

Evaluasi Penilaian Otomatis Pemrograman Web Laravel pada Platform LAIBA

¹Dian Hanifudin Subhi, ²Putra Prima Arhandi, ³Annisa Taufika Firdausi,

⁴Putranda Bagus Cahya Lumintang

^{1,2,3,4}Politeknik Negeri Malang

dhanifudin@polinema.ac.id, putraprima@polinema.ac.id, annisa.taufika@polinema.ac.id,

putranda.bagus@gmail.com

Submit : 28 Okt 2025 | Diterima : 30 Nov 2025 | Terbit : 05 Des 2025

ABSTRAK

Laravel adalah *framework* PHP *open-source* yang memudahkan pengembangan aplikasi web, namun beberapa sistem pembelajaran yang ada belum memberikan umpan balik yang efektif. Learning Application in Balanced Assessment (LAIBA) diusulkan sebagai aplikasi pembelajaran berbasis Laravel dengan fitur penilaian otomatis dan umpan balik terarah untuk meningkatkan pemahaman mahasiswa. Dalam sistem ini, mahasiswa menyelesaikan studi kasus menggunakan Laravel, kemudian mengirimkan kode untuk diperiksa dan dinilai oleh sistem. Data penilaian digunakan untuk mengidentifikasi topik yang perlu dipahami lebih lanjut. Pengujian dilakukan dengan metode pre-test dan post-test, yang menunjukkan peningkatan pemahaman mahasiswa secara keseluruhan. Hasil pengujian secara umum, LAIBA mampu meningkatkan pemahaman mahasiswa berdasarkan nilai rata-rata dengan signifikansi sebesar 0,294. Walaupun belum dapat dinyatakan mendapatkan hasil peningkatan secara signifikan secara umum, tetapi untuk data cluster nilai rata-rata rendah naik sebesar 26% dan menunjukkan peningkatan performa yang signifikan pada kelompok tersebut.

Kata Kunci: Laravel, Penilaian Otomatis, Pemrograman Web dan Umpan Balik.

PENDAHULUAN

Pengembangan web saat ini menjadi salah satu kebutuhan utama bagi para developer karena mampu menghasilkan aplikasi yang mudah digunakan dan dapat diakses di berbagai platform. Salah satu bahasa pemrograman yang mendukung pengembangan web adalah PHP, yang populer karena sifatnya yang dinamis, fleksibel, mudah dipelajari, dan digunakan. Seiring meningkatnya permintaan untuk pengembangan web, efisiensi, keandalan, pemeliharaan, dan skalabilitas menjadi aspek penting, yang menjadikan *framework* PHP sebagai alat penting dalam pengembangan web (Laaziri et al., 2019). Laravel, salah satu *framework* PHP yang paling sering digunakan, dirancang untuk memudahkan dan mempercepat pengembangan aplikasi web dengan menyediakan berbagai fitur bawaan (Subecz, 2021).

Laravel mengusung arsitektur *Model-View-Controller* (MVC), yang merupakan pola standar dalam domain web. Arsitektur ini dikenal karena kemampuannya dalam skalabilitas, pemeliharaan, serta memisahkan logika bisnis dari antarmuka pengguna (Jailia et al., 2016). Meskipun sudah banyak sistem pembelajaran yang menggunakan modul untuk membantu mahasiswa mempelajari Laravel, sistem tersebut sering kali tidak memberikan umpan balik yang efektif, sehingga sulit bagi mahasiswa untuk mengetahui seberapa baik mereka memahami konsep yang diajarkan (Harms et al., 2015).

Dalam rangka meningkatkan pemahaman mahasiswa, aplikasi pembelajaran berbasis Laravel dengan sistem penilaian otomatis serta umpan balik yang berarti diusulkan. Sistem ini tidak hanya membantu mahasiswa dalam menyelesaikan tugas, tetapi juga menyediakan latihan berkelanjutan yang memungkinkan pemahaman yang lebih mendalam dan berdampak positif pada pencapaian akademik (Haldeman et al., 2018). Umpan balik yang baik membantu mahasiswa dalam mengenali kesalahan dan memperbaiki pemahaman, serta memberikan manfaat bagi pengajar dalam mengidentifikasi kesulitan belajar siswa (Higgins et al., 2005).

