

KLASIFIKASI TINGKAT PRODUKTIVITAS MAHASISWA TEKNIK INFORMATIKA PADA PENGEMBANGAN WEB MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM) (STUDI KASUS MAHASISWA UNISBANK PADA PHP NATIVE DAN LARAVEL)

Ilham Faishal Seto Aji, Rr. Dewi Handayani Untari Ningsih

Teknik Informatika, Universitas Stikubank Semarang

Jl. Kendeng V No. 49, Bendan, Semarang., Indonesia

ilhamfaishalsetoaji18@gmail.com

ABSTRAK

Pengembangan aplikasi web menuntut efisiensi tinggi, memicu perbandingan antara penggunaan PHP Native yang fleksibel dan Framework Laravel yang terstruktur. Permasalahan utama saat ini adalah belum adanya model klasifikasi komputasional untuk memetakan dampak kedua metode tersebut terhadap produktivitas mahasiswa secara akurat. Penelitian ini bertujuan mengklasifikasi tingkat produktivitas mahasiswa Teknik Informatika UNISBANK dan membandingkan efektivitas kedua pendekatan menggunakan algoritma *Support Vector Machine* (SVM). Metode penelitian menggunakan pendekatan kuantitatif melalui kuesioner terhadap 43 responden. Data diproses melalui tahapan *preprocessing*, normalisasi, dan klasifikasi menggunakan SVM kernel *Radial Basis Function* (RBF), kemudian dibandingkan dengan algoritma *Decision Tree* dan *Logistic Regression*. Hasil penelitian menunjukkan persepsi produktivitas pada Laravel ($\mu = 3.53$) lebih tinggi dibandingkan PHP Native ($\mu = 3.27$). Model SVM menghasilkan akurasi sebesar 66.67%, sedangkan *Logistic Regression* memberikan performa terbaik dengan akurasi 77.78%. Analisis *feature importance* mengungkapkan bahwa manajemen struktur file dan modul pada Laravel menjadi indikator paling berpengaruh dalam meningkatkan produktivitas mahasiswa

Kata kunci : *Support Vector Machine, Produktivitas, PHP Native, Laravel, Klasifikasi, Machine Learning*

1. PENDAHULUAN

Efisiensi dan produktivitas mahasiswa dalam pengembangan aplikasi web merupakan elemen krusial. Evolusi teknologi web menuntut adaptasi terhadap berbagai pendekatan pemrograman, terutama penggunaan PHP Native dan Framework Laravel yang memiliki karakteristik berbeda[1]. Perbedaan karakteristik kedua pendekatan tersebut memengaruhi efisiensi pengembangan proyek[2]. Kondisi tersebut menyebabkan variasi tingkat produktivitas mahasiswa selama proses pengembangan web berlangsung[3].

Pengukuran produktivitas mahasiswa pada penelitian sebelumnya umumnya dilakukan melalui pendekatan deskriptif berbasis kuesioner maupun analisis statistik sederhana[4]. Metode tersebut belum mampu memberikan klasifikasi produktivitas secara objektif berdasarkan pola data yang kompleks. Konteks ini masih relatif terbatas, khususnya untuk mengklasifikasi tingkat produktivitas mahasiswa pengembangan web berbasis PHP Native maupun Laravel[5]. Solusi komputasional diperlukan untuk memetakan pola produktivitas lebih akurat

Support Vector Machine merupakan salah satu algoritma klasifikasi yang efektif pada data berdimensi tinggi serta jumlah sampel terbatas[6]. Algoritma ini memiliki kemampuan membentuk *hyperplane* optimal guna memisahkan kelas secara maksimal. Penelitian ini menerapkan *Support Vector Machine* untuk mengklasifikasi tingkat produktivitas mahasiswa ke dalam kategori rendah, sedang, dan tinggi berdasarkan data kuesioner yang telah melalui proses pelabelan menggunakan metode persentil P^{33} - P^{66} . [7]

Penelitian ini bertujuan mengklasifikasi tingkat produktivitas mahasiswa dalam pengembangan web menggunakan *Support Vector Machine* serta membandingkan performanya terhadap *Logistic Regression* maupun *Decision Tree*[8]. Kontribusi penelitian terletak pada penerapan pendekatan *machine learning* untuk klasifikasi produktivitas mahasiswa berbasis data persepsional, analisis komparatif framework PHP dengan PHP Native maupun Laravel, serta identifikasi indikator dominan yang mempengaruhi produktivitas pengembangan web pada lingkungan akademik[9].

2. TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Berbagai penelitian telah mengkaji penggunaan framework PHP, khususnya Laravel serta membandingkan dengan pendekatan PHP Native dalam konteks pengembangan sistem berbasis web. Beberapa studi menunjukkan bahwa Laravel memiliki keunggulan pada konsistensi terstruktur proyek, penerapan arsitektur *Model-View-Control* (MVC), serta ketersediaan fitur bawaan seperti routing otomatis dan ORM yang mendukung efisiensi proses pengembangan, meskipun PHP Native cenderung lebih ringan dan unggul pada eksekusi sistem sederhana karena minimnya lapisan abstraksi[10].

Penelitian lain meninjau efektivitas Laravel dan PHP Native dalam lingkungan akademik dengan membandingkan efisiensi waktu pengembangan, tingkat kesalahan serta jumlah baris kode yang dihasilkan. Hasil Penelitian menunjukkan bahwa Laravel mampu meningkatkan efisiensi pengembangan

aplikasi web dan mengurangi kompleksitas penulisan kode, sedangkan PHP *Native* memberikan fleksibilitas penuh dalam pengaturan struktur dan alur logika program[11].

Kajian kompatif terhadap beberapa *framework* PHP menggunakan pendekatan statistik ANOVA menunjukkan bahwa setiap *framework* memiliki karakteristik performa yang berbeda pada aspek *response time*, *throughput*, serta penggunaan sumber daya sistem. Hasil penelitian tersebut menegaskan bahwa pemilihan *framework* perlu disesuaikan dengan kebutuhan sistem dan konteks penggunaan[12].

Kajian terkait *framework* web dari beberapa penelitian menerapkan metode *Support Vector Machine* dalam klasifikasi performa mahasiswa berbasis data pendidikan. Hasil penelitian menunjukkan bahwa SVM mampu memberikan tingkat akurasi yang tinggi pada *dataset* berskala kecil hingga menengah karena kemampuannya membentuk *hyperplane* optimal, termasuk pada data dengan distribusi tidak linear melalui penggunaan kernel tertentu[13].

Penelitian lain membandingkan *framework* berbasis Bahasa *interpreted* seperti Laravel dengan *framework* berbasis Bahasa *compiled* seperti Gin. Hasil kajian menunjukkan bahwa *framework* *interpreted* unggul pada kemudahan penggunaan serta kecepatan pengembangan, namun memiliki *overhead* performa yang lebih besar dibandingkan *framework* *compiled* yang menawarkan eksekusi lebih cepat dengan kompleksitas pengembangan yang lebih tinggi.[14].

Studi terkait efektivitas pembelajaran *framework* pengembangan web pada mahasiswa menunjukkan bahwa penggunaan *framework* modern mampu meningkatkan kecepatan penyelesaian proyek dan produktivitas, meskipun berpotensi mengurangi pemahaman logika pemrograman dasar dibandingkan pendekatan *native programming*.

2.2. Komparasi Teknis dan Beban Kognitif

Perbedaan mendasar antara PHP *Native* dan Laravel terletak pada manajemen kompleksitas. Pengembangan berbasis *native* menuntut pemrograman eksplisit untuk setiap fungsi dasar, yang meningkatkan beban kognitif pengembangan dan risiko kesalahan sintaksis[15]. Laravel mengatasi ini melalui abstraksi fitur seperti *Eloquent ORM* dan *Blade Template Engine*. Fitur-fitur ini mengeliminasi tugas repetitif, sehingga pengembang dapat memfokuskan sumber daya pada logika bisnis utama. Efisiensi ini menjadi variabel kunci dalam pengukuran produktivitas mahasiswa[16].

