Analisis dilakukan dengan memproses data EEG untuk menghilangkan noise dan artefak, menghitung spektral daya menggunakan periodogram Welch, dan menganalisis koherensi *intra-hemisphere* dan *inter-hemisphere*. Hasil penelitian menunjukkan peningkatan spektral daya pada pita *delta* dan penurunan pada pita *alpha* pada anak autisme, serta koherensi yang lebih rendah pada pita *delta* dan *theta*, khususnya pada *area frontal*.

Karakteristik Data:

1. Subjek Penelitian: Dataset terdiri dari 6 anak penderita autisme dan 5 anak sehat sebagai kontrol, dengan rentang usia 10-15 tahun.
2. Kondisi Perekaman: Data EEG direkam selama 15 menit pada kondisi rileks dan mata tertutup menggunakan Emotiv EPOC 14 *channel* dan 2 *channel* referensi.
3. Proses *Pre-Processing*: Data EEG diproses untuk menghilangkan *noise* dan *artefak*, termasuk *centering*, *filtering*, dan *rejecting* data yang dipengaruhi oleh kedipan mata dan gerakan otot.
4. Spektral Daya: Spektral daya dihitung menggunakan periodogram Welch, yang memberikan informasi tentang konten harmonik dari sinyal EEG.
5. Analisis Koherensi: Koherensi *intra-hemisphere* dan *inter-hemisphere* dihitung untuk mengevaluasi konektivitas fungsional otak, dengan fokus pada pita frekuensi *delta* dan *theta*, serta *area frontal*.

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek (*subject_id*), usia (*age*), jenis kelamin (*gender*), diagnosis medis (*diagnosis*), tanggal perekaman (*recording_date*), dan kondisi saat perekaman (*condition*). Data sinyal EEG yang direkam terdiri dari *timestamp*, yang menunjukkan waktu perekaman dalam detik, serta nilai tegangan listrik yang direkam dari 14 saluran EEG utama (AF3, F7, F3, FC5, T7, P7, O1, O2, P8, T8, FC6, F4, F8, AF4) dan 2 saluran referensi (CMS dan DRL). Total, terdapat

23 kolom dalam dataset ini. Jumlah baris data tergantung pada durasi perekaman dan frekuensi *sampling*.

2. Analisa Pengaruh Rangsangan Aromaterapi Lavender dan Kayu Cendana Terhadap Kualitas Tidur Berbasis Gelombang EEG

Penelitian yang dilakukan oleh Fauzan dkk (2019), membahas pengaruh aromaterapi terhadap kualitas tidur. Penelitian ini dilakukan dengan memanfaatkan sinyal EEG untuk mengukur dan menganalisis kualitas tidur subjek. Perekaman EEG dilakukan menggunakan Mitsar-EEG-202 dan perangkat lunak WinEEG. Subjek penelitian terdiri dari 9 laki-laki berusia 20-23 tahun. Eksperimen dilakukan dalam tiga kondisi: tanpa stimulus, dengan stimulus aromaterapi lavender, dan dengan stimulus kayu cendana. Hasil analisis menunjukkan bahwa secara statistik, tidak terdapat pengaruh signifikan dari penggunaan aromaterapi terhadap kualitas tidur, meskipun beberapa subjek menunjukkan peningkatan efisiensi tidur secara individual.

Karakteristik Data:

1. Subjek Penelitian: Dataset terdiri dari 9 laki-laki sehat berusia 20-23 tahun.
2. Perekaman EEG: Data EEG direkam menggunakan perangkat Mitsar-EEG-202 dengan 19 *channel* dan 2 *channel* referensi.
3. Kondisi Perekaman: Perekaman dilakukan dalam tiga kondisi: tanpa stimulus, dengan stimulus aromaterapi lavender, dan dengan stimulus kayu cendana. Setiap sesi berlangsung selama 54-94 menit.
4. Analisis Data: Analisis meliputi perhitungan latensi tidur, durasi tidur, efisiensi tidur, dan *Wake After Sleep Onset* (WASO), serta analisis spektral daya untuk fase tidur NREM dan REM menggunakan perangkat lunak WinEEG dan MATLAB.