Beberapa penelitian sebelumnya mengenai sistem penilaian otomatis dan autograding menunjukkan bahwa pemberian umpan balik otomatis dalam tugas pemrograman memiliki dampak yang signifikan terhadap pemahaman siswa. Penelitian (Hao et al., 2022) menunjukkan bahwa penggunaan sistem penilaian otomatis berbasis GitHub dan alat Continuous Integration membantu mengurangi waktu penilaian dan memberikan umpan balik yang berguna bagi siswa. Selain itu, penelitian (Delgado-Pérez & Medina-Bulo, 2020) menekankan pentingnya umpan balik spesifik dalam tugas pemrograman C/C++, yang secara signifikan meningkatkan prestasi akademik siswa. Oleh karena itu, aplikasi pembelajaran ini diusulkan untuk memberikan penilaian formatif dan sumatif pada pemrograman Laravel MVC, guna membantu mahasiswa memahami materi dengan lebih baik.

Selain itu, (González-Carrillo et al., 2021; Kavita, 2025) mengembangkan sistem penilaian otomatis pada file Jupyter Notebook untuk pembelajaran kecerdasan buatan, yang mengintegrasikan penilaian sumatif dan formatif. Sistem tersebut memberikan umpan balik yang jelas dan membantu siswa dalam mengidentifikasi kesalahan dan memperbaikinya, serta menerima respons positif dari pengguna.

Lebih lanjut, (Hagerer et al., 2021) melakukan analisis perbandingan pembelajaran pemrograman sebelum dan sesudah penerapan sistem autograder. Hasil penelitian mereka menunjukkan bahwa penggunaan sistem penilaian otomatis meningkatkan kemampuan pemecahan masalah pemrograman, interaksi antara pengajar dan siswa, kualitas materi, serta keberhasilan pembelajaran secara keseluruhan. Selain itu, (Grenander et al., 2021) mengeksplorasi pengembangan pembuatan umpan balik otomatis yang dipersonalisasi dalam sistem tutor cerdas untuk memberikan umpan balik yang kontekstual dan efektif.

Berdasarkan kajian tersebut, penelitian ini mengembangkan sistem pembelajaran pemrograman web Laravel pada platform LAIBA (*Learning Application in Balanced Assessment*) yang menerapkan sistem penilaian otomatis dengan umpan balik formatif dan sumatif secara berkelanjutan. Sistem ini diharapkan mampu mendukung proses pembelajaran interaktif dan meningkatkan prestasi mahasiswa dalam memahami pemrograman Laravel.

TINJAUAN PUSTAKA

Balanced Assessment System

Balanced Assessment System dipahami sebagai sistem penilaian kemampuan, keterampilan, dan/atau kompetensi peserta didik yang mengintegrasikan secara proporsional penilaian formatif dan sumatif untuk memenuhi kebutuhan informasi tentang proses sekaligus capaian pembelajaran (Gong, 2010). Penilaian formatif berfokus pada aktivitas penilaian yang dilakukan pendidik selama proses belajar untuk menyediakan informasi yang bermanfaat bagi perbaikan pembelajaran, sedangkan penilaian sumatif digunakan untuk mengetahui sejauh mana pengetahuan dan keterampilan yang telah dikuasai siswa pada akhir suatu periode pembelajaran (Sumintono, 2016). Suatu sistem penilaian disebut seimbang apabila penilaian dirancang selaras dengan tujuan pembelajaran yang dirumuskan secara jelas, memanfaatkan berbagai sumber bukti secara komprehensif untuk mendukung pengambilan keputusan pendidikan, serta mampu merekam perkembangan peserta didik secara berkelanjutan dari waktu ke waktu (Marion et al., 2019).

Pada sistem pembelajaran yang dikembangkan dalam penelitian, kerangka ini diadopsi untuk menjaga keseimbangan antara penilaian formatif dan sumatif. Penilaian formatif diwujudkan melalui pemantauan proses mahasiswa saat menyelesaikan latihan berbasis studi kasus yang disediakan dalam aplikasi, sementara penilaian sumatif direpresentasikan oleh penilaian akhir terhadap hasil pengerjaan latihan tersebut. Kombinasi kedua jenis penilaian ini menghasilkan umpan balik yang lebih kaya dan terarah, sehingga diharapkan dapat mendukung peningkatan prestasi mahasiswa sekaligus memperdalam pemahaman mereka terhadap materi pembelajaran.

