

Analisis Perbandingan Penggunaan Pengkodean Low-Code dan Konvensional Pada Pengembangan Sistem Informasi Kepegawaian (Studi Kasus : PT ABC)

Stieven Limanjaya

Universitas Multi Data Palembang

Email: stievenlimanjaya@mhs.mdp.ac.id

Muhammad Rachmadi

Universitas Multi Data Palembang

Email: rachmadi@mdp.ac.id

Alamat: Jl. Rajawali No. 14 Palembang 30113

Korespondensi penulis: stievenlimanjaya@mhs.mdp.ac.id*

Abstract. *The creation of a web-based application management system is essential to assist organisations in maintaining their operational efficiency in the rapidly growing digital era. The purpose of this research is to apply the waterfall development method approach to the development of a web-based application management system. We compared the time efficiency and features of a low-code platform such as OutSystems and traditional coding using native PHP. By streamlining procedures for recruitment, leave, warning or dismissal data, and human resource management, the system can improve overall operational effectiveness and efficiency. This research can provide in-depth information on the efficiency of development time and overall features. Our contribution is to clarify the performance comparison in the context of application management system development between low-code platforms and traditional development approaches..*

Keywords: *Waterfall, Outsystems, PHP Native, Human Resources Management, Low-Code Development.*

Abstrak. Pembuatan sistem manajemen aplikasi berbasis web sangat penting untuk membantu organisasi dalam menjaga efisiensi operasional mereka di era digital yang berkembang pesat. Tujuan dari penelitian ini adalah untuk menerapkan pendekatan metode pengembangan waterfall pada pengembangan sistem manajemen aplikasi berbasis web. Kami membandingkan efisiensi waktu dan fitur dari platform *low-code* seperti OutSystems dan pengkodean tradisional menggunakan PHP native. Dengan merampingkan prosedur perekrutan, cuti, data peringatan atau pemecatan, dan manajemen sumber daya manusia, sistem dapat meningkatkan efektivitas dan efisiensi operasional secara keseluruhan. Penelitian ini dapat memberikan informasi yang mendalam tentang efisiensi waktu pengembangan dan keseluruhan fitur. Kontribusi kami adalah untuk memperjelas perbandingan kinerja dalam konteks pengembangan sistem manajemen aplikasi antara platform *low-code* dan pendekatan pengembangan tradisional.

Kata kunci: Waterfall, Outsystems, PHP Native, Sistem Informasi Kepegawaian, Pengembangan Kode Rendah.

LATAR BELAKANG

Dalam era digital yang terus berkembang, tuntutan akan pengembangan aplikasi digital yang cepat dan efisien semakin meningkat. Dalam konteks ini, digunakan dua pendekatan, yaitu PHP native dan OutSystems. Kedua situs web dengan fitur yang sama dikembangkan menggunakan kedua pendekatan tersebut. Selain itu, satu database MySQL digunakan untuk menyimpan data. Penggunaan kedua pendekatan ini bertujuan untuk membandingkan efisiensi waktu pembuatan website.

Penggunaan platform pengembangan berbasis *low-code*, seperti OutSystems, telah membuka peluang baru dalam proses pembuatan sistem secara cepat dan efisien. Pendekatan *low-code* memungkinkan para pengembang untuk menciptakan sistem dengan mengurangi ketergantungan pada penulisan kode manual, bahkan dalam beberapa kasus, tanpa memerlukan penulisan kode sama sekali (Golovin, 2017) .

OutSystems adalah sebuah produk *Platform as a Service* (PaaS) yang berjalan di cloud, sehingga persyaratan sistem untuk perangkat keras komputer para pengembangnya rendah. Namun, dibutuhkan koneksi internet yang stabil dengan kecepatan tinggi dan latensi rendah secara konstan (Golovin, 2017). Platform *Low-code* ini dapat mempercepat dan mempermudah pengguna atau developer dalam proses pembuatan aplikasi tanpa harus menyetikkan kode program. Teknik yang digunakan oleh *low-code* juga memungkinkan pengguna untuk mengembangkan sebuah aplikasi dengan menggunakan fitur *drag-and-drop*. Sebuah teknik dimana programmer tanpa menulis kode terlebih dahulu untuk membuat sebuah tombol dan sebagainya (Martins et al., 2020). OutSystems juga memungkinkan integrasi yang stabil dengan basis data eksternal seperti MySQL, Oracle, MongoDB, dan masih banyak lagi.