2.3. Indikator Produktivitas Pengembangan

Produktivitas dalam penelitian ini didefinisikan secara operasional sebagai rasio efisiensi antara fungsionalitas yang dihasilkan terhadap waktu dan upaya yang dikeluarkan. Metrik pengukuran difokuskan pada tiga indikator utama: kecepatan siklus pengembangan, stabilitas kode, dan kualitas struktur

perangkat lunak[17]. Penggunaan alat bantu (*framework*) diasumsikan berkorelasi positif dengan peningkatan ketiga indikator tersebut dibandingkan metode konvensional[18].

2.4. Urgensi Pendekatan Machine Learning

Metode evaluasi konvensional berbasis statistik deskriptif memiliki keterbatasan dalam memetakan pola hubungan *non-linear*. Pendekatan *machine learning* diperlukan untuk mengkonversi data subjektif responden menjadi klasifikasi kinerja yang objektif. Algoritma pembelajaran mesin mampu mengidentifikasi pola tersembunyi dari variabel teknis penggunaan PHP *Native* dan Laravel untuk memprediksi kategori produktivitas secara lebih akurat dibandingkan metode statistik linear sederhana[19].

2.5. Algoritma Support Vector Machine (SVM)

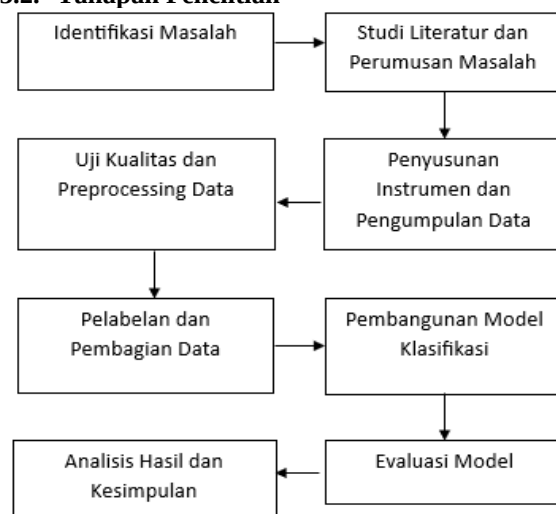
Pemilihan algoritma *Support Vector Machine* (SVM) didasarkan pada karakteristik *dataset* penelitian yang memiliki dimensi fitur tinggi namun jumlah sampel terbatas. Algoritma ini bekerja dengan memaksimalkan margin pemisahan antar kelas optimal. Penggunaan kernel *Radial Basis Function* (RBF) pada SVM diterapkan untuk menangani karakteristik data skala Likert yang cenderung tidak terpisah secara linear (*non-linear separable*), sehingga mampu menghasilkan akurasi klasifikasi yang lebih tinggi dibandingkan model linear[20].

3. METODE PENELITIAN

3.1. Desain Penelitian

Penelitian ini menggunakan pendekatan kuantitatif dengan metode eksperimen berbasis *machine learning*. Penelitian bertujuan untuk mengklasifikasi tingkat produktivitas mahasiswa dalam menggunakan PHP *Native* dan Laravel menggunakan SVM dengan pembandingan *Logistic regression* dan *Decision Tree*.

3.2. Tahapan Penelitian



Gambar 1. Alur Tahapan Penelitian

Tahapan ini disusun secara sistematis untuk memastikan proses klasifikasi berjalan terstruktur, pada Gambar 1. Diagram menunjukkan alur penelitian mulai dari indentifikasi masalah hingga analisis hasil penarikan kesimpulan. Berikut penjelasan tahapan penelitiannya

3.3. Identifikasi Masalah

Tahap awal dengan mengidentifikasi permasalahan terkait variasi tingkat produktivitas mahasiswa dalam pengembangan web menggunakan PHP *Native* dan *framework* Laravel. Permasalahan difokuskan pada belum adanya klasifikasi produktivitas berbasis data menggunakan pendekatan *machine learning*.