CI/CD

Continuous Integration (CI) dipahami sebagai praktik ketika pengembang secara berkala mengintegrasikan perubahan kode ke dalam sebuah repositori bersama, sehingga setiap perubahan dapat segera diuji dan diverifikasi. Continuous Delivery (CDE) merujuk pada proses otomatisasi

pelepasan artefak perangkat lunak yang telah teruji ke repositori pusat atau lingkungan rilis, sedangkan Continuous Deployment (CD) menekankan penyebaran perubahan yang telah tervalidasi ke lingkungan produksi atau layanan komputasi awan secara otomatis setelah proses pembangunan aplikasi dinyatakan berhasil (Purohit, 2020). Berbagai tahapan tersebut dapat diotomatisasi dalam bentuk pipeline CI/CD, misalnya melalui alur kerja (*workflow*) di GitHub, untuk mendukung proses pengiriman perangkat lunak yang berkelanjutan dan minim intervensi manual (Chandrasekara & Herath, 2021).

Dalam sistem pembelajaran yang dikembangkan pada penelitian ini, pipeline CI/CD dimanfaatkan untuk menjalankan *unit testing* dan mekanisme penilaian otomatis setiap kali mahasiswa mengirimkan atau memperbarui solusi pemrograman mereka.

Laravel

Laravel merupakan salah satu framework PHP open-source yang banyak digunakan untuk pengembangan aplikasi web karena menyediakan beragam fitur bawaan yang mempercepat dan mempermudah proses pengembangan (Subecz, 2021). Framework ini dikenal efisien, memiliki kurva belajar yang relatif bersahabat, dokumentasi yang lengkap, serta ekosistem komunitas yang aktif mendukung pengembang (Laaziri et al., 2019). Laravel juga menyediakan berbagai komponen umum yang sering dibutuhkan dalam aplikasi web, seperti mekanisme otentikasi, routing, manajemen sesi, caching, fasilitas migrasi basis data, serta dukungan pengujian unit yang terintegrasi. Kombinasi fitur-fitur tersebut memberikan landasan yang kuat bagi pengembang untuk membangun aplikasi berskala kompleks secara lebih terstruktur (Chen et al., 2017).

Secara arsitektural, Laravel mengadopsi pola *Model View Controller* (MVC). Pola MVC telah menjadi standar dalam pengembangan aplikasi web karena mendukung skalabilitas, kemudahan pemeliharaan, kemampuan penggunaan kembali (*reusability*), serta kemudahan pengujian (Jivani et al., 2013). Pemisahan yang jelas antara logika bisnis (*model* dan *controller*) dan antarmuka pengguna (*view*) memungkinkan penyesuaian dan pengembangan aplikasi dilakukan secara lebih fleksibel dan terorganisasi (Jailia et al., 2016). Dalam penelitian ini, Laravel tidak hanya dimanfaatkan sebagai *backend* utama sistem pembelajaran, tetapi sekaligus menjadi materi atau topik pembelajaran yang dipraktikkan oleh mahasiswa dalam pengembangan aplikasi web.

METODE PENELITIAN

Penelitian ini bertujuan untuk mengevaluasi efektivitas sistem pembelajaran LAIBA dalam meningkatkan pemahaman mahasiswa terhadap framework Laravel dengan menggunakan penilaian otomatis dan umpan balik berbasis pengujian kode. Pendekatan yang digunakan adalah eksperimen kuantitatif, di mana data diperoleh melalui pre-test dan post-test yang diberikan kepada mahasiswa, serta analisis statistik untuk mengukur perubahan pemahaman mahasiswa sebelum dan setelah menggunakan sistem pembelajaran.

