

Artikel

Pemeringkatan Framework Backend Terbaik untuk Aplikasi Web Menggunakan Metode TOPSIS

Rivalri Kristianto Hondro^{1,*}, Alwin Fau², Mawati Zalukhu³

¹Informatika, Universitas Audi Indonesia; Medan; Sumatera Utara; Indonesia; rivalryhondro@gmail.com

²Informatika, Universitas Satya Terra Bhinneka; Medan; Sumatera Utara; Indonesia; Indonesia; alwinfau@gmail.com

³Independent Researcher; Medan; Sumatera Utara; Indonesia

Abstrak: Pemilihan *framework backend* dalam pengembangan aplikasi web menjadi aspek penting karena berpengaruh terhadap performa, efisiensi penggunaan sumber daya, kemudahan pengembangan, dan keamanan sistem. Banyaknya alternatif *framework* menyebabkan proses pemilihan memerlukan pendekatan yang objektif agar keputusan tidak hanya didasarkan pada popularitas atau preferensi pengembang. Penelitian ini bertujuan untuk melakukan pemeringkatan *framework backend* terbaik untuk aplikasi web dengan menggunakan metode *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*. Metode TOPSIS diterapkan dengan lima alternatif, yaitu Laravel, Django, Express.js, Gin, dan Spring Boot. Kriteria yang digunakan meliputi performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan. Proses perhitungan dilakukan melalui penyusunan matriks keputusan, normalisasi matriks, pembentukan matriks ternormalisasi terbobot, penentuan solusi ideal positif dan negatif, perhitungan jarak solusi ideal, serta penentuan nilai preferensi setiap alternatif. Hasil penelitian menunjukkan bahwa Gin memperoleh nilai preferensi tertinggi sebesar 0,8150, diikuti oleh Spring Boot sebesar 0,5700, Express.js sebesar 0,3203, Laravel sebesar 0,2571, dan Django sebesar 0,2402. Hasil tersebut menunjukkan bahwa Gin memiliki posisi paling dekat dengan solusi ideal positif dan paling jauh dari solusi ideal negatif dibandingkan alternatif lainnya. Berdasarkan hasil pemeringkatan, Gin direkomendasikan sebagai *framework backend* terbaik untuk pengembangan aplikasi web berdasarkan kriteria yang digunakan dalam penelitian ini. Penerapan metode TOPSIS pada penelitian ini menunjukkan bahwa pemilihan *framework backend* dapat dilakukan secara lebih sistematis, terukur, dan proporsional dengan mempertimbangkan kriteria benefit dan cost.

Kata Kunci: TOPSIS; *Framework Backend*; Sistem Pendukung Keputusan; Aplikasi Web; Pemeringkatan.

Abstract: The selection of a backend framework in web application development is an important aspect because it affects system performance, resource efficiency, development ease, and application security. The large number of available framework alternatives requires an objective approach so that the selection process is not based solely on popularity or developer preference. This study aims to rank the best backend frameworks for web applications using the *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)* method. The TOPSIS method was applied to five alternatives, namely Laravel, Django, Express.js, Gin, and Spring Boot. The criteria used in this study include performance, memory consumption, library ecosystem, ease of learning, and built-in security. The calculation process consists of constructing the decision matrix, normalizing the matrix, forming the weighted normalized matrix, determining the positive and negative ideal solutions, calculating the distance from the ideal solutions, and determining the preference value of each alternative. The results show that Gin achieved the highest preference value of 0.8150, followed by Spring Boot with 0.5700, Express.js with 0.3203, Laravel with 0.2571, and Django with 0.2402. These results indicate that Gin has the closest position to the positive ideal solution and the farthest distance from the negative ideal solution compared with the other alternatives. Based on the ranking results, Gin is recommended as the best backend framework for web application development according to the criteria used in this study. The application of the TOPSIS method in this study shows that backend framework selection can be conducted in a more systematic, measurable, and proportional manner by considering both benefit and cost criteria.

Kata Kunci: TOPSIS, Backend Framework, Decision Support System, Web Application, Ranking.

Riwayat Artikel:

Diajukan : 01-05-2026

Direvisi : 01-06-2026

Diterima : 16-06-2026

Diterbitkan : 30-06-2026

Hak Cipta: © 2026 oleh penulis.

Artikel ini adalah artikel akses terbuka yang didistribusikan di bawah syarat dan ketentuan Creative Commons Attribution-ShareAlike 4.0
creativecommons.org/licenses/by-sa/4.0/

1. Pendahuluan

Perkembangan aplikasi web modern menuntut backend yang tidak hanya mampu memproses permintaan pengguna, tetapi juga mendukung keamanan, skalabilitas, efisiensi sumber daya, serta kemudahan integrasi dengan layanan lain. *Backend framework* menjadi komponen strategis karena berperan dalam pengelolaan logika bisnis, komunikasi API, autentikasi, manajemen basis data, serta pengaturan alur pertukaran data antara client dan server. Dalam konteks tersebut, pemilihan *framework backend* tidak dapat lagi hanya didasarkan pada popularitas teknologi atau preferensi subjektif pengembang, karena setiap framework memiliki karakteristik teknis yang berbeda dari sisi performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan [1], [2], [3].

Urgensi pemeringkatan *framework backend* muncul karena keputusan pemilihan teknologi dapat memengaruhi kualitas aplikasi dalam jangka panjang. *Framework* dengan performa tinggi dapat meningkatkan kecepatan pemrosesan request, sedangkan *framework* dengan konsumsi memori yang efisien dapat mengurangi beban infrastruktur. Di sisi lain, ekosistem library yang matang mempercepat proses pengembangan, kemudahan pembelajaran mendukung produktivitas tim, dan fitur keamanan bawaan dapat mengurangi risiko kerentanan aplikasi. Oleh karena itu, pemilihan *framework backend* idealnya dilakukan melalui pendekatan sistematis yang mampu mengevaluasi banyak kriteria secara bersamaan [1], [4], [5].