3.4. Studi literatur dan Perumusan Masalah

Studi literatur dilakukan untuk mengkaji teori pendukung serta penelitian terdahulu terkait *framework* PHP, produktivitas pengembangan perangkat lunak, dan algoritma *Support Vector Machine*. Hasil kajian digunakan untuk merumuskan permasalahan penelitian dan tujuan yang ingin dicapai.

3.5. Penyusunan Instrumen dan Pengumpulan Data

Penyusunan penelitian disusun dalam bentuk kuesioner skala Likert untuk mengukur indikator produktivitas mahasiswa. Pengumpulan data dilakukan dengan menyebarkan kuesioner kepada mahasiswa Teknik Informatika yang telah mempelajari atau mengembangkan aplikasi web menggunakan PHP *Native* maupun Laravel. Berikut tabel penentuan indikatornya:

Tabel 1. Indikator Produktivitas Mahasiswa

No	Indikator	Deskripsi
1	Kecepatan pengembangan	Waktu penyelesaian proyek
2	Struktur kode	Keteraturan dan modularitas
3	Kemudahan implementasi	Pengembangan fitur
4	Tingkat kesalahan	Frekuensi <i>error</i>
5	Efisiensi <i>framework</i>	Pemanfaatan fitur bawaan

3.6. Uji Kualitas dan Preprocessing Data

Data yang terkumpul diuji kualitasnya melalui pengujian validitas dan realibilitas (Cronbach Alpha ≥ 0.70), dilanjutkan dengan *preprocessing* data yang meliputi pembersihan data (*data cleaning*) dan standarisasi nilai setiap variabel agar berada pada skala yang seragam.

3.7. Pelabelan dan Pembagian Data

Pelabelan tingkat produktivitas dilakukan berdasarkan nilai rata-rata skor responden menggunakan metode persentil P^{33} - P^{66} . *Dataset* kemudian dibagi menjadi data latih dan data uji menggunakan metode *train-test split*.

3.8. Pembangunan Model Klasifikasi

Model klasifikasi dibangun menggunakan algoritma *Support Vector Machine* sebagai model utama. *Logistic regression* dan *Decision Tree* digunakan sebagai model pembanding untuk menilai efektivitas klasifikasi. *Support Vector Machine* bekerja dengan membentuk *hyperplane* optimal yang memaksimalkan margin antar kelas.

3.9. Evaluasi Model

Evaluasi model dilakukan menggunakan *confusion matrix* dan metrik evaluasi klasifikasi berupa akurasi, presisi, *recall*, dan F1-score.

Tabel 2. Metrik Evaluasi Model

Metrik	Keterangan
Akurasi	ketepatan prediksi keseluruhan
Presisi	Ketepatan prediksi per kelas
Recall	Kemampuan mendeteksi kelas
F1-Score	Harmoni presisi dan <i>recall</i>

3.10. Analisis Hasil dan Kesimpulan

Tahap akhir dilakukan dengan menganalisis hasil evaluasi model untuk menentukan performa terbaik serta mengidentifikasi perbedaan produktivitas mahasiswa antara penggunaan PHP *Native* dan Laravel. Hasil analisis digunakan sebagai dasar penarikan kesimpulan dan rekomendasi.

4. HASIL DAN PEMBAHASAN

4.1. Karakteristik Data Penelitian

Data penelitian diperoleh dari mahasiswa Program Studi Teknik Informatika Universitas Stikubank yang berpengalaman pengembangan menggunakan PHP *Native* maupun *framework* Laravel. Pengumpulan data dilakukan menggunakan instrumen kuesioner berbasis skala Likert untuk mengukur tingkat produktivitas mahasiswa.

