

Implementation of Low-Code Development Platform (LCDP) Appian Designer in the Development of a Waste Bank Application

Yustia Hapsari^{1*}, Marsella Lusio Netanya S², Agnes Dwita Susilawati³, Nur Tulus Ujjianto⁴, Sumarno⁵

^{1,2,3} Bisnis Digital, Fakultas Ekonomi dan Bisnis, Universitas Pancasakti Tegal

⁴ Sistem Informasi, Fakultas Ekonomi dan Bisnis, Universitas Pancasakti Tegal

⁵ Manajemen, Fakultas Ekonomi dan Bisnis, Universitas Pancasakti Tegal

Informasi Artikel

Diterima : 11 Februari 2025
Revisi : 17 Mei 2025
Publikasi : 20 Juni 2025

Kata Kunci:

Platform Low-Code
Appian Designer
Bank Sampah
Rapid Application Development
Interface Performance View

ABSTRAK

Permasalahan pengelolaan sampah, khususnya pada aspek pencatatan dan manajemen transaksi di Bank Sampah, masih menjadi kendala di masyarakat. Penelitian ini bertujuan mengembangkan aplikasi Bank Sampah berbasis digital menggunakan platform Low-Code Development Platform (LCDP) Appian Designer. Metode RAD (Rapid Application Development) digunakan karena yang mendukung iterasi cepat dan pengembangan antarmuka fleksibel. Tahapan penelitian mencakup perencanaan kebutuhan, desain antarmuka, konstruksi sistem, dan pengujian performa. Data kebutuhan dirancang dalam bentuk database relasional dan diimplementasikan ke dalam Appian. Pengujian performa menggunakan tools bawaan Appian menunjukkan bahwa waktu render untuk form login, setoran sampah, dan penarikan saldo hanya 3 milidetik, registrasi 14 milidetik, serta laporan dan menu warga 4 milidetik. Hasil ini menunjukkan bahwa seluruh form memiliki performa yang responsif dan stabil. Platform Appian terbukti mampu menghasilkan aplikasi dengan antarmuka ringan dan cepat. Penggunaan RAD melalui fitur Interface Performance View mempercepat proses pengembangan sekaligus menjaga kualitas sistem. Kesimpulannya, pendekatan ini efektif dalam mendukung digitalisasi pengelolaan Bank Sampah secara lebih sistematis, efisien, dan berkelanjutan.

ABSTRACT

Waste management issues, particularly in the areas of record-keeping and transaction management within Waste Banks, remain a challenge in communities. This study aims to develop a digital-based Waste Bank application using the Low-Code Development Platform (LCDP) Appian Designer. The method applied is Rapid Application Development (RAD), which supports rapid iteration and flexible interface development. The research stages include requirements planning, interface design, system construction, and performance testing. The feature requirements were designed in a relational database structure and implemented in Appian. Performance testing using Appian's built-in tools showed that the render times for the login, waste deposit, and withdrawal forms were only 3 milliseconds, registration 14 milliseconds, and report and resident menu forms 4 milliseconds. These results indicate that all forms exhibit responsive and stable performance. Appian has proven capable of delivering applications with lightweight and fast user interfaces. The use of RAD through the Interface Performance View feature accelerates the development process while maintaining system quality. In conclusion, this approach is effective in supporting the digitalization of Waste Bank management in a more systematic, efficient, and sustainable manner.

This is an open-access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license



***Penulis Koresponden**Email: yustia.hapsari@upstegal.ac.id

Cara sitasi IEEE::

Y. Hapsari, M. L. Netanya, A. D. Susilawati, N. T. Ujionto & Sumarno, "Implementation of Low-Code Development Platform (LCDP) Appian Designer in the Development of a Waste Bank Application" *Journal of Artificial Intelligence and Software Engineering (J-AISE)*, vol. 5, no. 2, pp. 421-431, Juni 2025. doi: 10.30811/jaise.v5i2.6682

1. PENDAHULUAN

Perkembangan teknologi perangkat lunak dalam beberapa tahun terakhir menunjukkan perubahan yang substansial, ditandai dengan kemunculan berbagai inovasi. Salah satu yang menonjol adalah Platform Pengembangan Low-Code (LCDP), yang mulai banyak diadopsi karena efisiensi dan kemudahannya dalam membangun aplikasi dan minim penulisan kode secara manual [1]. LCDP menawarkan pendekatan berbasis visual dalam pembuatan perangkat lunak, sehingga mempercepat proses pengembangan serta mengurangi kompleksitas yang biasanya terjadi pada metode tradisional [2]. Platform ini semakin relevan seiring dengan meningkatnya kebutuhan solusi perangkat lunak di berbagai sektor, termasuk pengelolaan lingkungan dan keberlanjutan seperti bank sampah.