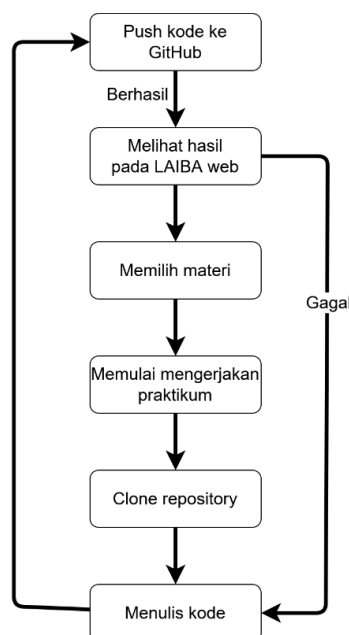
1. Pengumpulan Data

Proses pengumpulan data dilakukan dalam dua tahap, yaitu pre-test dan post-test. Pada pre-test mahasiswa mengerjakan soal yang berkaitan dengan materi Laravel (Route, Controller, View, dan Blade Template) sebelum menggunakan sistem LAIBA. Setelah menggunakan LAIBA untuk mengerjakan praktikum, mahasiswa mengikuti post-test yang berisi soal yang sama untuk mengukur peningkatan pemahaman mereka.

Gambar 1 menunjukkan arsitektur sistem, di mana mahasiswa mengakses materi pembelajaran melalui aplikasi LAIBA, mengerjakan tugas di GitHub, dan mengirimkan hasil pekerjaan mereka untuk diuji menggunakan GitHub Actions.

2. Pengujian dan Evaluasi Hasil

Data pre-test dan post-test dianalisis menggunakan Uji Kolmogorov-Smirnov untuk menguji normalitas data. Selanjutnya, untuk mengukur apakah ada perbedaan yang signifikan antara pre-test dan post-test, dilakukan *paired-samples t-test*. Hasil dari uji ini digunakan untuk



Gambar 2 Alur Sistem Pembelajaran

HASIL DAN PEMBAHASAN

Uji Pre-test dan Post-test

Evaluasi sistem dilakukan pada mahasiswa tingkat 2 Jurusan Teknologi Informasi yang sedang melakukan pembelajaran mata kuliah Pemrograman Web Lanjut. Mahasiswa mempelajari framework Laravel dengan materi Route, Controller, View dan Blade Template seperti dijabarkan pada Tabel 2. Materi dan pengenalan sistem dilakukan sebelum dilakukan pengambilan nilai pre-test untuk pengambilan nilai awal.

Pengambilan nilai post-test dilakukan setelah mahasiswa menggunakan aplikasi LAIBA sebagai sistem pembelajaran. Mahasiswa akan mendapatkan umpan balik ketika mengerjakan praktikum dan aktifitas pengerjaannya akan disimpan di dalam basis data. Nilai post-test digunakan untuk mengetahui pengaruh sistem pembelajaran LAIBA terhadap pembelajaran web Laravel. Hasil nilai pre-test dan post-test dari mahasiswa, dapat dilihat pada Tabel 1.

Tabel 1 Nilai Pre-test dan Post-test Mahasiswa

No.	Pre-test		Post-test	
	Skor	Nilai	Skor	Nilai
1	7	47	12	80
2	12	80	8	53
3	11	73	11	73
4	9	60	12	80
5	14	93	11	73
6	15	100	15	100
7	13	87	13	87
8	9	60	13	87
9	14	93	14	93
10	14	93	14	93
11	13	87	15	100
12	11	73	14	93
13	10	67	9	60
14	15	100	14	93
15	7	47	10	67