Penelitian terdahulu menunjukkan bahwa *framework backend* telah banyak dikaji dari berbagai sudut pandang. Azzahidi, Wijayanto, dan Darmawan melakukan evaluasi performa beberapa backend framework untuk REST API, termasuk Spring Boot, Express.js, Laravel, dan Gin, dengan mempertimbangkan parameter response time, throughput, dan penggunaan sumber daya [1]. Penelitian tersebut menegaskan bahwa perbedaan framework dapat menghasilkan karakteristik performa yang berbeda, sehingga proses pemilihan framework perlu disesuaikan dengan kebutuhan sistem. Selain itu, Kadim, Hadjaratie, dan Muthia menunjukkan bahwa Laravel dapat digunakan untuk membangun sistem berbasis web secara terstruktur melalui dukungan fitur framework yang membantu proses implementasi aplikasi [2]. Sementara itu, Yanuardi, Aminudin, dan Faiqurahman menekankan bahwa efektivitas layanan web juga dipengaruhi oleh arsitektur komunikasi API yang digunakan, sehingga aspek backend perlu dipandang sebagai bagian penting dalam kualitas layanan sistem [3].

Kajian lain juga memperlihatkan bahwa framework tertentu memiliki keunggulan sesuai konteks penggunaannya. Prasha, Sambada, Athillah, dan Salamah menggunakan Django dalam pengembangan sistem lelang online berbasis *Model View Controller* dan menunjukkan bahwa Django dapat mendukung percepatan pengembangan serta peningkatan keamanan sistem [4]. Pada sisi pengambilan keputusan, Hts dkk. menerapkan TOPSIS dalam sistem pendukung keputusan seleksi guru baru, sedangkan Simanjuntak, Siagian, dan Anggriani menerapkan TOPSIS untuk pemilihan smartphone Android [5], [6]. Rizki, Juledi, dan Irmayani juga menggunakan TOPSIS dalam pemilihan produktivitas tanaman terbaik, yang menunjukkan bahwa TOPSIS relevan digunakan pada kasus pemeringkatan alternatif berdasarkan banyak kriteria [7].

Metode TOPSIS dipilih dalam penelitian ini karena memiliki prinsip pengambilan keputusan yang kuat, yaitu menentukan alternatif terbaik berdasarkan kedekatannya terhadap solusi ideal positif dan jaraknya dari solusi ideal negatif [8]. Berbeda dari pendekatan penjumlahan terbobot sederhana, TOPSIS mempertimbangkan posisi relatif setiap alternatif terhadap kondisi terbaik dan terburuk pada seluruh kriteria. Karakteristik tersebut sesuai untuk pemeringkatan *framework backend* karena alternatif yang dinilai memiliki keunggulan dan kelemahan yang tidak selalu seimbang pada setiap kriteria. Dengan demikian, TOPSIS dapat menghasilkan urutan prioritas yang lebih proporsional, terutama ketika data penilaian terdiri dari kriteria benefit dan *cost* [7], [8], [9].

Penelitian ini menggunakan lima alternatif framework backend, yaitu Laravel, Django, Express.js, Gin, dan Spring Boot. Laravel dipilih karena banyak digunakan dalam pengembangan aplikasi web berbasis PHP serta memiliki dukungan fitur seperti *routing*, ORM, autentikasi, dan struktur MVC yang memudahkan pengembangan aplikasi [2], [10]. Django dipilih karena menawarkan pendekatan pengembangan cepat, fitur bawaan yang lengkap, serta dukungan keamanan yang kuat pada aplikasi web berbasis Python [4], [11]. Express.js dipilih karena merupakan *framework* Node.js yang minimalis, fleksibel, dan banyak digunakan dalam pengembangan REST API maupun aplikasi berbasis JavaScript [1], [12]. Gin dipilih karena dikenal sebagai framework Go yang ringan, cepat, dan sesuai untuk pengembangan API dengan kebutuhan performa tinggi [1], [13]. Spring Boot

dipilih karena banyak digunakan pada pengembangan aplikasi backend berbasis Java, terutama untuk aplikasi berskala enterprise, microservices, dan sistem yang memerlukan konfigurasi produksi yang matang [1], [14].

Berdasarkan uraian di atas maka penelitian ini berfokus pada pemeringkatan *framework backend* terbaik untuk aplikasi web menggunakan metode TOPSIS. Penelitian ini tidak hanya membandingkan *framework* berdasarkan satu aspek teknis, tetapi mengevaluasi beberapa kriteria secara terpadu agar hasil pemeringkatan lebih objektif, terukur, dan dapat dipertanggungjawabkan. Kontribusi penelitian ini terletak pada penyediaan model pendukung keputusan yang dapat membantu pengembang, mahasiswa, maupun organisasi dalam memilih *framework backend* yang paling sesuai dengan kebutuhan pengembangan aplikasi web.

2. Metodologi

Bagian metodologi menjelaskan tahapan penelitian dalam pemeringkatan *framework backend* menggunakan metode TOPSIS. Penelitian ini mencakup penentuan alternatif, penetapan kriteria, penyusunan matriks keputusan, normalisasi, pembobotan, dan perhitungan nilai preferensi. Alternatif yang digunakan adalah Laravel, Django, Express.js, Gin, dan Spring Boot, dengan kriteria performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan. Hasil akhir metodologi ini berupa peringkat *framework backend* terbaik berdasarkan kedekatannya dengan solusi ideal positif dan jaraknya dari solusi ideal negatif.