Bank sampah merupakan inisiatif yang bertujuan untuk mengelola limbah dengan prinsip ekonomi sirkular, di mana sampah dikonversi menjadi nilai ekonomi bagi masyarakat [3]. Bank Sampah memiliki peran strategis sebagai wahana edukatif bagi masyarakat dalam mengimplementasikan prinsip-prinsip Reduce, Reuse, dan Recycle guna mendorong praktik pengelolaan sampah yang berorientasi pada keberlanjutan lingkungan [4]. Namun, implementasi bank sampah sering kali menghadapi tantangan administratif, seperti pencatatan transaksi, pemantauan sampah yang dikumpulkan, serta distribusi insentif kepada anggota. Dalam konteks ini, LCDP dapat berperan sebagai solusi yang efisien dalam membangun sistem pengelolaan bank sampah berbasis digital tanpa memerlukan keterampilan pemrograman yang mendalam [5].

Keunggulan utama LCDP dalam pengembangan aplikasi adalah kemampuannya untuk mengurangi waktu dan biaya pengembangan perangkat lunak [6]. Low-Code Development merupakan sebuah pendekatan dalam pengembangan perangkat lunak yang memungkinkan berbagai kalangan, termasuk individu tanpa latar belakang teknis atau keahlian pemrograman yang mendalam, seperti pelaku bisnis dan analis sistem, untuk ikut serta dalam proses perancangan dan pembuatan aplikasi. Dengan antarmuka visual yang intuitif serta fitur drag-and-drop, platform ini mempermudah pengguna non-profesional dalam menyusun alur kerja dan mengintegrasikan komponen aplikasi tanpa harus menulis kode secara manual dalam jumlah besar [7]. Studi terbaru juga menunjukkan bahwa penggunaan LCDP meningkatkan efisiensi dalam pengembangan sistem berbasis web dan mobile, terutama dalam bidang manajemen bisnis dan otomatisasi proses kerja [8].

Meskipun LCDP menawarkan berbagai keuntungan, tantangan dalam penggunaannya tetap ada. Beberapa penelitian mencatat bahwa fleksibilitas dan skalabilitas LCDP masih terbatas dibandingkan dengan pengembangan perangkat lunak konvensional [9]. Selain itu, karena ketergantungan terhadap vendor penyedia platform, ada potensi risiko dalam jangka panjang terkait dengan kompatibilitas dan dukungan teknologi [10]. Diperlukan analisis yang komprehensif terhadap efektivitas penggunaan Low-Code Development Platform (LCDP) dalam pengembangan aplikasi spesifik, seperti sistem manajemen bank sampah, guna memaksimalkan potensi keunggulannya serta mengidentifikasi dan mengantisipasi berbagai keterbatasan yang mungkin muncul.

Jurnal-jurnal yang mengangkat konteks bank sampah menunjukkan novelty dalam penerapan teknologi digital pada praktik pengelolaan sampah berbasis komunitas. Kontribusi utamanya terletak pada bagaimana sistem informasi dirancang untuk menjawab permasalahan administratif seperti pencatatan, pemantauan, dan distribusi insentif dalam bank sampah. Pendekatan ini menekankan pentingnya efisiensi operasional dan pemberdayaan masyarakat melalui teknologi, serta integrasi prinsip ekonomi sirkular dalam solusi digital namun dalam skalabilitas pendekatan pengujian performa.

Sementara itu, jurnal yang berfokus pada penggunaan Appian secara spesifik menawarkan nilai kebaruan dari sisi teknologi dengan memanfaatkan platform low-code untuk mempercepat pengembangan sistem informasi. Appian sebagai platform LCDP yang dipilih menonjol karena kemampuannya dalam otomatisasi proses bisnis dan kemudahan penggunaan, sehingga memungkinkan kolaborasi antara pengguna

teknis dan non-teknis. Di sisi lain, beberapa studi menunjukkan novelty melalui pendekatan pengujian performa, yakni dengan menilai efektivitas sistem yang dibangun dari segi kecepatan, kestabilan, dan skalabilitas. Pendekatan ini menambah kedalaman analisis dalam konteks implementasi sistem, memastikan bahwa solusi yang dikembangkan tidak hanya efisien dalam pembuatannya, tetapi juga andal dalam penggunaannya.