Jumlah responden pada penelitian ini sebanyak 43 mahasiswa. Setiap responden mengisi 30 indikator produktivitas yang mempresnetasiikan aspek efisiensi waktu pengembangan, efektivitas proses kerja, serta kualitas hasil pengembangan web. Data yang terkumpul berbentuk data numerik hasil konversi skala Likert yang digunakan sebagai fitur dalam proses klasifikasi.

Tabel 3. Karakteristik Data Penelitian

Elemen	Isi
Subjek data	Mahasiswa Teknik Informatika
Jumlah data	Jumlah responden
Sumber data	Kuesioner
Skala data	Likert
Variabel	Produktivitas PHP <i>Native</i> dan Laravel
Label	Rendah, sedang, tinggi

4.2. Hasil Uji Kualitas Data

Pengujian kualitas data dilakukan untuk memastikan kelayakan *dataset* sebelum digunakan pada proses klasifikasi berbasis *machine learning*.

4.3. Hasil Uji Validitas

Uji validitas bertujuan mengetahui kemampuan setiap indikator dalam mengukur konstruk produktivitas yang diteliti. Hasil pengujian menunjukkan seluruh item kusioner memiliki nilai korelasi lebih besar dari nilai r-tabel. Kondisi tersebut menunjukkan seluruh indikator dinyatakan valid dan layak digunakan pada tahap analisis selanjutnya. Hasil uji validitas instrumen ditunjukkan pada tabel 4.

Tabel 4. Uji Validitas Instrumen

Metode	Total butir	Total valid	r-tabel
PHP Native	30	19	0.301
Framework laravel	30	20	0.301

4.4. Hasil Uji Reliabilitas

Uji reliabilitas digunakan untuk mengukur konsisten internal instrumen penelitian. Nilai Cronbach's Alpha yang diperoleh berada pada kategori reliabel. Kondisi tersebut menunjukkan instrumen memiliki tingkat konsistensi yang baik untuk mengukur produktivitas mahasiswa. Nilai reliabilitas instrumen disajikan pada tabel 5.

Tabel 5. Uji Reliabilitas Instrumen

Metode	Cronbach's Alpha	Keterangan
PHP Native	0.8074	Reliabel
Framework laravel	0.8	Reliabel

4.5. Hasil Pelabelan Tingkat Produktivitas

Pelabelan tingkat produktivitas mahasiswa dilakukan berdasarkan nilai rata-rata skor responden menggunakan metode persentil $P^{33} - P^{66}$. Metode tersebut membagi data ke dalam tiga kategori produktivitas yaitu rendah, sedang, dan tinggi.

$$P_k = \frac{k}{100} \times (N + 1)$$

Keterangan simbol:

P_k : Nilai persentil ke- k

k : Tingkat persentil (33 atau 66)

N : Jumlah data

Distribusi label menunjukkan seberapa besar mahasiswa berada pada kategori produktivitas sedang. Kategori produktivitas tinggi lebih banyak ditemukan pada mahasiswa yang menggunakan *framework* Laravel dibandingkan mahasiswa yang menggunakan *PHP Native*. Temuan tersebut mengindikasikan adanya perbedaan tingkat produktivitas berdasarkan pendekatan pengembangan web yang digunakan. Distribusi tingkat produktivitas mahasiswa ditunjukkan pada Tabel 6.

Tabel 6. Distribusi Tingkat Produktivitas Mahasiswa

Kategori	Jumlah Responden
Sedang	15
Tinggi	15
Sedang	13

4.6. Hasil Klasifikasi Menggunakan Support Vector Machine

Support Vector Machine digunakan sebagai model utama dalam mengklasifikasi tingkat produktivitas mahasiswa. Pelatihan model dilakukan menggunakan data latih dan pengujian dilakukan menggunakan data uji sebagaimana ditetapkan pada tahap metode penelitian.

$$f(x) = w \cdot x + b$$

Keterangan simbol

$f(x)$: Fungsi keputusan

w : Vektor bobot

x : Vektor bias

b : Bias

Hasil pengujian menunjukkan *Support Vector Machine* mampu mengklasifikasi tingkat produktivitas mahasiswa secara efektif pada setiap kelas. Model menghasilkan performa yang stabil meskipun *dataset* memiliki jumlah sampel terbatas. Visualisasi hasil klasifikasi menggunakan *confusion matrix* ditampilkan pada Tabel 7.