2.1 Tahapan Penelitian

Tahapan penelitian dapat digambarkan dan dijelaskan sebagai berikut:



Gambar 1. Tahapan Penelitian

Tahapan penelitian dilakukan sebagai berikut:

1. Studi Literatur

Mengkaji teori mengenai *framework backend*, sistem pendukung keputusan, dan metode TOPSIS sebagai dasar pemeringkatan alternatif.

2. Penentuan Alternatif

Menentukan lima *framework backend* yang akan dibandingkan, yaitu Laravel, Django, Express.js, Gin, dan Spring Boot.

3. Penentuan Kriteria

Menetapkan kriteria penilaian yang terdiri dari performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan.

4. Penyusunan Matriks Keputusan

Menyusun nilai setiap alternatif berdasarkan masing-masing kriteria ke dalam bentuk matriks keputusan.

5. Perhitungan Metode TOPSIS

Melakukan normalisasi matriks, pembobotan kriteria, penentuan solusi ideal positif dan negatif, serta perhitungan jarak setiap alternatif terhadap solusi ideal.

6. Pemeringkatan Alternatif

Menentukan nilai preferensi setiap *framework* dan mengurutkannya dari nilai tertinggi hingga terendah sebagai hasil akhir pemeringkatan.

2.2 Metode TOPSIS

Metode TOPSIS digunakan untuk menentukan alternatif terbaik berdasarkan kedekatannya terhadap solusi ideal positif dan jaraknya dari solusi ideal negatif. Pada penelitian ini, TOPSIS digunakan untuk melakukan pemeringkatan *framework backend* berdasarkan beberapa kriteria, yaitu performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan. Tahapan metode TOPSIS adalah sebagai berikut:

1. Menyusun matriks keputusan

Matriks keputusan berisi nilai setiap alternatif terhadap masing-masing kriteria.

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix} \dots\dots\dots (1)$$

Keterangan:

x_{ij} = nilai alternatif ke-i pada kriteria ke-j,

m = jumlah alternatif,

n = jumlah kriteria.

2. Melakukan normalisasi matriks keputusan

Normalisasi dilakukan agar seluruh nilai kriteria berada pada skala yang sebanding.

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \dots\dots\dots (2)$$

Keterangan:

r_{ij} = nilai matriks ternormalisasi,

x_{ij} = nilai awal alternatif ke-i pada kriteria ke-j.

3. Menghitung matriks ternormalisasi terbobot

Nilai hasil normalisasi dikalikan dengan bobot masing-masing kriteria

$$y_{ij} = w_j \times r_{ij} \dots\dots\dots (3)$$

Keterangan:

y_{ij} = nilai ternormalisasi terbobot,

w_j = bobot kriteria ke-j,

r_{ij} = nilai ternormalisasi.

4. Menentukan solusi ideal positif dan solusi ideal negatif

Solusi ideal positif merupakan nilai terbaik dari setiap kriteria, sedangkan solusi ideal negatif merupakan nilai terburuk dari setiap kriteria.

$$\begin{aligned} A^+ &= \{y_1^+, y_2^+, \dots, y_n^+\} \\ A^- &= \{y_1^-, y_2^-, \dots, y_n^-\} \end{aligned} \dots\dots\dots (4)$$

Untuk kriteria benefit, nilai ideal positif adalah nilai maksimum dan nilai ideal negatif adalah nilai minimum. Untuk kriteria cost, nilai ideal positif adalah nilai minimum dan nilai ideal negatif adalah nilai maksimum.

5. Menghitung jarak setiap alternatif terhadap solusi ideal

Jarak terhadap solusi ideal positif dihitung dengan rumus:

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_{ij} - y_j^+)^2} \dots\dots\dots (5)$$

Jarak terhadap solusi ideal negatif dihitung dengan rumus:

$$D_i^- = \sqrt{\sum_{j=1}^n (y_{ij} - y_j^-)^2} \dots\dots\dots (6)$$

Keterangan:

D_i^+ = jarak alternatif ke-i terhadap solusi ideal positif,

D_i^- = jarak alternatif ke-i terhadap solusi ideal negatif.

6. Menghitung nilai preferensi setiap alternatif

Nilai preferensi menunjukkan tingkat kedekatan alternatif terhadap solusi ideal positif.

$$C_i = \frac{D_i^-}{D_i^+ + D_i^-} \dots\dots\dots (7)$$

Keterangan:

C_i = nilai preferensi alternatif ke-i.

Semakin besar nilai C_i , maka semakin baik posisi alternatif tersebut dalam pemeringkatan.

7. Menentukan peringkat akhir

Alternatif diurutkan berdasarkan nilai preferensi tertinggi hingga terendah. Framework dengan nilai preferensi tertinggi ditetapkan sebagai framework backend terbaik dalam penelitian ini.

2.2 Penentuan Alternatif Framework Backend

Pemilihan alternatif didasarkan pada relevansi teknologi dalam pengembangan aplikasi web modern, keterwakilan ekosistem bahasa pemrograman, dukungan dokumentasi resmi, serta ketersediaan data evaluasi yang bersumber dari hasil benchmark, penelitian terdahulu, dokumentasi resmi, dan penilaian berbasis karakteristik teknis *framework*. Beberapa alternatif *framework backend* yang digunakan sebagai berikut:

1. Laravel dipilih karena merepresentasikan *framework backend* berbasis PHP yang banyak digunakan dalam pengembangan sistem informasi berbasis web [10]. Penelitian Kadim dkk. menunjukkan Laravel mampu mendukung pembangunan sistem web secara terstruktur, sedangkan dokumentasi Laravel menegaskan ketersediaan fitur seperti routing, ORM, autentikasi, validasi, dan manajemen basis data [2].
2. Django dipilih karena mewakili *framework backend* berbasis Python yang menekankan pengembangan cepat, keamanan, dan kelengkapan fitur bawaan. Penelitian Prasha dkk. menunjukkan bahwa penggunaan Django dalam sistem lelang online dapat mempercepat pengembangan dan meningkatkan keamanan sistem [1]. Hal ini sejalan dengan dokumentasi Django yang menyediakan fitur autentikasi, session, caching, pagination, dan mekanisme keamanan aplikasi web [15].
3. Express.js dipilih karena merepresentasikan *framework backend* berbasis JavaScript/Node.js yang ringan dan fleksibel. Express.js relevan untuk pengembangan REST API karena menyediakan mekanisme routing dan middleware yang sederhana, serta banyak digunakan pada aplikasi web dan mobile berbasis API [1], [12].
4. Gin dipilih karena merepresentasikan *framework backend* berbasis Go yang menonjol pada performa dan efisiensi. Dokumentasi Gin menjelaskan bahwa *framework* ini menggunakan routing berbasis radix tree, footprint memori kecil, serta dukungan middleware, sehingga sesuai untuk aplikasi API yang membutuhkan respons cepat dan penggunaan sumber daya yang efisien [1], [13].
5. Spring Boot dipilih karena merepresentasikan *framework backend* berbasis Java yang banyak digunakan pada aplikasi enterprise dan microservices. Spring Boot menyediakan fitur auto-configuration, embedded server, starter dependencies, metrics, health checks, dan konfigurasi produksi yang matang. Dalam penelitian Azzahidi dkk., Spring Boot juga termasuk *framework* yang dievaluasi dalam konteks performa REST API [1], [14].

2.4 Penentuan Kriteria dan Bobot

Kriteria ditentukan berdasarkan aspek teknis dan nonteknis yang relevan dalam pemilihan *framework backend* untuk pengembangan aplikasi web. Penentuan kriteria diperlukan agar proses pemeringkatan tidak hanya didasarkan pada popularitas *framework*, tetapi juga mempertimbangkan karakteristik yang berpengaruh terhadap performa, efisiensi, kemudahan pengembangan, dan keamanan sistem. Kriteria yang digunakan dalam penelitian ini terdiri dari lima aspek, yaitu performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan.

1. Performa (C1)

Performa digunakan untuk menilai kemampuan framework dalam memproses permintaan atau request dari pengguna. Semakin tinggi kemampuan framework dalam menangani request, maka semakin baik nilai yang diberikan. Kriteria ini termasuk kategori benefit karena nilai yang lebih besar menunjukkan kualitas yang lebih baik.

2. Konsumsi Memori (C2)

Konsumsi memori digunakan untuk mengukur jumlah memori yang dibutuhkan framework ketika menjalankan proses backend. Framework dengan konsumsi memori yang lebih rendah dinilai lebih efisien karena dapat mengurangi penggunaan sumber daya server. Kriteria ini termasuk kategori cost karena nilai yang lebih kecil menunjukkan hasil yang lebih baik.

3. Ekosistem Library (C3)

Ekosistem library digunakan untuk menilai ketersediaan pustaka, modul, atau paket pendukung yang dapat membantu proses pengembangan aplikasi. Semakin luas ekosistem library suatu framework, maka semakin besar pula dukungan yang tersedia bagi pengembang. Kriteria ini termasuk kategori benefit.

4. Kemudahan Pembelajaran (C4)

Kemudahan pembelajaran digunakan untuk menilai tingkat kemudahan framework dipahami dan digunakan oleh pengembang. Aspek ini mencakup kemudahan sintaks, kelengkapan dokumentasi, ketersediaan materi pembelajaran, serta tingkat kompleksitas arsitektur framework. Kriteria ini termasuk kategori benefit karena framework yang lebih mudah dipelajari dapat meningkatkan produktivitas pengembangan.

5. Keamanan Bawaan (C5)

Keamanan bawaan digunakan untuk menilai kelengkapan fitur keamanan yang telah tersedia secara default pada framework. Fitur tersebut dapat mencakup autentikasi, proteksi CSRF, validasi input, pencegahan SQL injection, enkripsi, dan mekanisme keamanan lainnya. Kriteria ini termasuk kategori benefit karena semakin lengkap fitur keamanan bawaan, maka semakin baik framework tersebut dalam mendukung pengembangan aplikasi yang aman.

Bobot kriteria ditentukan berdasarkan tingkat kepentingannya dalam pemilihan framework backend. Performa diberikan bobot tertinggi karena berkaitan langsung dengan responsivitas aplikasi, sedangkan konsumsi memori dan ekosistem library juga menjadi aspek penting dalam efisiensi dan keberlanjutan pengembangan. Kemudahan pembelajaran dan keamanan bawaan tetap diperhitungkan karena berpengaruh terhadap produktivitas pengembang serta perlindungan sistem dari potensi kerentanan.

Tabel 1. Penentuan Kriteria dan Nilai Bobot Kriteria

Kode	Kriteria	Bobot	Jenis
C1	Performa	0,3	Benefit
C2	Konsumsi Memori	0,2	Cost
C3	Ekosistem Library	0,2	Benefit
C4	Kemudahan Pembelajaran	0,15	Benefit
C5	Keamanan Bawaan	0,15	Benefit

Total bobot seluruh kriteria berdasarkan data pada tabel 1 adalah 1,00. Komposisi bobot tersebut menunjukkan bahwa performa menjadi kriteria paling dominan dalam pemeringkatan *framework backend*, sedangkan konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan berperan sebagai faktor pendukung dalam menghasilkan keputusan yang lebih objektif dan proporsional melalui metode TOPSIS.

3. Hasil dan Pembahasan

Hasil pengolahan data dalam pemeringkatan *framework backend* terbaik untuk pengembangan aplikasi web menggunakan metode TOPSIS. Pembahasan dilakukan melalui beberapa tahap, yaitu pengumpulan data, penyusunan matriks keputusan, normalisasi matriks, pembentukan matriks

ternormalisasi terbobot, penentuan solusi ideal positif dan negatif, perhitungan jarak solusi ideal, serta penentuan nilai preferensi. Hasil akhir dari proses ini digunakan untuk menentukan urutan peringkat framework backend berdasarkan nilai preferensi tertinggi.