Penelitian ini bertujuan untuk mengevaluasi efektivitas penggunaan Low-Code Development Platform dalam pengembangan aplikasi bank sampah. Evaluasi akan dilakukan berdasarkan parameter kecepatan pengembangan, kemudahan penggunaan, serta kemampuan integrasi sistem yang relevan dengan kebutuhan pengelolaan bank sampah. Hasil dari desain penelitian ini diharapkan mampu memberikan wawasan khususnya bagi pengelola bank sampah dan pengembang perangkat lunak dalam memilih solusi teknologi yang optimal guna meningkatkan efisiensi dan keberlanjutan program bank sampah di berbagai daerah.

2. METODE

Metode yang digunakan adalah model RAD (Rapid Application Development) yang sesuai dengan karakteristik Low-Code Development Platform (LCDP), yaitu kecepatan dalam iterasi dan kemudahan modifikasi antarmuka [11][12]. Platform low-code yang digunakan akan dijelaskan secara teknis, mencakup fitur-fitur pendukung, struktur data, serta hasil prototipe yang dihasilkan. Pengolahan data dalam penelitian ini dilakukan melalui pendekatan deskriptif kualitatif, dengan menerapkan proses reduksi, penyajian, dan penarikan kesimpulan atas data yang diperoleh dari hasil observasi serta dokumentasi secara sistematis.

Pengumpulan data dilakukan melalui studi pustaka, simulasi pengembangan sistem dalam konteks sistem pengelolaan bank sampah, dan evaluasi berdasarkan indikator: kecepatan pengembangan, kebutuhan keahlian teknis, kemudahan modifikasi, dan dukungan kolaborasi lintas peran. Melalui pendekatan ini, peneliti dapat mengidentifikasi keunggulan dan keterbatasan dari masing-masing metode serta memberikan rekomendasi kontekstual sesuai kebutuhan dan sumber daya organisasi. Hasil evaluasi dirangkum dalam tabel perbandingan berikut:

Tabel 1. Perbandingan LCDP vs Waterfall (Manual Coding)

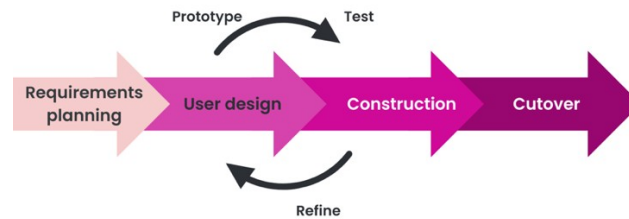
No	Aspek	LCDP	Waterfall (Manual Coding)
1	Kecepatan Pengembangan	Cepat, karena berbasis visual dan drag-drop	Lambat, karena memerlukan penulisan kode penuh
2	Keterlibatan Non-Teknis	Tinggi, pengguna awam dapat ikut terlibat	Rendah, hanya tim teknis yang bisa berkontribusi
3	Fleksibilitas & Perubahan	Tinggi, mudah diubah kapan saja	Rendah, sulit mengubah setelah fase selesai
4	Kebutuhan Keahlian Teknis	Minimal, cukup pelatihan dasar penggunaan	Tinggi, butuh pemahaman bahasa pemrograman
5	Dokumentasi & Proses	Ringkas, cenderung ringan	Mendalam dan berlapis sesuai tahapan
6	Kesesuaian untuk Prototipe	Sangat cocok, iteratif dan cepat dicoba	Kurang cocok, proses linier
7	Skalabilitas Sistem	Terbatas pada platform yang digunakan	Lebih fleksibel dan scalable secara teknis

Fokus analisis diarahkan pada efektivitas proses pengembangan, efisiensi waktu, kemudahan penggunaan, dan keterlibatan pengguna dalam proses iterasi desain aplikasi [13]. Temuan dari penelitian ini diharapkan mampu menyajikan perspektif implementatif terkait pemanfaatan platform Low-Code Development Platform (LCDP) sebagai alternatif solusi yang praktis dan efisien dalam mempercepat proses digitalisasi sistem manajemen Bank Sampah.

Dalam pelaksanaan penelitian ini, proses pengembangan sistem informasi dilakukan menggunakan pendekatan Low-Code Development Platform (LCDP) dengan total waktu pengembangan selama tiga minggu. Minggu pertama difokuskan pada perancangan alur kerja dan struktur database secara visual melalui platform, minggu kedua pada pembuatan antarmuka pengguna dan logika bisnis, sedangkan minggu ketiga digunakan untuk pengujian sistem serta perbaikan berdasarkan masukan pengguna. Pendekatan ini menunjukkan efisiensi signifikan dibandingkan metode pengembangan tradisional yang umumnya memerlukan waktu dua hingga tiga kali lebih lama untuk lingkup proyek serupa.