Dengan adopsi teknologi, diharapkan perusahaan dapat meningkatkan efisiensi dalam manajemen sumber daya manusia, mempercepat proses penerimaan karyawan baru, mempermudah proses pengajuan izin dan cuti, serta menyimpan pendataan surat peringatan atau surat keterangan PHK. Dengan demikian, OutSystems tidak hanya menawarkan solusi untuk pengembangan yang cepat, tetapi juga mengurangi kesulitan yang sering kali terkait dengan pengembangan aplikasi secara tradisional (Waszkowski, 2019) . Ini menjadikan platform ini sebagai pilihan yang menarik bagi banyak organisasi yang ingin mengoptimalkan waktu dan sumber daya dalam pengembangan aplikasi digital mereka.

KAJIAN TEORITIS

1. Sistem Informasi

Sistem informasi adalah entitas terstruktur yang terdiri dari orang-orang, perangkat keras, perangkat lunak, jaringan komunikasi, dan sumber daya data. Tujuannya adalah mengumpulkan, mengolah, dan menyebarkan informasi di dalam suatu organisasi secara terorganisir dan terkoordinasi (Anggraeni, 2017).

2. Sistem Informasi Kepegawaian

Sistem Informasi Kepegawaian (SIMPEG) mengelola data pegawai dalam perusahaan atau instansi untuk mencapai target. Meliputi pencatatan data, perencanaan kepegawaian,

penggajian, penilaian kinerja, perpindahan pegawai, dan pelaporan data (Universitas Andalas, 2016).

3. PHP Native

PHP Native adalah bahasa pemrograman atau instruksi yang dibuat oleh satu orang pengembang tanpa menggunakan bantuan dari pihak lain. Kode program tersebut hanya diketahui oleh pengembang yang membuatnya. (Endra et al., 2021).

4. Low-Code Platform

Platform *low-code* adalah sumber daya untuk programmer dan non-programmer yang meminimalkan penggunaan bahasa pemrograman dan kompleksitas lingkungan pengembangan. Ini memungkinkan pembuatan dan implementasi aplikasi bisnis dengan cepat tanpa memerlukan pengaturan lingkungan yang rumit atau pelatihan yang intens. Pendekatan ini memungkinkan bisnis untuk mengembangkan perangkat lunak dengan biaya rendah dan cepat menyesuaikan diri dengan perubahan pasar dan permintaan konsumen (Waszkowski, 2019).

5. Outsystems

OutSystems adalah PaaS yang berbasis *cloud*, membutuhkan persyaratan sistem rendah namun koneksi internet stabil dengan kecepatan tinggi. Dengan pendekatan *low-code*, memungkinkan pengembang atau pengguna untuk membuat aplikasi tanpa mengetikkan banyak kode, menggunakan fitur *drag-and-drop* untuk pengembangan aplikasi. (Golovin, 2017; Martins et al., 2020)

6. PIECES

Kerangka PIECES adalah metode analisis sebagai dasar untuk memperoleh pokok-pokok permasalahan yang lebih spesifik. Dalam menganalisis sebuah sistem, biasanya akan dilakukan terhadap beberapa aspek antara lain analisis terhadap kinerja, informasi, ekonomi, pengendalian, efisiensi dan pelayanan. A.R. Dewi dalam (Muliensah dan Budihartanti 2020).

7. Use Case Diagram

Use Case adalah deskripsi sistem yang berfokus pada bagaimana sistem akan digunakan oleh aktor-aktor tertentu untuk mencapai tujuan tertentu (Whitten & Bentley, 2007).

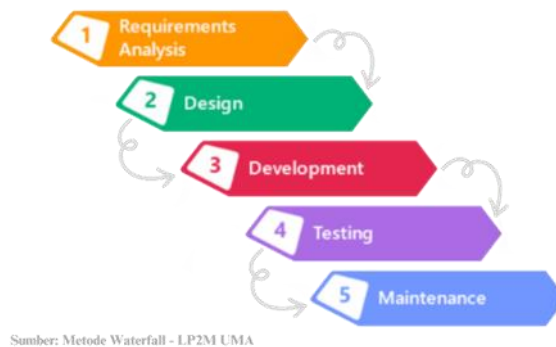
8. Class Diagram

Class Diagram adalah representasi visual dari struktur statis dari sistem yang menunjukkan kelas-kelas dalam sistem, atribut-atribut yang dimiliki oleh setiap kelas, dan hubungan antara kelas-kelas tersebut (Whitten & Bentley, 2007).