Tabel 7 Confusion Matrix Model SVM

	Prediksi rendah	Prediksi sedang	Prediksi tinggi
Aktual rendah	2	1	0
Aktual sedang	0	2	1
Aktual tinggi	0	1	2

4.7. Perbandingan Performa Model Klasifikasi

Evaluasi performa model dilakukan dengan membandingkan *Support Vector Machine* memiliki nilai akurasi tertinggi dibanding model pembandingan.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Keterangan simbol:

TP : True positive

TN : True negative

FP : False positive

FN : False negative

Logistic Regression menunjukkan performa yang stabil pada sebagian kelas. *Decision Tree* menunjukan variasi performa akibat distribusi kelas yang tidak seimbang. Perbandingan performa model klasifikasi disajikan pada Tabel 8.

Tabel 8. Perbandingan Performa Model Klasifikasi

Algoritma	Akurasi	F1-score
Logistic regression	0.777778	0.774603
SVM(RBF)	0.666667	0.679365
Decision tree	0.444444	0.433333

4.8. Analisis Perbandingan Produktivitas PHP Native dan Laravel

Analisis hasiln klasifikasi menunjukan mahasiswa yang menggunakan *Framework* Laravel cenderung memiliki tingkat produktivitas lebih tinggi dibandingkan mahasiswa yang menggunakan PHP *Native*. Struktur *framework* yang konsisten, fitur otomatisasi, serta penerapan arsitektur MVC berkontribusi terhadap percepatan proses pengembangan web.

Pendekatan PHP *Native* memberikan fleksibilitas tinggi dalam penulisan kode. Waktu pengembangan yang lebih lama serta risiko kesalahan yang lebih besar muncul akibat tidak adanya pola kerja baku pada struktur proyek. Perbandingan tingkat produktivitas dilakukan berdasarkan nilai rata-rata masing-masing kelompok pengembangan.

$$\bar{X}_g = \frac{\sum X_g}{N_g}$$

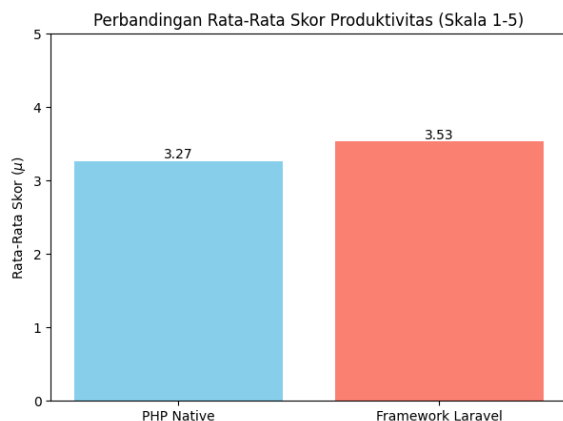
Keterangan simbol:

$\bar{X}_g$: Rata-rata produktivitas kelompok

X_g : Total skor kelompok

N_g : Jumlah responden kelompok

Berikut hasil dari perhitungan diatas yang ditampilkan dalam Gambar 2.



Gambar 2. Perbandingan Tingkat Produktivitas

4.9. Pembahasan Temuan Penelitian

Temuan penelitian menunjukan pendekatan *machine learning* efektif digunakan untuk mengklasifikasi tingkat produktivitas mahasiswa berbasis data persepsional, *Support Vector Machine* menunjukan keunggulan dalam menangani *dataset* berdimensi tinggi dengan jumlah sampel terbatas.