Untuk pengujian sistem yang terlibat sebanyak 3 orang pengguna dilibatkan, yang terdiri dari pengelola bank sampah, anggota masyarakat pengguna layanan, serta staf administrasi internal. Pemilihan responden ini bertujuan untuk mengukur sejauh mana sistem yang dikembangkan dapat digunakan oleh pihak

non-teknis, serta menilai aspek usability, kecepatan akses, dan ketepatan fungsi. Umpan balik dari pengujian digunakan sebagai dasar penyempurnaan sistem dan menjadi indikator awal keberhasilan pendekatan LCDP dalam pengembangan aplikasi berbasis komunitas.



Gambar 1. Fase Rapid Application Development (RAD)

Metode RAD dipilih karena memiliki fleksibilitas tinggi dan memungkinkan proses pengembangan sistem dilakukan dalam waktu yang relatif singkat dan efisien [14]. Untuk Low-Code Development Platform pada penelitian ini menggunakan Appian. Appian merupakan salah satu platform low-code terkemuka yang mendukung pengembangan aplikasi web dan mobile secara cepat dan efisien. Platform ini menyediakan berbagai fitur seperti penyesuaian antarmuka, kolaborasi tim, manajemen tugas, dan intranet sosial. Keunggulan lainnya adalah adanya mesin pengambilan keputusan yang mampu memodelkan logika bisnis kompleks, sehingga mempermudah otomatisasi proses dan pengambilan keputusan dalam sistem [15].

3. HASIL DAN PEMBAHASAN

3.1. Tahap Requirements Planning

Tahapan utama penelitian ini adalah Requirements Planning yang merupakan fase awal dalam metode RAD yang digunakan untuk mengidentifikasi kebutuhan sistem secara menyeluruh sebelum proses desain dan pengembangan dilakukan. Kegiatan ini bertujuan untuk merumuskan fitur-fitur penting yang harus tersedia dalam aplikasi, serta alur bisnis yang akan didukung oleh sistem digital yang dikembangkan menggunakan platform LCDP Appian Designer.

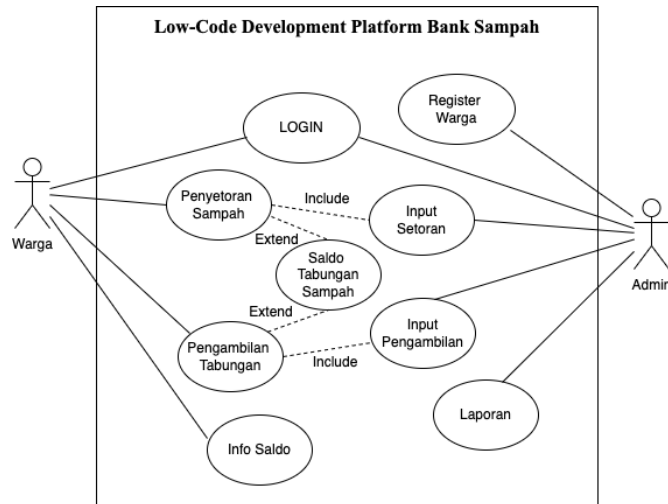
Dari hasil identifikasi, diperoleh bahwa pengelolaan data transaksi setoran sampah, data tabungan dari setoran dan pengambilan tabungan sampah, informasi saldo dan penyusunan laporan oleh admin merupakan proses utama yang perlu didigitalisasi. Dalam platform Appian Designer, seluruh kebutuhan ini dapat dirancang dalam bentuk form, workflow, dan interface interaktif yang dapat dikustomisasi dengan cepat. Hal ini sangat sesuai dengan karakteristik RAD yang mengutamakan iterasi cepat dan responsif terhadap perubahan kebutuhan. Berikut adalah ringkasan kebutuhan sistem berdasarkan hasil analisis awal yang ditunjukkan Tabel 1 berikut.

Tabel 2. Tahap Requirements Planning

No	Fitur	Deskripsi
1	Registrasi dan login warga	Warga didaftarkan, lalu diberikan hak akses oleh admin
2	Pencatatan transaksi setoran sampah	Data setoran sampah diinput admin, yang akan otomatis menjadi tabungan warga
3	Pencatatan transaksi pengambilan saldo bank sampah oleh warga	Admin akan menginput data pengambilan tabungan warga yang akan otomatis mengurangi jumlah tabungan
4	Laporan keuangan	Admin membuat laporan berkala
5	Informasi saldo tabungan warga	Pembuatan informasi saldo otomatis yang dapat diakses oleh warga

3.2. Tahap User Desain

Dalam tahap awal pengembangan aplikasi Bank Sampah berbasis Low-Code Development Platform (LCDP), dilakukan analisis awal kebutuhan sistem yang menunjukkan bahwa aplikasi yang dikembangkan harus mampu menangani aktivitas utama operasional, seperti pencatatan transaksi setoran sampah, saldo tabungan sampah, pengambilan tabungan sampah, serta pelaporan secara berkala dan info saldo yang dapat diakses oleh warga. Perancangan Use Case dapat dilihat pada Gambar 2 berikut.