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek (*subject_id*), usia

(*age*), jenis kelamin (*gender*), kondisi saat perekaman (*condition*), dan tanggal perekaman (*recording_date*). Data sinyal EEG yang direkam terdiri dari *timestamp*, yang menunjukkan waktu perekaman dalam detik, serta nilai tegangan listrik yang direkam dari 19 saluran EEG utama (Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2) dan 2 saluran referensi (A1, A2). Total, terdapat 25 kolom dalam dataset ini. Jumlah baris data tergantung pada durasi perekaman dan frekuensi *sampling*.

3. Objective Stress Measurement: Studi Korelasi Parameter Saliva Amylase Dan Aktivitas Gelombang Otak Menggunakan Electroencephalograph (EEG)

Penelitian yang dilakukan oleh Sahroni dkk (2020), membahas tentang pengukuran tingkat stres menggunakan kombinasi parameter saliva amylase dan aktivitas gelombang otak yang direkam dengan EEG. Penelitian ini melibatkan 10 subjek yang diberikan stimulasi stres melalui permainan komputer untuk meningkatkan tingkat stres mereka. Hasil penelitian menunjukkan korelasi kuat antara aktivitas gelombang beta di area frontal otak dan tingkat saliva amylase, terutama pada elektroda F8. Hal ini menunjukkan bahwa aktivitas gelombang otak di area frontal merupakan indikator penting untuk mengukur tingkat stres secara objektif.

Karakteristik Data:

1. Subjek Penelitian: Dataset terdiri dari 10 subjek mahasiswa aktif Universitas Islam Indonesia (UII) dengan rentang umur 19-22 tahun.
2. Perekaman EEG: Data EEG direkam menggunakan alat MITSAR-EEG dengan 21 kanal elektrode. Frekuensi *sampling* yang digunakan adalah 250 Hz dengan jenis elektroda basah yang memerlukan gel khusus untuk konduktivitas.
3. Kondisi Perekaman: Perekaman dilakukan selama 40 menit yang dibagi menjadi beberapa segmen: istirahat awal, tiga segmen

permainan dengan jeda untuk pengambilan saliva, dan istirahat akhir.

4. Analisis Data: Analisis korelasi Pearson digunakan untuk mengevaluasi hubungan antara parameter EEG (*delta, theta, alpha, beta, dan gamma*) dengan skor saliva amylase, yang menunjukkan korelasi kuat terutama pada gelombang beta di area frontal (F8).

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek (*subject_id*), usia (*age*), jenis kelamin (*gender*), kondisi saat perekaman (*condition*), dan tanggal perekaman (*recording_date*). Data sinyal EEG yang direkam terdiri dari *timestamp*, yang menunjukkan waktu perekaman dalam detik, serta nilai tegangan listrik yang direkam dari 19 saluran EEG utama (Fp1, Fp2, F7, F3, Fz, F4, F8, T3, C3, Cz, C4, T4, T5, P3, Pz, P4, T6, O1, O2). Total, terdapat 24 kolom dalam dataset ini. Jumlah baris data tergantung pada durasi perekaman dan frekuensi *sampling*.

4. Pengambilan Data Pasien Stroke Menggunakan Sinyal EEG Sebagai Informasi Pengambilan Keputusan Melakukan Tindak Lanjut Rehabilitasi

Penelitian yang dilakukan oleh Sulistyono (2021), membahas tentang penggunaan sinyal EEG untuk membantu pengambilan keputusan dalam rehabilitasi pasien stroke. Penelitian ini bertujuan untuk mengembangkan metode yang akurat untuk mengukur perkembangan rehabilitasi pasien stroke melalui analisis sinyal EEG. Metode yang digunakan meliputi persiapan, studi literatur, pemilihan partisipan, eksplorasi alat, pemilihan gerakan, dan pengambilan data. Alat yang digunakan adalah *Headset UltraCortex "Mark IV"* dengan *OpenBCI Cyton Board*. Hasil penelitian menunjukkan bahwa data yang diambil dari sinyal EEG dapat digunakan sebagai dasar untuk pengambilan keputusan dalam rehabilitasi medis pasien stroke.