METODE PENELITIAN

Penelitian ini menerapkan metode waterfall. Karakteristik dari konsep Waterfall adalah pendekatan berbasis dokumen (Yurindra. 2017). Waterfall yang merupakan model yang memiliki struktur mulai dari perancangan analisis, desain, pengembangan, pengujian dan pemeliharaan, sehingga tahap pengembangan Waterfall memiliki struktur model pengembangan yang disebut linier dan berurutan (Nino, Azahari, & Kridalaksana, 2022). Tahap pertama adalah analisis kebutuhan, yang melibatkan identifikasi dan definisi kebutuhan sistem secara mendalam. Selanjutnya, pada tahap perancangan, sistem direncanakan secara detail termasuk struktur data, arsitektur perangkat lunak, antarmuka, dan prosedur pengkodean. Tahap implementasi melibatkan pembangunan kode sistem berdasarkan desain yang telah direncanakan sebelumnya. Dalam penelitian ini, pendekatan pengembangan menggunakan 2 metode: pengkodean konvensional (PHP Native) dan platform low-code (Outsystems). Kemudian, tahap pengujian dilakukan untuk pemeriksaan fungsi sistem dan identifikasi kesalahan dari segi logika dan fungsional. Terakhir, tahap pemeliharaan melibatkan pemantauan dan perbaikan setelah peluncuran sistem. Setiap tahap memungkinkan penyelesaian sebelum melanjutkan ke tahap berikutnya, menciptakan struktur yang jelas dalam pengembangan sistem informasi. (Rosa A.S., 2018)

HASIL DAN PEMBAHASAN



Gambar 1 . Tahapan Pengembangan Metode Waterfall

Analisis Kebutuhan (Requirement Analysis)

Pada tahap ini, dilakukan pengujian langsung dan wawancara kepada perusahaan untuk mengumpulkan data. Proses pengujian melibatkan pembelajaran fitur dan skema *low-code* selama 30 hari. Wawancara dilakukan untuk memahami sistem kepegawaian yang berlaku di perusahaan sebagai pertimbangan dalam pembuatan fitur. Tujuan utama dari tahap ini adalah untuk memahami kebutuhan utama dalam pembuatan website.

Desain (Design)

Desain penelitian ini melibatkan dua metode pembuatan website, PHP Native (konvensional) dan Outsystems (*low-code*). Tujuan utama dari tahap desain ini adalah untuk membandingkan efisiensi waktu antara kedua metode tersebut. Selama tahap desain, beberapa analisis dan diagram telah dirancang.

Analisis Permasalahan

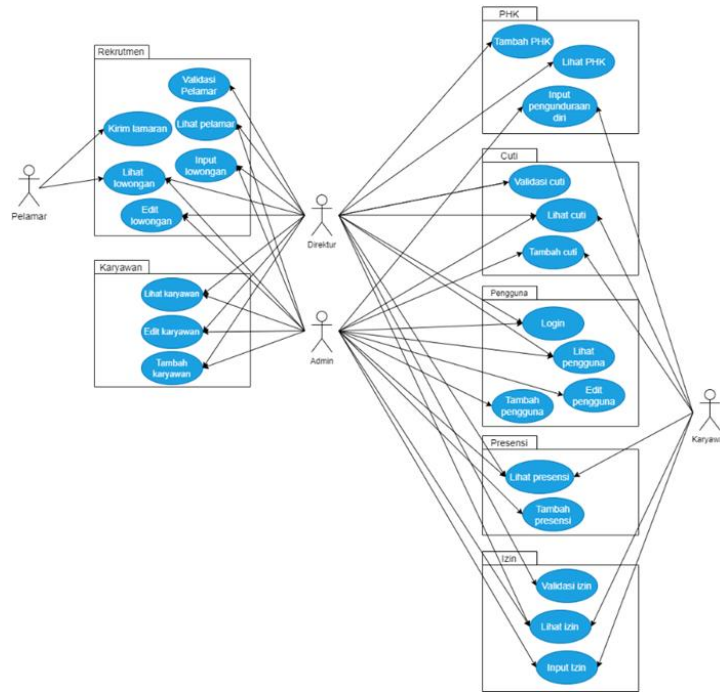
Dalam analisis permasalahan, kami menggunakan metode PIECES untuk mengidentifikasi dan memahami permasalahan yang ada. Kerangka PIECES dapat dilihat pada Tabel 1.