Gambar 2. Use Case Diagram Sistem Bank Sampah

Pengguna dilibatkan secara langsung untuk merancang dan meninjau prototype antarmuka aplikasi. Platform Appian memungkinkan pembuatan form, dashboard, dan workflow secara visual tanpa harus melakukan banyak penulisan kode (coding). Hasil dari tahap ini adalah mockup awal dari aplikasi yang secara modul fungsional menggambarkan proses kerja Bank Sampah secara digital.

Tabel 3. Modul Prototype Antarmuka Appian

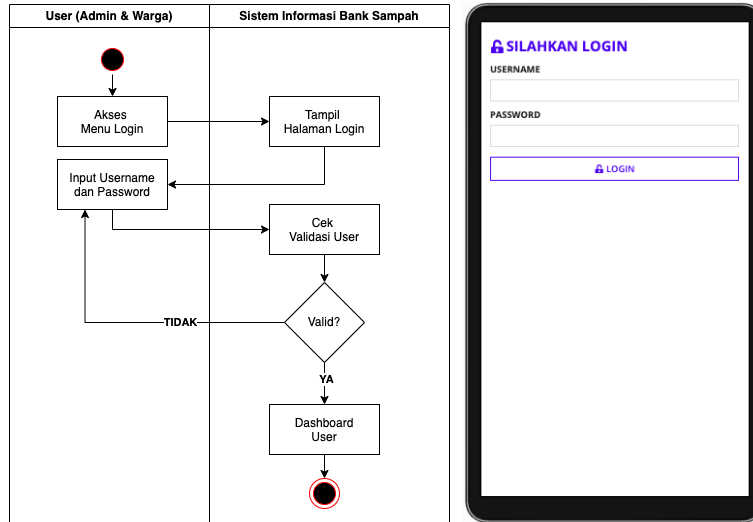
No	Komponen Modul	Pengguna	Fungsi
1	Modul Login	Admin/Warga	Untuk melakukan login kedalam sistem
2	Modul Registrasi	Admin	Untuk mendaftarkan akun user baru di sistem bank sampah
3	Dashboard Warga	Warga	Menampilkan informasi saldo, informasi setoran sampah dan transaksi pengambilan tabungan (histori)
4	Modul Transaksi	Admin	Input user baru, jenis sampah, transaksi setoran sampah, transaksi pengambilan tabungan
5	Modul Laporan	Admin	Pembuatan laporan berkala

Dengan menyusun modul-modul yang akan dibuat, akan memudahkan platform LCDP mendesain pemodelan fitur-fitur tersebut secara lebih cepat dan fleksibel, terutama dalam proses pembuatan prototipe antarmuka untuk pengguna yaitu admin dan warga.

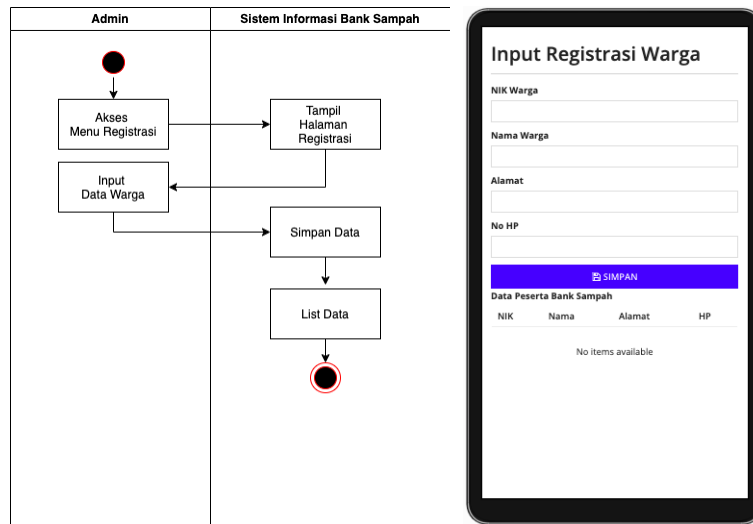
Pengguna dapat langsung melihat hasil perancangan dan memberikan masukan secara iteratif. Hasil pengujian awal terhadap prototipe menunjukkan bahwa fitur-fitur utama dapat direalisasikan dengan baik dalam waktu pengembangan yang relatif singkat. Hal ini menunjukkan efektivitas LCDP dalam mendukung proses pengembangan aplikasi yang responsif terhadap kebutuhan pengguna, sekaligus efisien dalam hal waktu dan sumber daya.