3.1 Pengumpulan Data

Pengumpulan data dilakukan dengan mengidentifikasi nilai setiap alternatif terhadap kriteria yang telah ditentukan. Alternatif yang digunakan dalam penelitian ini terdiri dari lima framework backend, yaitu Laravel, Django, Express.js, Gin, dan Spring Boot. Kriteria penilaian meliputi performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan.

Penilaian dilakukan dengan pendekatan kuantitatif dan kualitatif. Kriteria performa dan konsumsi memori menggunakan nilai numerik, sedangkan kriteria ekosistem library, kemudahan pembelajaran, dan keamanan bawaan dinilai menggunakan skala 1 sampai 5. Nilai yang lebih tinggi menunjukkan penilaian yang lebih baik pada kriteria benefit, sedangkan pada kriteria cost nilai yang lebih rendah menunjukkan kondisi yang lebih baik.

Tabel 2. Skala Nilai Kriteria Kuantitatif

Nilai	Keterangan
1	Sangat Rendah
2	Rendah
3	Sedang
4	Tinggi
5	Sangat Tinggi

Tabel 3. Skala Nilai Kriteria Kualitatif

Skor	C3 - Ekosistem Library	C4 - Kemudahan Pembelajaran	C5 - Keamanan Bawaan
5	Ekosistem library sangat luas, komunitas besar, dan dukungan package sangat lengkap	Sangat mudah dipelajari, dokumentasi sangat lengkap, dan banyak sumber pembelajaran	Fitur keamanan bawaan sangat lengkap, seperti autentikasi, CSRF protection, validasi input, dan proteksi SQL injection
4	Ekosistem library luas dan komunitas cukup matang	Mudah dipelajari, dokumentasi lengkap, dan materi pembelajaran cukup banyak	Fitur keamanan bawaan lengkap, tetapi masih membutuhkan beberapa konfigurasi tambahan
3	Ekosistem library cukup berkembang, namun belum seluas framework populer lainnya	Cukup mudah dipelajari, tetapi membutuhkan pemahaman konsep framework atau bahasa tertentu	Fitur keamanan dasar tersedia, tetapi sebagian besar masih memerlukan library atau konfigurasi tambahan
2	Ekosistem library terbatas dan dukungan komunitas relatif kecil	Kurang mudah dipelajari karena struktur atau konsep framework cukup kompleks	Fitur keamanan bawaan masih minim dan banyak bergantung pada middleware atau package eksternal
1	Ekosistem library sangat terbatas dan dukungan komunitas rendah	Sulit dipelajari karena dokumentasi terbatas dan kurva belajar tinggi	Hampir tidak memiliki fitur keamanan bawaan yang memadai

3.2 Penyusunan Matriks Keputusan

Penyusunan tabel matriks keputusan dengan ketentuan Nilai C1 dan C2 menggunakan skala nilai kriteria kuantitatif, sedangkan C3, C4, dan C5 menggunakan skala kualitatif 1–5. Pada artikel referensi, kriteria yang digunakan mencakup performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan; C2 bersifat *cost*, sedangkan kriteria lainnya bersifat *benefit*.

Tabel 4. Matriks Keputusan

Kode	Alternatif	C1 Performa (req/detik)	C2 Konsumsi Memori (MB)	C3 Ekosistem Library	C4 Kemudahan Pembelajaran	C5 Keamanan Bawaan
A1	Laravel	12.935	84,2	4	3	5
A2	Django	68.850	130,1	4	4	5
A3	Express.js	92.604	82,5	5	4	2
A4	Gin	406.943	16,7	3	3	3
A5	Spring Boot	310.250	156,8	4	3	5

3.3 Normalisasi Matriks

Normalisasi matriks dilakukan dengan menggunakan rumus normalisasi matriks TOPSIS, meskipun C2 adalah kriteria cost, pada tahap normalisasi TOPSIS tetap menggunakan rumus normalisasi vektor yang sama. Perbedaan jenis benefit dan cost digunakan pada tahap penentuan solusi ideal positif dan negatif.

Berikut hasil normalisasi matriks keputusan setiap kriteria:

Nilai pembagi Normalisasi C1

$$\sqrt{12.935^2 + 68.850^2 + 92.604^2 + 406.943^2 + 310.250^2} = 524.729,2686$$

Tabel 5. Perhitungan Normalisasi C1

Kode	Alternatif	Nilai C1	Rumus	Hasil Normalisasi
A1	Laravel	12.935	$12.935 / 524.729,2686$	0,0247
A2	Django	68.850	$68.850 / 524.729,2686$	0,1312
A3	Express.js	92.604	$92.604 / 524.729,2686$	0,1765
A4	Gin	406.943	$406.943 / 524.729,2686$	0,7755
A5	Spring Boot	310.250	$310.250 / 524.729,2686$	0,5913

Nilai pembagi Normalisasi C2

$$\sqrt{84,2^2 + 130,1^2 + 82,5^2 + 16,7^2 + 156,8^2} = 235,9810$$

Tabel 6. Perhitungan Normalisasi C2

Kode	Alternatif	Nilai C2	Rumus	Hasil Normalisasi
A1	Laravel	84,2	$84,2 / 235,9810$	0,3568
A2	Django	130,1	$130,1 / 235,9810$	0,5513
A3	Express.js	82,5	$82,5 / 235,9810$	0,3496
A4	Gin	16,7	$16,7 / 235,9810$	0,0708
A5	Spring Boot	156,8	$156,8 / 235,9810$	0,6645