Tabel 1. Kerangka PIECES

No.	PIECES	Permasalahan
1.	<i>Performance</i>	- Lambatnya kinerja dalam proses perekrutan.
2.	<i>Information</i>	- Kesulitan mengetahui informasi data karyawan dan data pelamar. - Karyawan tidak dapat melihat data presensi.
3.	<i>Economy</i>	- Pengelolaan dalam perhitungan presensi, dan cuti rentan terhadap kesalahan.
4.	<i>Control</i>	- Data karyawan dan data pelamar dapat saja hilang dan tercecer.
5.	<i>Efficiency</i>	- Proses pengajuan cuti dan rekrutmen pegawai yang tidak efisien.
6.	<i>Service</i>	- Terhambatnya proses layanan dalam pengajuan cuti dan PHK.

Analisis Kebutuhan

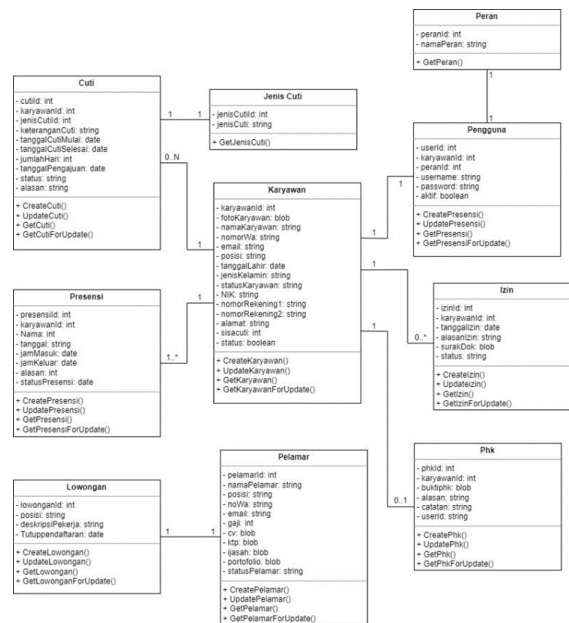
Sebagai bagian dari analisis kebutuhan, kami membuat diagram use-case. Diagram use-case merupakan alat bantu yang berharga yang membantu menjawab tantangan dalam menentukan fungsi yang harus dilakukan oleh sebuah sistem dengan melihat dari sudut pandang pengguna dan pemangku kepentingan. (Whitten dan Bentley, 2007). Diagram use-case dapat dilihat pada Gambar 2.



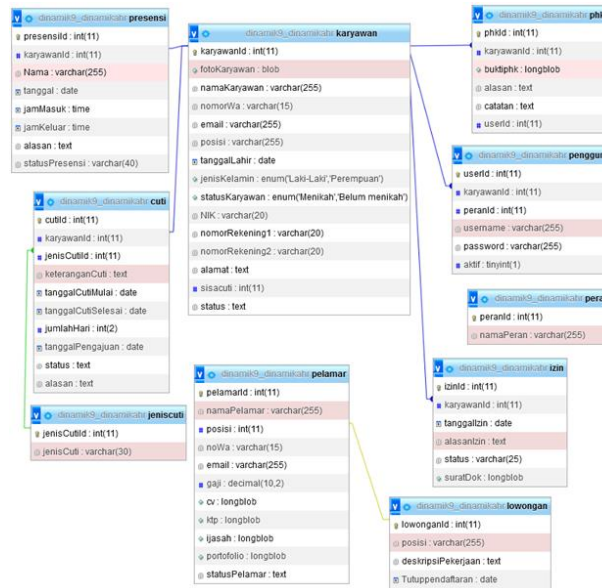
Gambar 2. Diagram use-case

Implementasi (Implementation)

Implementasi penelitian ini dilakukan dengan membangun website menggunakan kedua metode tersebut. Proses ini dilakukan selama kurang lebih 1 bulan setengah. Selama tahap implementasi, penekanan diberikan pada penerapan desain dan analisis yang telah dibuat pada tahap sebelumnya. Dalam tahap implementasi ini juga dirancang dengan menggunakan beberapa diagram seperti class diagram dalam Gambar 3 dan Relasi Antar Tabel pada Gambar 4.