3.3. Rapid Construction

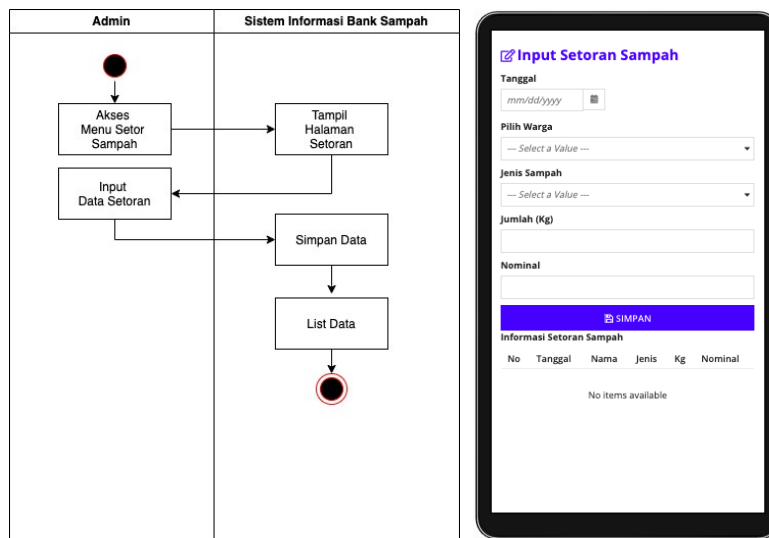
Setelah disetujui oleh pengguna, prototype dikembangkan lebih lanjut menjadi aplikasi fungsional menggunakan platform LCDP Appian. Proses ini dilakukan secara iteratif dengan melibatkan pengguna langsung serta dapat menguji versi awal aplikasi, memberi umpan balik, dan melakukan penyesuaian dengan cepat. Keunggulan LCDP Appian Designer dalam hal otomatisasi proses bisnis, antarmuka seret-lepas (drag and drop), serta koneksi ke database internal mempercepat siklus pembangunan aplikasi.



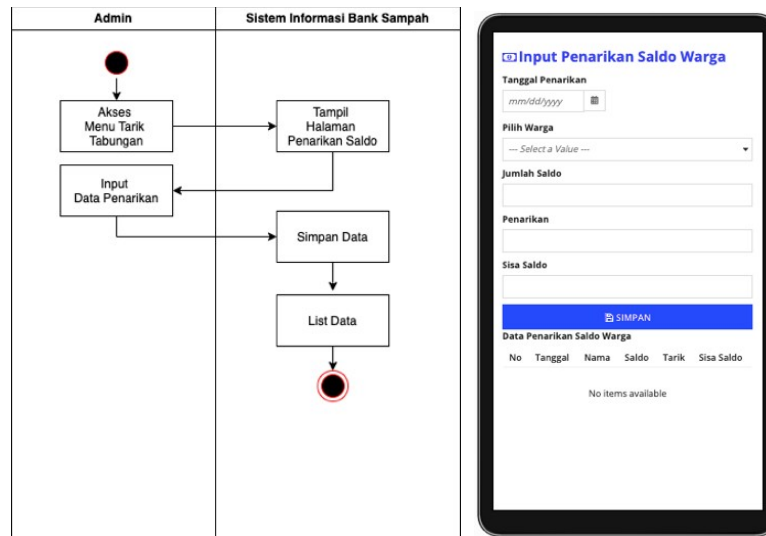
Gambar 3. Aktifitas Login dan Desain Antarmuka



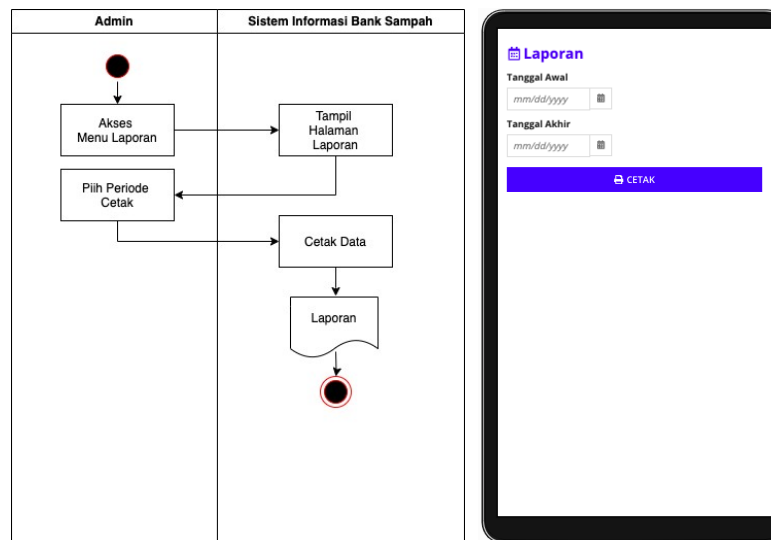
Gambar 4. Aktifitas Registrasi dan Desain Antarmuka