Tabel 7. Perhitungan Normalisasi C3

Kode	Alternatif	Nilai C3	Rumus	Hasil Normalisasi
A1	Laravel	4	$4 / 9,0554$	0,4417
A2	Django	4	$4 / 9,0554$	0,4417
A3	Express.js	5	$5 / 9,0554$	0,5522

A4	Gin	3	3 / 9,0554	0,3313
A5	Spring Boot	4	4 / 9,0554	0,4417

Tabel 8. Perhitungan Normalisasi C4

Kode	Alternatif	Nilai C4	Rumus	Hasil Normalisasi
A1	Laravel	3	3 / 7,6811	0,3906
A2	Django	4	4 / 7,6811	0,5208
A3	Express.js	4	4 / 7,6811	0,5208
A4	Gin	3	3 / 7,6811	0,3906
A5	Spring Boot	3	3 / 7,6811	0,3906

Tabel 9. Perhitungan Normalisasi C5

Kode	Alternatif	Nilai C5	Rumus	Hasil Normalisasi
A1	Laravel	5	5 / 9,3808	0,533
A2	Django	5	5 / 9,3808	0,533
A3	Express.js	2	2 / 9,3808	0,2132
A4	Gin	3	3 / 9,3808	0,3198
A5	Spring Boot	5	5 / 9,3808	0,533

Setelah proses perhitungan normalisasi matriks dilakukan disetiap kriteria, berikut hasil akhir matriks keputusan ternormalisasi.

Tabel 10. Normalisasi Matriks Keputusan

Kode	Alternatif	C1	C2	C3	C4	C5
A1	Laravel	0,0247	0,3568	0,4417	0,3906	0,533
A2	Django	0,1312	0,5513	0,4417	0,5208	0,533
A3	Express.js	0,1765	0,3496	0,5522	0,5208	0,2132
A4	Gin	0,7755	0,0708	0,3313	0,3906	0,3198
A5	Spring Boot	0,5913	0,6645	0,4417	0,3906	0,533

3.4 Pembentukan Matriks Ternormalisasi Terbobot

Matriks ternormalisasi terbobot diperoleh dengan mengalikan nilai normalisasi setiap alternatif dengan bobot masing-masing kriteria. Bobot yang digunakan adalah C1 = 0,30; C2 = 0,20; C3 = 0,20; C4 = 0,15; dan C5 = 0,15 sesuai dengan kriteria pada tabel 1.

Tabel 11. Perhitungan Matriks Ternormalisasi Terbobot

Alternatif	C1	C2	C3	C4	C5
A1	$0,30 \times 0,0247 =$ 0,0074	$0,20 \times 0,3568 =$ 0,0714	$0,20 \times 0,4417 =$ 0,0883	$0,15 \times 0,3906 =$ 0,0586	$0,15 \times 0,5330 =$ 0,0800
A2	$0,30 \times 0,1312 =$ 0,0394	$0,20 \times 0,5513 =$ 0,1103	$0,20 \times 0,4417 =$ 0,0883	$0,15 \times 0,5208 =$ 0,0781	$0,15 \times 0,5330 =$ 0,0800
A3	$0,30 \times 0,1765 =$ 0,0530	$0,20 \times 0,3496 =$ 0,0699	$0,20 \times 0,5522 =$ 0,1104	$0,15 \times 0,5208 =$ 0,0781	$0,15 \times 0,2132 =$ 0,0320

A4	$0,30 \times 0,7755 =$ 0,2327	$0,20 \times 0,0708 =$ 0,0142	$0,20 \times 0,3313 =$ 0,0663	$0,15 \times 0,3906 =$ 0,0586	$0,15 \times 0,3198 =$ 0,0480
A5	$0,30 \times 0,5913 =$ 0,1774	$0,20 \times 0,6645 =$ 0,1329	$0,20 \times 0,4417 =$ 0,0883	$0,15 \times 0,3906 =$ 0,0586	$0,15 \times 0,5330 =$ 0,0800

Tabel 12. Hasil Matriks Ternormalisasi Terbobot

Alternatif	C1	C2	C3	C4	C5
A1	0,0074	0,0714	0,0883	0,0586	0,08
A2	0,0394	0,1103	0,0883	0,0781	0,08
A3	0,053	0,0699	0,1104	0,0781	0,032
A4	0,2327	0,0142	0,0663	0,0586	0,048
A5	0,1774	0,1329	0,0883	0,0586	0,08

3.5 Penentuan Solusi Ideal Positif dan Negatif

Tahap penentuan solusi ideal positif dan negatif dilakukan berdasarkan nilai pada matriks ternormalisasi terbobot. Solusi ideal positif (A^+) menunjukkan nilai terbaik dari setiap kriteria, sedangkan solusi ideal negatif (A^-) menunjukkan nilai terburuk dari setiap kriteria. Pada kriteria benefit, nilai terbaik diperoleh dari nilai maksimum dan nilai terburuk diperoleh dari nilai minimum. Sebaliknya, pada kriteria cost, nilai terbaik diperoleh dari nilai minimum karena semakin kecil nilai cost maka semakin baik, sedangkan nilai terburuk diperoleh dari nilai maksimum.