Tabel 2 Daftar Topik Praktikum Laravel pada LAIBA

Judul	Tipe	Skenario Pengujian
Route	Praktikum	<ul style="list-style-type: none"> - Cek apakah route index ("/") menampilkan tulisan "Hello World". - Cek apakah route "about" menampilkan tulisan nama dan nim mahasiswa. - Cek apakah route "page" dengan parameter menampilkan tulisan "Halaman dengan ide 1".
Controller	Praktikum	<ul style="list-style-type: none"> - Cek apakah route indeks ("/") dengan controller "HomeController" dapat diakses. - Cek apakah route "about" dengan single action controller "AboutController" dapat diakses. - Cek berulang apakah route "page" dengan controller berparameter menampilkan tulisan "Hello World Page 1" sampai "Hello World Page 5".
View	Praktikum	<ul style="list-style-type: none"> - Menampilkan halaman "home" - Menampilkan halaman "page" - Cek apakah key "email" telah dipanggil dan ditampilkan di halaman "About" - Cek apakah key "halaman" telah dipanggil dan ditampilkan di halaman "page"
Blade Template	Praktikum	<ul style="list-style-type: none"> - Cek apakah bagian "Header" telah ditampilkan - Cek apakah bagian "Footer" telah ditampilkan - Cek apakah bagian halaman dari view "about" dan "home" dapat ditampilkan dengan benar - Cek apakah halaman "about" menggunakan view "main" sebagai kerangka utama halaman
Kuis	Kuis	<ul style="list-style-type: none"> - Cek apakah route utama ("/") menampilkan fungsi "index" dari controller "MainController". - Cek apakah route "about" menampilkan fungsi "about" dari controller "MainController". - Cek apakah route mengirim data "Home" dari key "title" ke dalam alamat utama ("/") - Cek apakah route mengirim data "About" dari key "title" ke dalam alamat "about" - Cek apakah route mengirim data "TI Fast" dari key "nama" ke dalam alamat "about" - Cek apakah view "master" menjadi kerangka utama dari tampilan halaman "home" dan "about" - Cek apakah alamat utama menampilkan bagian "Header". - Cek apakah alamat utama menampilkan bagian "Footer". - Cek apakah halaman "About" menampilkan data dari key "page_name" - Cek apakah halaman "Home" menampilkan data dari key "page_name"

Analisis Data Clustering

Untuk analisis lebih mendalam, data pre-test dikelompokkan berdasarkan nilai median, yang diperoleh sebesar 80. Dua kelompok terbentuk seperti pada Tabel 3: cluster rendah (nilai pre-test antara 47 dan 73) sejumlah 7 buah data dan cluster atas (nilai pre-test antara 80 dan 100) sejumlah 8 data. Masing-masing cluster dianalisis menggunakan uji *paired-samples t test* untuk melihat apakah terdapat perbedaan yang signifikan antara nilai pre-test dan post-test dalam masing-masing kelompok.

Tabel 3 Tabel Clustering Berdasarkan Median

No.	Pre-test		Post-test	
	Skor	Nilai	Skor	Nilai
1	7	47	12	80
2	7	47	10	67
3	9	60	12	80
4	9	60	13	87
5	10	67	9	60
6	11	73	11	73
7	11	73	14	93
8	12	80	8	53
9	13	87	13	87
10	13	87	15	100
11	14	93	11	73
12	14	93	14	93
13	14	93	14	93
14	15	100	15	100
15	15	100	14	93

Tabel 4 Statistik rata-rata cluster rendah

Paired Samples Statistics				
	Mean	N	Std. Deviation	Std. Error Mean
pre test rendah	61.0000	7	10.93923	4.13464
post test rendah	77.1429	7	11.39340	4.30630

Tabel 5 Statistik rata-rata cluster tinggi

Paired Samples Statistics				
	Mean	N	Std. Deviation	Std. Error Mean
pre test tinggi	90.0000	8	11.68638	4.13176
post test tinggi	76.5000	8	19.25765	6.80861

Pada cluster rendah, nilai rata-rata pre-test adalah 61, dan rata-rata nilai post-test adalah 77,14 seperti terlihat Tabel 4. Uji statistik pada

Tabel 6 menunjukkan nilai signifikansi sebesar 0,025, yang lebih kecil dari 0,05, sehingga dapat disimpulkan bahwa terdapat peningkatan yang signifikan pada kelompok nilai rendah setelah menggunakan sistem pembelajaran LAIBA.

Pada cluster tinggi, nilai rata-rata pre-test adalah 90, sementara rata-rata post-test adalah 76,5 seperti dipaparkan pada Tabel 5. Hasil uji statistik menunjukkan nilai signifikansi sebesar 0,293, yang lebih besar dari 0,05, sehingga tidak ada peningkatan signifikan pada kelompok ini. Pada Tabel 7 menunjukkan analisis statistik pada cluster atas, yang memperlihatkan hasil uji statistik dengan nilai signifikansi 0,293, yang mengindikasikan tidak adanya perubahan yang signifikan pada kelompok nilai tinggi.