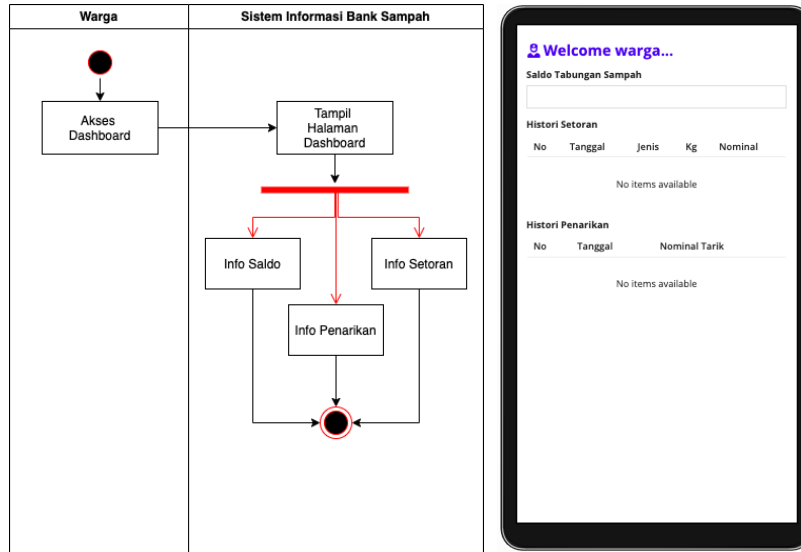
Gambar 5. Aktifitas Input Setoran Sampah dan Desain Antarmuka



Gambar 6. Aktifitas Input Penarikan Saldo dan Desain Antarmuka

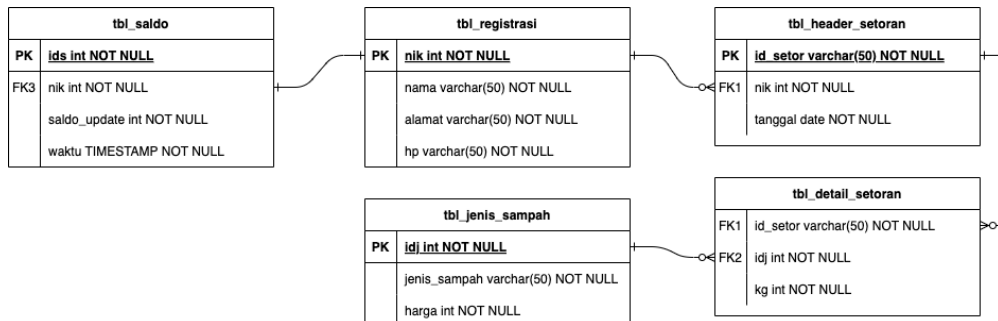


Gambar 7. Aktifitas Laporan dan Desain Antarmuka



Gambar 8. Aktifitas Dashboard Warga dan Desain Antarmuka

Langkah awal dalam konstruksi database pada pengembangan aplikasi Bank Sampah dimulai dengan proses identifikasi kebutuhan data dari setiap fitur utama sistem. Tahap ini dilakukan dengan merinci entitas-entitas yang terlibat, seperti pengguna yaitu admin dan warga, transaksi sampah, jenis sampah, dan saldo tabungan. Setelah entitas ditentukan, hubungan antarentitas dianalisis untuk membentuk struktur relasional yang konsisten. Selanjutnya, di dalam lingkungan Appian Designer, pengembang membuat Data Type (tipe data kustom) untuk mendefinisikan struktur masing-masing entitas. Data Type ini kemudian digunakan sebagai dasar dalam membangun Data Store Entity yang merepresentasikan tabel-tabel dalam database. Appian menyediakan fitur visual untuk memetakan field-field dan tipe data dengan antarmuka yang intuitif, sehingga memudahkan perancangan tabel tanpa memerlukan penulisan SQL secara eksplisit.