Gambar 3. Class Diagram



Gambar 4. Relasi Antar Tabel

Pengujian (Testing)

Pengujian dilakukan untuk memverifikasi hipotesis bahwa pembuatan website menggunakan *low-code* lebih cepat dan mudah. Hasil pengujian menunjukkan bahwa pembuatan website menggunakan *low-code* dua kali lebih cepat dibandingkan dengan metode konvensional. Namun, ditemukan juga bahwa *low-code* memiliki keterbatasan dalam membuat *logic* yang rumit. Hal tersebut dapat dilihat di perbandingan waktu pada tabel 3.

Tabel 3. Perbandingan waktu

Total Keseluruhan			
Outsystems		PHP Native	
Frontend	02:17:28	Frontend	05:43:09
Backend	02:26:10	Backend	04:49:41
Debug	00:34:03	Debug	00:08:10
Total	05:17:41	Total	10:42:00

Efisiensi dan Fleksibilitas

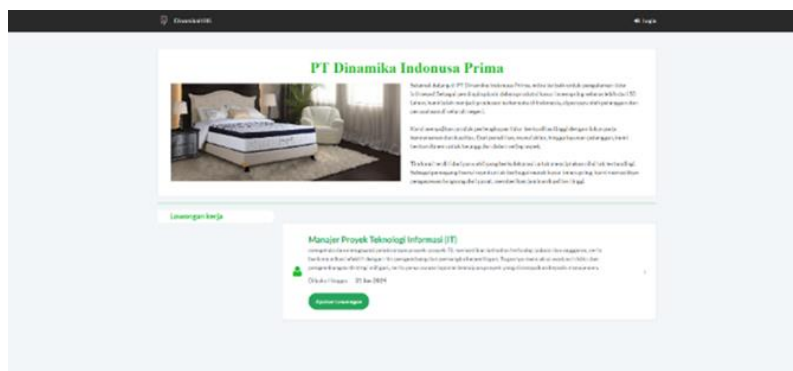
Hasil perbandingan platform Outsystems dan PHP Native dari sisi efisiensi waktu dan fleksibilitas fitur dapat dilihat pada tabel 4.

Tabel 4. Efisiensi dan Fleksibilitas

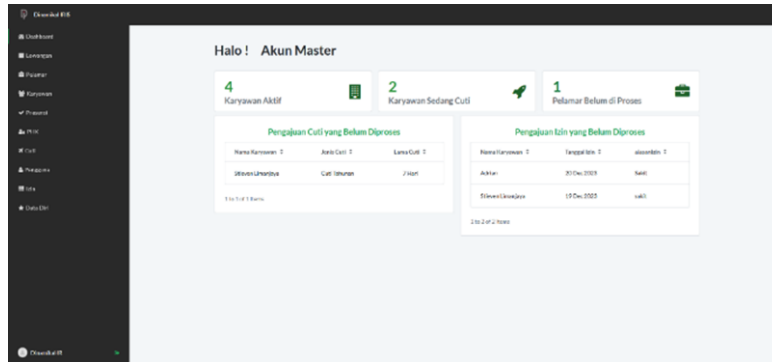
EFISIENSI DAN FLEKSIBILITAS	
PHP NATIVE	OUTSYSTEMS
Kemampuan kontrol kode (kustomisasi kode)	Keterbatasan kontrol kode (kustomisasi kode)
Kendali <i>source code</i> Penuh	Tidak dapat mengambil <i>source code</i>
Kustomisasi ekspansif	Kustomisasi terbatas
Antarmuka pengguna <i>Custom</i>	Antarmuka pengguna <i>drag-and-drop</i>
Fungsi aritmatika yang lebih luas	Fungsi aritmatika yang terbatas
Kustomisasi antarmuka pengguna yang lebih luas	Kustomisasi antarmuka pengguna terbatas
Proses <i>debugging</i> membutuhkan waktu yang singkat	Proses <i>debugging</i> membutuhkan waktu yang cukup lama
Memiliki komunitas pengembang yang besar dan aktif dengan banyak sumber daya, <i>tutorial</i> , dan dukungan.	Terdapat komunitas dan dukungan, tetapi tidak sebesar komunitas PHP Native
Memerlukan perangkat manajemen proyek tambahan dan pemantauan yang lebih mandiri.	Memiliki alat bawaan untuk manajemen proyek dan pemantauan yang memudahkan pengelolaan proyek.