Karakteristik Data:

1. Subjek Penelitian: Dataset terdiri dari pasien stroke yang sedang menjalani rehabilitasi di Rumah Sakit Umum Universitas Airlangga (RSUA).
2. Perekaman EEG: Data EEG direkam menggunakan *Headset UltraCortex "Mark IV"* dan *OpenBCI Cyton Board* dengan 6 *channel* yang terletak pada frontal, *cortex*, dan *occipital* (F3, F4, C3, C4, O1, O2) serta 2 *channel* referensi (A1, A2).
3. Kondisi Perekaman: Perekaman dilakukan selama sesi rehabilitasi dengan partisipan diminta untuk melakukan gerakan berdasarkan *Manual Muscle Testing* (MMT), termasuk flexi-ekstensi siku, flexi-ekstensi bahu, dan flexi-ekstensi genggam.
4. Analisis Data: Data dianalisis menggunakan metode *Analysis of Variance* (ANOVA) untuk mengevaluasi perbedaan rerata antar grup. Hasil menunjukkan bahwa data EEG dapat digunakan untuk memantau perkembangan rehabilitasi pasien stroke.

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek (*subject_id*), usia (*age*), jenis kelamin (*gender*), kondisi saat perekaman (*condition*), dan tanggal perekaman (*recording_date*). Data sinyal EEG yang direkam terdiri dari *timestamp*, yang menunjukkan waktu perekaman dalam detik, serta nilai tegangan listrik yang direkam dari 6 saluran EEG utama (F3, F4, C3, C4, O1, O2) dan 2 saluran referensi (A1, A2). Total, terdapat 11 kolom dalam dataset ini. Jumlah baris data tergantung pada durasi perekaman dan frekuensi *sampling*.

5. Quantitative Electroencephalography: Cortical Responses under Different Postural Conditions.

Penelitian yang dilakukan oleh Ivaldi dkk (2023), membahas penelitian tentang analisis kuantitatif EEG yang direkam dari 10 sukarelawan sehat. Penelitian ini bertujuan untuk memvalidasi alat yang mampu mengurangi artefak gerakan selama analisis EEG dan menggunakan metode ini untuk menganalisis sinyal EEG yang direkam selama aktivitas motorik tertentu, seperti kontrol postur bi- dan monopodalik. Perekaman EEG dilakukan menggunakan topi eksperimen dengan elektroda berlapis neoprene, yang dirancang untuk meminimalkan artefak gerakan. Data EEG dianalisis menggunakan perangkat lunak MATLAB dan Eeglab. Hasil penelitian menunjukkan bahwa prosedur yang digunakan sangat andal, memungkinkan penurunan signifikan dalam artefak gerakan bahkan selama tugas motorik dengan mata terbuka dan tertutup. Penelitian ini membuktikan bahwa EEG dapat menjadi alat non-invasif yang sangat baik untuk menilai keterlibatan kortikal dalam kontrol postural.

Karakteristik Data:

1. Subjek Penelitian: Dataset melibatkan 10 sukarelawan sehat dengan rata-rata usia 29 tahun.
2. Kondisi Perekaman: Data direkam pada 16 saluran menggunakan topi EEG eksperimental pada frekuensi sampling 512 Hz. Sukarelawan menjalani tugas postural dalam kondisi mata terbuka dan tertutup, termasuk posisi duduk, berdiri bipodalik, dan berdiri monopodalik.
3. Proses *Pre-Processing*: Data EEG diproses untuk menghilangkan *noise* dan *artefak*, termasuk *centering*, *filtering*, dan *rejecting* data yang terpengaruh oleh gerakan tubuh.
4. Analisis Data: Perhitungan kepadatan spektral daya (PSD) untuk pita frekuensi alpha dan beta.