Hasil penelitian menguatkan bahwa penggunaan *framework* modern seperti Laravel mampu meningkatkan produktivitas mahasiswa dalam pengembangan web pada lingkungan akademik, terutama aspek efisiensi waktu dan konsistensi struktur kode.

5. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengklasifikasi tingkat produktivitas mahasiswa Teknik Informatika Unisbank dalam pengembangan web menggunakan

pendekatan *machine learning* berbasis *Support Vector Machine*. Data penelitian berasal dari kuesioner skala Likert yang mempresentasikan efisiensi waktu, efektivitas proses, serta kualitas hasil pengembangan. Hasil klasifikasi menunjukan *Support Vector Machine* mampu mengelompokkan tingkat produktivitas ke dalam kategori rendah, sedang, dan tinggi dengan performa yang stabil pada *dataset* berukuran terbatas serta menunjukan kinerja lebih baik dibandingkan *Logistic Regression* dan *Decision Tree*. Analisis perbandingan produktivitas menunjukan mahasiswa yang menggunakan *framework* Laravel cenderung memiliki tingkat produktivitas lebih tinggi dibandingkan mahasiswa yang menggunakan PHP *Native* akibat struktur *framework* yang konsisten, fitur otomatisasi, serta penerapan arsitektur MVC. Penelitian selanjutnya disarankan melibatkan jumlah responden yang lebih besar, menambahkan variabel objektif, serta menguji algoritma *machine learning* lain atau optimasi parameter *Support Vector Machine* untuk meningkatkan performa klasifikasi. Intitusi pendidikan disarankan mempertimbangkan pemanfaatan *framework* modern sebagai media pembelajaran pengembangan web untuk meningkatkan produktivitas mahasiswa secara berkelanjutan.

DAFTAR PUSTAKA

- [1] R. Y. Endra, Y. Aprilinda, Y. Y. Dharmawan, and W. Ramadhan, "Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 11, no. 1, p. 48, Jun. 2021, doi: 10.36448/expert.v11i1.2012.
- [2] K. N. Feblia *et al.*, "Studi Perbandingan Penerapan Pola Model-View-Controller (MVC) dalam Lima Framework Web Populer: Laravel, Django, Ruby on Rails, Asp. net MVC, Spring MVC," *Indonesian Journal of Computer Science and Engineering*, vol. 2, no. 01, pp. 16–22, 2025.
- [3] H. E. Abdelkader, A. G. Gad, A. A. Abohany, and S. E. Sorour, "An Efficient Data Mining Technique for Assessing Satisfaction Level With Online Learning for Higher Education Students during the COVID-19," *IEEE Access*, vol. 10, pp. 6286–6303, 2022, doi: 10.1109/ACCESS.2022.3143035.
- [4] H. Effendy, B. R. Machmoed, and A. Rasyid, "Pengukuran dan Analisis Produktivitas Menggunakan Metode Objective Matrix (OMAX) (Studi Kasus: di PDAM Kabupaten Gorontalo)," *Jambura Industrial Review*, vol. 1, no. 1, p. 2021, doi: 10.XXXXX/jirev.vXiX.XX-XX.
- [5] R. C. Chen, C. Dewi, S. W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J Big Data*, vol. 7, no. 1, Dec. 2020, doi: 10.1186/s40537-020-00327-4.