Tabel 12. Perhitungan Solusi Ideal Positif dan Negatif

Kriteria	Jenis	Solusi Ideal Positif (A^+)	Solusi Ideal Negatif (A^-)
C1	Benefit	$\max(0,0074; 0,0394; 0,0530; 0,2327; 0,1774) =$ 0,2327	$\min(0,0074; 0,0394; 0,0530; 0,2327; 0,1774) =$ 0,0074
C2	Cost	$\min(0,0714; 0,1103; 0,0699; 0,0142; 0,1329) =$ 0,0142	$\max(0,0714; 0,1103; 0,0699; 0,0142; 0,1329) =$ 0,1329
C3	Benefit	$\max(0,0883; 0,0883; 0,1104; 0,0663; 0,0883) =$ 0,1104	$\min(0,0883; 0,0883; 0,1104; 0,0663; 0,0883) =$ 0,0663
C4	Benefit	$\max(0,0586; 0,0781; 0,0781; 0,0586; 0,0586) =$ 0,0781	$\min(0,0586; 0,0781; 0,0781; 0,0586; 0,0586) =$ 0,0586
C5	Benefit	$\max(0,0800; 0,0800; 0,0320; 0,0480; 0,0800) =$ 0,0800	$\min(0,0800; 0,0800; 0,0320; 0,0480; 0,0800) =$ 0,0320

Berdasarkan hasil perhitungan tabel 12, diperoleh nilai solusi ideal positif dan solusi ideal negatif sebagai berikut.

Tabel 13. Hasil Solusi Ideal Positif dan Negatif

Solusi Ideal	C1	C2	C3	C4	C5
A^+	0,2327	0,0142	0,1104	0,0781	0,08
A^-	0,0074	0,1329	0,0663	0,0586	0,032

3.6 Perhitungan Jarak Solusi Ideal

Tahap perhitungan jarak solusi ideal dilakukan untuk mengetahui kedekatan setiap alternatif terhadap solusi ideal positif A^+ dan solusi ideal negatif A^- . Jarak terhadap solusi ideal positif menunjukkan seberapa dekat suatu alternatif dengan kondisi terbaik, sedangkan jarak terhadap solusi ideal negatif menunjukkan seberapa jauh alternatif tersebut dari kondisi terburuk. Perhitungan jarak

dilakukan menggunakan nilai pada matriks ternormalisasi terbobot yang telah dibandingkan dengan nilai solusi ideal positif dan negatif.

Tabel 14. Perhitungan Jarak terhadap Solusi Ideal Positif (D_i^+)

Alternatif	Perhitungan	Jumlah	D_i^+
A1	$((0,0074-0,2327)^2 + (0,0714-0,0142)^2 + (0,0883-0,1104)^2 + (0,0586-0,0781)^2 + (0,0800-0,0800)^2)$	0,0549	0,2343
A2	$((0,0394-0,2327)^2 + (0,1103-0,0142)^2 + (0,0883-0,1104)^2 + (0,0781-0,0781)^2 + (0,0800-0,0800)^2)$	0,0471	0,217
A3	$((0,0530-0,2327)^2 + (0,0699-0,0142)^2 + (0,1104-0,1104)^2 + (0,0781-0,0781)^2 + (0,0320-0,0800)^2)$	0,0377	0,1942
A4	$((0,2327-0,2327)^2 + (0,0142-0,0142)^2 + (0,0663-0,1104)^2 + (0,0586-0,0781)^2 + (0,0480-0,0800)^2)$	0,0033	0,0579
A5	$((0,1774-0,2327)^2 + (0,1329-0,0142)^2 + (0,0883-0,1104)^2 + (0,0586-0,0781)^2 + (0,0800-0,0800)^2)$	0,018	0,1342

Tabel 15. Perhitungan Jarak terhadap Solusi Ideal Negatif (D_i^-)

Alternatif	Perhitungan	Jumlah	(D_i^-)
A1	$((0,0074-0,0074)^2 + (0,0714-0,1329)^2 + (0,0883-0,0663)^2 + (0,0586-0,0586)^2 + (0,0800-0,0320)^2)$	0,0066	0,0811
A2	$((0,0394-0,0074)^2 + (0,1103-0,1329)^2 + (0,0883-0,0663)^2 + (0,0781-0,0586)^2 + (0,0800-0,0320)^2)$	0,0047	0,0686
A3	$((0,0530-0,0074)^2 + (0,0699-0,1329)^2 + (0,1104-0,0663)^2 + (0,0781-0,0586)^2 + (0,0320-0,0320)^2)$	0,0084	0,0915
A4	$((0,2327-0,0074)^2 + (0,0142-0,1329)^2 + (0,0663-0,0663)^2 + (0,0586-0,0586)^2 + (0,0480-0,0320)^2)$	0,0651	0,2552
A5	$((0,1774-0,0074)^2 + (0,1329-0,1329)^2 + (0,0883-0,0663)^2 + (0,0586-0,0586)^2 + (0,0800-0,0320)^2)$	0,0317	0,178

Tabel 16. Hasil Akhir Jarak Solusi Ideal

Alternatif	D_i^+	D_i^-
A1	0,2343	0,0811
A2	0,217	0,0686
A3	0,1942	0,0915
A4	0,0579	0,2552
A5	0,1342	0,178

Berdasarkan hasil perhitungan pada tabel 16, A4 memiliki jarak paling kecil terhadap solusi ideal positif dan jarak paling besar terhadap solusi ideal negatif. Kondisi ini menunjukkan bahwa A4 memiliki posisi paling mendekati kondisi terbaik dibandingkan alternatif lainnya. Nilai jarak solusi ideal ini selanjutnya digunakan untuk menghitung nilai preferensi setiap alternatif.

3.7 Penentuan Nilai Preferensi

Tahap terakhir menghitung nilai preferensi setiap alternatif. Nilai preferensi digunakan untuk menentukan tingkat kedekatan suatu alternatif terhadap solusi ideal positif dan solusi ideal negatif. Semakin besar nilai preferensi, maka semakin baik posisi alternatif tersebut dalam hasil pemeringkatan.