Tabel 6 Uji statistik akhir cluster rendah

Mean	Std. Deviation	Std. Error Mean	Paired Differences		t	df	Sig. (2-tailed)
			95% Confidence Interval of the Difference				
			Lower	Upper			

pre_test rendah -	-16.14286	14.39246	5.43984	-29.45366	-2.83205	-2.968	6	.025
post test rendah								

Tabel 7 Uji statistik akhir cluster atas

	Paired Differences				t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference Lower Upper			
pre_test tinggi - post test tinggi	5.12500	12.74405	4.50570	-5.52929 15.77929	1.137	7	.293

Pembahasan

Hasil analisis menunjukkan bahwa sistem LAIBA berperan penting dalam meningkatkan pemahaman mahasiswa, terutama pada kelompok dengan nilai pre-test rendah. Kelompok ini menunjukkan peningkatan yang signifikan pada nilai post-test, seperti yang terlihat pada Tabel 6, dengan nilai signifikansi 0,025 yang menunjukkan bahwa LAIBA efektif untuk mahasiswa yang memulai dengan pemahaman yang lebih rendah.

Sebaliknya, pada kelompok dengan nilai pre-test tinggi (cluster atas), meskipun ada penurunan rata-rata nilai post-test menjadi 76,5 dari 90 pada pre-test, penurunan ini tidak signifikan secara statistik (signifikansi 0,293) sebagaimana terlihat pada Tabel 7. Hal ini mengindikasikan adanya *ceiling effect*, di mana mahasiswa yang sudah memiliki pemahaman yang tinggi pada awalnya tidak dapat menunjukkan peningkatan yang signifikan lagi. Fenomena ini adalah hal yang wajar, karena mereka sudah berada pada tingkat pemahaman yang hampir maksimal.

Secara keseluruhan, meskipun perubahan nilai rata-rata post-test tidak signifikan untuk semua peserta secara keseluruhan, sistem LAIBA terbukti memberikan manfaat yang signifikan bagi mahasiswa yang membutuhkan lebih banyak dukungan dalam memahami materi. Sistem ini juga menunjukkan keterbatasan efektivitasnya bagi mahasiswa yang sudah memiliki pemahaman yang baik sejak awal.

KESIMPULAN

Berdasarkan implementasi sistem dan hasil pengujian, dapat disimpulkan bahwa LAIBA berkontribusi positif terhadap pemahaman mahasiswa dalam pembelajaran Laravel. Secara agregat, peningkatan yang terjadi memang belum signifikan secara statistik, dengan nilai signifikansi 0,294. Namun, ketika data dianalisis lebih rinci berdasarkan kategori kemampuan awal, terlihat bahwa sistem ini memberikan dampak yang lebih kuat pada kelompok mahasiswa dengan rata-rata nilai awal yang lebih rendah.

Pada kelompok tersebut, terjadi peningkatan pemahaman yang signifikan, tercermin dari kenaikan rata-rata nilai yang didukung oleh nilai signifikansi 0,025. Sementara itu, pada mahasiswa dengan pemahaman awal yang sudah tinggi, rata-rata nilai justru sedikit menurun. Fenomena ini dapat dijelaskan sebagai keterbatasan ruang peningkatan (*ceiling effect*), mengingat mahasiswa tersebut telah berada pada tingkat penguasaan materi yang relatif tinggi sejak awal, sehingga peluang untuk menunjukkan peningkatan lebih lanjut menjadi lebih kecil.

REFERENSI

- Chandrasekara, C., & Herath, P. (2021). Hands-on GitHub Actions: Implement CI/CD with GitHub Action Workflows for Your Applications. Apress. <https://doi.org/10.1007/978-1-4842-6464-5>
- Chen, X., Ji, Z., Fan, Y., & Zhan, Y. (2017). Restful API Architecture Based on Laravel Framework. Journal of Physics: Conference Series, 910, 012016. <https://doi.org/10.1088/1742-6596/910/1/012016>