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek (*subject_id*), usia (*age*), jenis kelamin (*gender*), kondisi saat perekaman (*condition*). Data sinyal EEG yang direkam terdiri dari *timestamp*, yang menunjukkan waktu perekaman dalam detik, serta nilai tegangan listrik yang direkam dari 16 saluran EEG utama (Fp1, Fp2, F3, F4, C3, C4, P3, P4, O1, O2, F7, F8, T3, T4, T5, T6). Total, terdapat 21 kolom dalam dataset ini. Jumlah baris data tergantung pada durasi perekaman dan frekuensi *sampling*.

6. Auditory evoked potential EEG-Biometric dataset.

Penelitian yang dilakukan oleh Abo Alzahab dkk (2021) membahas tentang studi pengembangan sistem biometrik berdasarkan rekaman EEG terhadap 20 relawan. Tujuan dari penelitian ini adalah untuk mengetahui penggunaan sinyal EEG sebagai alat otentikasi pengguna, dengan menggunakan perbedaan *respons* otak saat istirahat dan saat menerima rangsangan pendengaran. Sinyal EEG direkam selama percobaan menggunakan metode rangsangan auditori yang berbeda, yaitu menggunakan *headphone in-ear* dan *bone-conducting*. Data EEG dianalisis untuk membandingkan respons otak dalam kondisi mata terbuka dan tertutup, serta pengaruh rangsangan auditori yang berbeda terhadap sinyal EEG. Perekaman EEG dilakukan menggunakan papan OpenBCI Ganglion dengan empat saluran utama, yaitu T7, F8, Cz, dan P4, yang dipilih sesuai dengan sistem penempatan elektroda internasional 10/10. Data EEG direkam pada frekuensi *sampling* 200 Hz dan diolah menggunakan perangkat lunak OpenBCI GUI untuk meminimalkan noise dan artefak selama perekaman. Hasilnya menunjukkan bahwa sinyal EEG dapat digunakan sebagai biometrik yang efektif untuk autentikasi pengguna, dengan hasil yang konsisten dalam kondisi mata terbuka dan tertutup, serta *respons* yang berbeda terhadap rangsangan auditori.

Karakteristik Data:

1. **Subjek Penelitian:** Dataset ini melibatkan 20 sukarelawan dengan usia yang tidak ditentukan, di mana kondisi perekaman mencakup mata terbuka dan mata tertutup, serta percobaan dengan rangsangan auditori menggunakan *in-ear* dan *bone-conducting headphones*.
2. **Kondisi Perekaman:** Data direkam pada 4 saluran EEG utama (T7, F8, Cz, dan P4) menggunakan papan OpenBCI Ganglion dengan frekuensi sampling 200 Hz. Sukarelawan menjalani sesi rekaman dalam kondisi istirahat dengan mata terbuka dan tertutup, serta saat menerima rangsangan auditori berupa lagu dalam bahasa asli dan non-asli, serta musik netral.
3. **Proses *Pre-Processing*:** Data EEG diproses untuk menghilangkan noise dan artefak, termasuk segmentasi manual dari rekaman asli dan penerapan *filter Butterworth* orde pertama dengan rentang frekuensi 1-40 Hz serta *filter notch* 50 Hz.
4. **Analisis Data:** Analisis data melibatkan pemisahan sinyal EEG berdasarkan kondisi rekaman dan jenis rangsangan auditori yang diberikan, serta perbandingan *respons* otak dalam kondisi mata terbuka dan tertutup.

Struktur data EEG dalam penelitian ini mencakup beberapa elemen penting. Data EEG terdiri dari metadata dan data sinyal EEG yang direkam. Metadata mencakup kolom seperti ID subjek, nomor eksperimen, dan nomor sesi rekaman. Data sinyal EEG yang direkam terdiri dari indeks sampel, yang menunjukkan waktu perekaman dalam sampel, serta nilai tegangan listrik yang direkam dari 4 saluran EEG utama (T7, F8, Cz, dan P4). Total, terdapat 5 kolom dalam dataset ini, dengan jumlah baris terhadap 1 sukarelawan total 240.010 baris data yang dihasilkan dari berbagai sesi dan kondisi.