Tabel 17. Perhitungan Nilai Preferensi

Alternatif	Perhitungan	Nilai Preferensi
A1	$0,0811 / (0,2343 + 0,0811)$	0,2571
A2	$0,0686 / (0,2170 + 0,0686)$	0,2402
A3	$0,0915 / (0,1942 + 0,0915)$	0,3203
A4	$0,2551 / (0,0579 + 0,2551)$	0,815
A5	$0,1780 / (0,1343 + 0,1780)$	0,57

Tabel 18. Hasil Akhir Nilai Preferensi

Alternatif	D_i^+	D_i^-	Nilai Preferensi	Peringkat
A4	0,0579	0,2551	0,815	1
A5	0,1343	0,178	0,57	2
A3	0,1942	0,0915	0,3203	3
A1	0,2343	0,0811	0,2571	4
A2	0,217	0,0686	0,2402	5

Hasil perhitungan nilai preferensi menunjukkan A4 memperoleh nilai tertinggi sebesar 0,8150 sehingga menempati peringkat pertama sebagai alternatif terbaik. A5 berada pada peringkat kedua dengan nilai preferensi sebesar 0,5700, diikuti oleh A3 sebesar 0,3203, A1 sebesar 0,2571, dan A2 sebesar 0,2402. Dengan demikian, hasil pemeringkatan menunjukkan bahwa A4 merupakan framework backend terbaik berdasarkan perhitungan metode TOPSIS.

4. Kesimpulan

Kesimpulan dari hasil pemeringkatan *framework backend* menggunakan metode TOPSIS, diperoleh bahwa Gin menempati peringkat pertama dengan nilai preferensi sebesar 0,8150, diikuti oleh Spring Boot sebesar 0,5700, Express.js sebesar 0,3203, Laravel sebesar 0,2571, dan Django sebesar 0,2402. Gin menjadi alternatif terbaik karena memiliki kedekatan paling tinggi terhadap solusi ideal positif dan jarak paling jauh dari solusi ideal negatif, terutama didukung oleh keunggulan pada aspek performa dan efisiensi konsumsi memori. Hasil penelitian ini menunjukkan bahwa metode TOPSIS mampu memberikan proses pemeringkatan yang objektif dan terukur karena mempertimbangkan kriteria benefit dan cost secara proporsional. Dengan demikian, Gin dapat direkomendasikan sebagai framework backend terbaik untuk pengembangan aplikasi web berdasarkan kriteria performa, konsumsi memori, ekosistem library, kemudahan pembelajaran, dan keamanan bawaan.

Daftar Pustaka

- [1] A. S. Azzahidi, B. Wijayanto, and A. Darmawan, "Performance Evaluation of Backend Frameworks for REST API: A Comparative Study of Spring Boot, Flask, Express.js, Laravel FrankenPHP, and Gin," *J. Tek. Inform. JUTIF*, vol. 6, no. 4, pp. 2405–2419, Aug. 2025.
- [2] A. A. Kadim, L. Hadjaratie, and M. Muthia, "Implementasi Framework Laravel Dalam Pembuatan Sistem Pencatatan Notula Berbasis Website," *J. Sist. Inf. Bisnis*, vol. 13, no. 1, pp. 45–51, Jul. 2023.
- [3] M. I. Yanuardi, A. Aminudin, and M. Faiqurahman, "Analisis Perbandingan Efektivitas Arsitektur RESTful dan Arsitektur gRPC pada Implementasi Web Service," *J. Edukasi Dan Penelit. Inform. JEPIN*, vol. 10, no. 2, pp. 333–341, Aug. 2024.
- [4] A. A. Prasha, A. Sambada, N. Athillah, and U. Salamah, "Pengembangan Sistem Lelang Online Menggunakan Metode Algoritma Greedy Berbasis Model-View-Controller (MVC)," *JSAI J. Sci. Appl. Inform.*, vol. 7, no. 3, 2024.
- [5] D. I. G. Hts, E. Desi, S. Aliyah, F. P. Nasution, U. Indriani, and F. Edi, "Decision Support System Using the TOPSIS Method in New Teacher Selection," *Sink. J. Dan Penelit. Tek. Inform.*, vol. 8, no. 3, pp. 1706–1714, 2024.
- [6] R. L. Simanjuntak, T. R. Siagian, and V. Anggriani, "Sistem Pendukung Keputusan Menggunakan Metode TOPSIS Dalam Pemilihan Smartphone Android," *J. Ilm. Komputasi*, vol. 23, no. 3, pp. 405–412, Sep. 2024.

- [7] N. S. Rizki, A. P. Juledi, and D. Irmayani, "Sistem Pendukung Keputusan Pemilihan Produktivitas Tanaman Terbaik dengan Menggunakan Metode TOPSIS," *J. Teknikom Tek. Inf. Dan Komput.*, vol. 6, no. 2, pp. 835–844, 2023.
- [8] C. L. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications*. Berlin, Germany: Springer, 1981.
- [9] D. Tinambunan, M. Masrizal, and I. R. Munthe, "Sistem Pendukung Keputusan Penentuan Pemberian Penghargaan Bagi Pelanggan Terbaik Menggunakan Metode TOPSIS," *J. Teknikom Tek. Inf. Dan Komput.*, vol. 6, no. 2, pp. 276–281, 2023.
- [10] Laravel, "Laravel Documentation: Authentication." 2026.
- [11] M. Meidilianto, A. Nurahman, M. S. Ramadhan, and Z. Niqotaini, "Penerapan Metode SAW Dalam Menentukan Teknologi Framework Backend Terbaik Untuk Pengembangan Aplikasi Berbasis Web," *JIKTEKS J. Ilmu Komput. Dan Teknol. Inf.*, vol. 4, no. 1, 2025.
- [12] Express.js, "Application Framework Express Node.js," expressjs. [Online]. Available: <https://expressjs.com/en/blog/>
- [13] Gin Web Framework, "Gin Web Framework Documentation." 2026.
- [14] Spring, "Spring Boot," Spring Framework. [Online]. Available: <https://spring.io/projects/spring-framework>
- [15] F. Django, "Django Documentation," Django Project. [Online]. Available: <https://www.djangoproject.com/start/overview/>