- Delgado-Pérez, P., & Medina-Bulo, I. (2020). Customizable and scalable automated assessment of C/C++ programming assignments. *Computer Applications in Engineering Education*, 28(6), 1449–1466. <https://doi.org/10.1002/cae.22317>
- Gong, B. (2010). *Using Balanced Assessment Systems To Improve Student Learning and School Capacity: An Introduction*.
- González-Carrillo, C. D., Restrepo-Calle, F., Ramírez-Echeverry, J. J., & González, F. A. (2021). Automatic Grading Tool for Jupyter Notebooks in Artificial Intelligence Courses. *Sustainability*, 13(21), 12050. <https://doi.org/10.3390/su132112050>
- Grenander, M., Belfer, R., Kochmar, E., Serban, I. V., St-Hilaire, F., & Cheung, J. C. K. (2021). Deep Discourse Analysis for Generating Personalized Feedback in Intelligent Tutor Systems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17), 15534–15544. <https://doi.org/10.1609/aaai.v35i17.17829>
- Hagerer, G. J., Lahesoo, L., Anschutz, M., Krusche, S., & Groh, G. (2021). An Analysis of Programming Course Evaluations Before and After the Introduction of an Autograder. 2021 19th International Conference on Information Technology Based Higher Education and Training (ITHET), 01–09. <https://doi.org/10.1109/ITHET50392.2021.9759809>
- Haldeman, G., Tjang, A., Babeş-Vroman, M., Bartos, S., Shah, J., Yucht, D., & Nguyen, T. D. (2018). Providing Meaningful Feedback for Autograding of Programming Assignments. *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 278–283. <https://doi.org/10.1145/3159450.3159502>
- Hao, Q., Smith Iv, D. H., Ding, L., Ko, A., Ottaway, C., Wilson, J., Arakawa, K. H., Turcan, A., Pohlman, T., & Greer, T. (2022). Towards understanding the effective design of automated formative feedback for programming assignments. *Computer Science Education*, 32(1), 105–127. <https://doi.org/10.1080/08993408.2020.1860408>
- Harms, K. J., Rowlett, N., & Kelleher, C. (2015). Enabling independent learning of programming concepts through programming completion puzzles. 2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 271–279. <https://doi.org/10.1109/VLHCC.2015.7357226>
- Higgins, C. A., Gray, G., Symeonidis, P., & Tsintsifas, A. (2005). Automated assessment and experiences of teaching programming. *Journal on Educational Resources in Computing*, 5(3), 5. <https://doi.org/10.1145/1163405.1163410>
- Jailia, M., Kumar, A., Agarwal, M., & Sinha, I. (2016). Behavior of MVC (Model View Controller) based Web Application developed in PHP and .NET framework. 2016 International Conference on ICT in Business Industry & Government (ICTBIG), 1–5. <https://doi.org/10.1109/ICTBIG.2016.7892651>
- Jivani, P., Chopara, C., & Prashant, M. (2013). Over All Idea about MVC: How to use Model-View-Controller (MVC). *International Journal of Innovations in Engineering and Technology*.
- Kavita. (2025). Automated Grading and Feedback Systems for Programming in Higher Education Using Machine Learning. *Journal of Informatics Education and Research*, 5(1). <https://doi.org/10.52783/jier.v5i1.2142>
- Laaziri, M., Benmoussa, K., Khouliji, S., Mohamed Larbi, K., & Yamami, A. E. (2019). A comparative study of laravel and symfony PHP frameworks. *International Journal of Electrical and Computer Engineering (IJECE)*, 9(1), 704. <https://doi.org/10.11591/ijece.v9i1.pp704-712>
- Marion, S., Thompson, J., Evans, C., Martineau, J., & Dadey, N. (2019). THE CHALLENGES AND OPPORTUNITIES OF BALANCED SYSTEMS OF ASSESSMENT: A POLICY BRIEF.
- Purohit, K. (2020). Executing DevOps & CI/CD, Reduce in Manual Dependency. 5(9).
- Subecz, Z. (2021). Web-development with Laravel framework. *Gradus*, 8(1), 211–218. <https://doi.org/10.47833/2021.1.CSC.006>
- Sumintono, B. (2016). Aplikasi Pemodelan Rasch pada Asesmen Pendidikan: Implementasi Penilaian Formatif (assessment for learning).