- [6] X. Shu and Y. Ye, "Knowledge Discovery: Methods from data mining and machine learning," *Soc Sci Res*, vol. 110, p. 102817, Feb. 2023, doi: 10.1016/j.ssresearch.2022.102817.
- [7] B. I. Nugroho, N. A. Santoso, and A. A. Murtopo, "Prediksi Kemampuan Akademik Mahasiswa dengan Metode Support Vector Machine," *remik*, vol. 7, no. 1, pp. 177–188, Jan. 2023, doi: 10.33395/remik.v7i1.12010.
- [8] C. Peng and Q. Cheng, "Discriminative Ridge Machine: A Classifier for High-Dimensional Data or Imbalanced Data," *IEEE Trans Neural Netw Learn Syst*, vol. 32, no. 6, pp. 2595–2609, Jun. 2021, doi: 10.1109/TNNLS.2020.3006877.
- [9] D. Yuswanto Jaya and M. Ulfah Siregar, "Analisis Perbandingan Learnability Antara Framework dan Native PHP pada Mahasiswa Informatika Universitas XYZ."
- [10] Y. A. Politeknik and N. Malang, "Laravel framework and native PHP: Comparison in the creation of rest API", doi: 10.31940/matrix.v14i2.66-73.
- [11] N. Andayaningtyas, "EFEKTIVITAS PENGGUNAAN BAHASA PEMROGRAMAN PHP LARAVEL DENGAN PHP NATIVE PADA PENGEMBANGAN WEBSITE," *Consilium: Education and Counseling Journal*, vol. 4, no. 1, p. 366, Mar. 2024, doi: 10.36841/consilium.v4i1.6459.
- [12] Moch. Faisal and I. K. D. Nuryana, "Comparative Analysis Of Laravel, Lumen, Guzzle, Leaf, and Slim Framework Performance On Rest API Using One Way Anova," *Journal of Emerging Information Systems and Business Intelligence (JEISBI)*, vol. 6, no. 2, pp. 77–83, Jul. 2025, doi: 10.26740/jeisbi.v6i2.66146.
- [13] M. H. Mohd Zaki, M. A. Mohd Aziz, S. Sulaiman, and N. Hambali, "Student Performance Classification using Support Vector Machine (SVM) with Polynomial Kernel on Online Student Activities," *Journal of Electrical & Electronic Systems Research*, pp. 80–90, Oct. 2023, doi: 10.24191/jeesr.v23i1.009.
- [14] J. A. Yang and S. A. Aklani, "PERFORMANCE ANALYSIS BETWEEN INTERPRETED LANGUAGE-BASED (LARAVEL) AND COMPILED LANGUAGE-BASED (GIN) WEB FRAMEWORKS," *Computer Based Information System Journal*, vol. 11, no. 1, pp. 12–16, Mar. 2023, doi: 10.33884/cbis.v11i1.6583.
- [15] B. Pasaribu and W. Susanti, "Sistem Informasi Pengajuan Rancangan Usulan Penelitian Menggunakan PHP Native dan Bot Telegram," *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi*, vol. 3, no. 1, 2021, [Online]. Available: <http://www.php.net>.
- [16] K. Pradipta Wistika, D. Pramana, and N. W. Setiasih, "Sistem Informasi Pemesanan Jasa Fotografi pada Julian Photography Menggunakan Framework Laravel," *Jurnal Informasi dan Teknologi*, pp. 240–249, May 2023, doi: 10.37034/jidt.v5i1.301.
- [17] S. MacNeil *et al.*, "Experiences from Using Code Explanations Generated by Large Language Models in a Web Software Development E-Book," Nov. 2022, [Online]. Available: <http://arxiv.org/abs/2211.02265>
- [18] A. Anoesyirwan, H. Madiistriyatno, and S. Mutmainnah, "Peningkatan Kualitas Manajemen Publikasi Ilmiah Menggunakan Metode Agile," *ADI Bisnis Digital Interdisiplin Jurnal*, vol. 1, no. 2 Desember, pp. 31–39, Nov. 2020, doi: 10.34306/abdi.v1i2.99.
- [19] F. Nie, Z. Hao, and R. Wang, "Multi-Class Support Vector Machine with Maximizing Minimum Margin," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 13, pp. 14466–14473, Mar. 2024, doi: 10.1609/aaai.v38i13.29361.
- [20] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine Learning*, Elsevier, 2020, pp. 101–121. doi: 10.1016/B978-0-12-815739-8.00006-7