Perancangan Sistem

Dalam tahap perancangan sistem ini, dibuat dua situs web dengan tujuan yang sama: menyediakan pengalaman pengguna yang konsisten dan intuitif. Satu situs menggunakan platform *low-code* (Outsystems), sementara yang lain menggunakan pendekatan konvensional (PHP Native). Kedua situs memiliki desain antarmuka pengguna (UI) yang serupa dalam tata letak, navigasi, dan interaksi pengguna. Pemilihan platform didasarkan pada kebutuhan proyek dan keunggulan masing-masing. Berikut ini merupakan beberapa tampilan sistem yang telah dibangun.



Gambar 5. Tampilan Halaman Rekrutmen



Gambar 6. Tampilan Halaman Dashboard

KESIMPULAN DAN SARAN

Dengan menggunakan pendekatan *low-code*, pengembangan aplikasi menjadi lebih cepat dan efisien. Platform *low-code*, seperti OutSystems, menawarkan kelebihan dalam hal kecepatan, kemudahan integrasi, dan kemampuan untuk inovasi yang lebih inklusif. Selain itu, penggunaan *low-code* juga dapat mengurangi biaya pengembangan, mempercepat proses, dan meningkatkan kelancaran sistem. Di sisi lain, pendekatan konvensional memberikan kontrol penuh dan kustomisasi tak terbatas, namun memerlukan waktu pengembangan yang lebih lama.

Dengan demikian, penggunaan pendekatan *low-code*, seperti OutSystems, sangat disarankan untuk pengembangan yang membutuhkan kecepatan dan inovasi yang inklusif, sementara pendekatan konvensional lebih cocok untuk pengembangan yang memerlukan kontrol penuh dan kustomisasi tak terbatas. Kedua pendekatan ini dapat diintegrasikan sesuai dengan kebutuhan pengembangan aplikasi yang kompleks..

DAFTAR REFERENSI

- Anggraeni, E. Y. (2017). *Pengantar sistem informasi*. Penerbit Andi.
- Arnomo, S. A., Yulia, Y., & Ukas, U. (2023). Building The Prediction of Sales Evaluation on Exponential Smoothing using The OutSystems Platform. *ILKOM Jurnal Ilmiah*, 15(2), 222-228.
- Bentley, L. D., & Whitten, J. L. (2007). *Systems analysis and design for the global enterprise* (Vol. 417). New York: McGraw-Hill/Irwin.
- Endra, R. Y., Aprilinda, Y., Dharmawan, Y. Y., & Ramadhan, W. (2021). Analisis Perbandingan Bahasa Pemrograman PHP Laravel dengan PHP Native pada Pengembangan Website. *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, 11(1), 48. <https://doi.org/10.36448/expert.v11i1.2012>.
- Golovin, D. (2017). Outsystems as a rapid application development platform for mobile and web applications.

- Gottschalk, S.F., Bhat, R., Weidmann, N., Kirchhoff, J., & Engels, G. (2022). Low-code experimentation on software products. Proceedings of the 25th International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings.
- Martin, J. (1990). *Information engineering, planning & analysis: book 2*. Prentice-Hall, Inc.
- Muliansah, Rendi, and Cahyani Budihartanti. 2020. "Analisa Pemanfaatan E-Puskesmas Di Loket Pendaftaran Pada Puskesmas Kecamatan Pademangan Dengan Metode PIECES." JCSE 1. <https://doi.org/10.36596/jcse.v1i1.22>
- Nino, F., Azahari, A., & Kridalaksana, A. H. Nodemcu Based Prototype Tool for Noise Detection. *Tepian*, 2(2), 43-48.
- R. Martins, F. Caldeira, F. Sa, M. Abbasi, and P. Martins, "An overview on how to develop a low-code application using OutSystems," Proc. Int. Conf. Smart Technol. Comput. Electr. Electron. ICSTCEE 2020, pp. 395–401, 2020.
- Rosa A.S., M. S. (2018). *Rekasaya Perangkat Lunak Terstruktur dan Berorientasi Objek*.
- Universitas Andalas. (2023). Sistem Informasi Manajemen Kepegawaian (SIMPEG). <https://lptik.unand.ac.id/berita-dan-panduan/item/149-sistem-informasi-manajemen-kepegawaian-simpeg-universitas-andalas>
- Waszkowski, R. (2019). Low-code platform for automating business processes in manufacturing. *IFAC-PapersOnLine*, 52(10), 376–381. <https://doi.org/10.1016/j.ifacol.2019.10.060>
- Yurindra. (2017). *Software Engineering*. Yogyakarta: Deepublish.