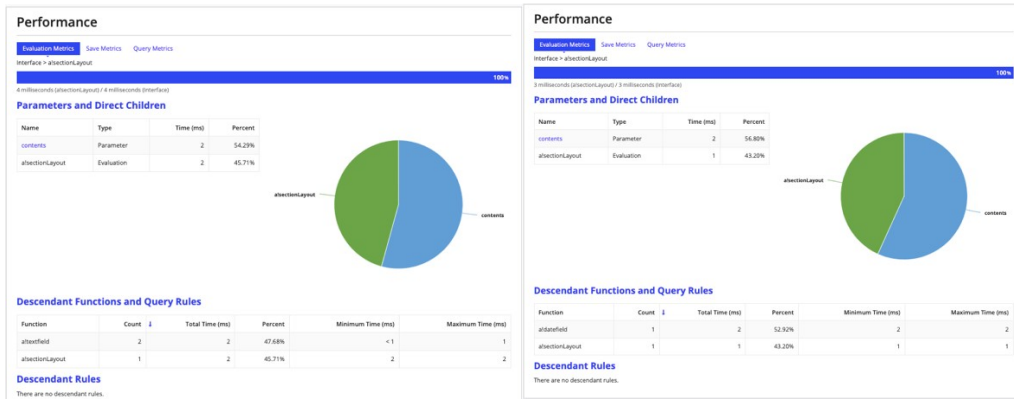
Gambar 9. Diagram Skema Basis Data Relasional Aplikasi Bank Sampah

Dengan pendekatan ini, pengembangan database menjadi lebih cepat, efisien, dan mudah dimodifikasi sesuai kebutuhan aplikasi yang terus berkembang. Tahapan ini menjadi fondasi penting dalam menjamin keberlanjutan proses digitalisasi pengelolaan Bank Sampah yang efisien dan mudah digunakan.

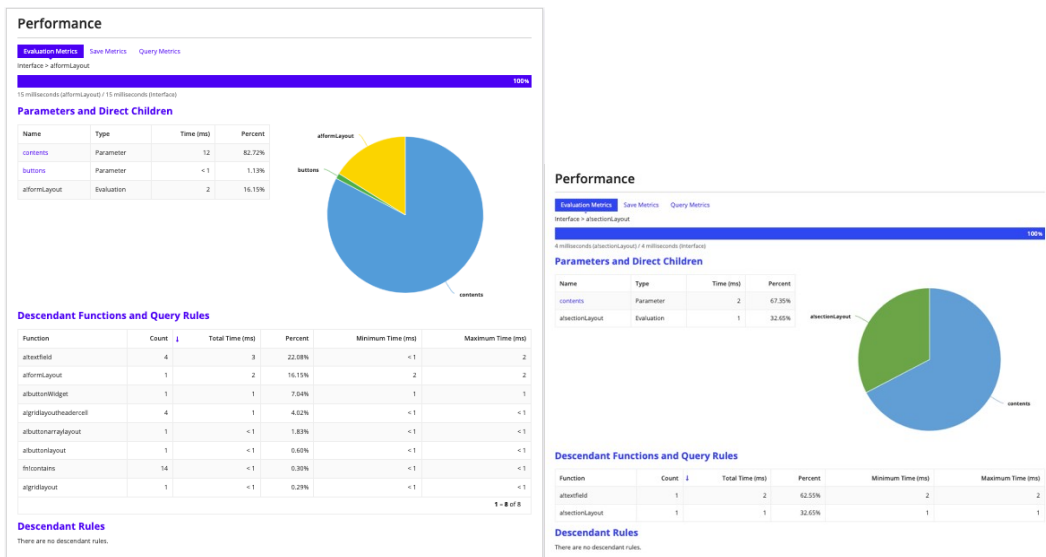
Tabel 4. Struktur Tabel Aplikasi Bank Sampah

No	Nama Tabel	Deskripsi	Fields
1	tbl_registrasi	Input data warga/user	nik (INT) PK, nama (Varchar), alamat (Varchar), hp (Varchar)
2	tbl_jenis_sampah	Input data jenis sampah dan harga per kg	Idj (INT) AI, jenis_sampah (Varchar), harga (INT)
3	tbl_header_setoran	Input data header setoran sampah warga	id_setor (Varchar), tanggal (Date), nik (INT)
4	tbl_detail_setoran	Input data detail setoran sampah warga	id_setor (Varchar), idj (INT), kg (INT)
5	tbl_saldo	Informasi saldo tabungan sampah warga	ids (INT) AI, nik (INT), saldo_update (INT), waktu (TIMESTAMP)

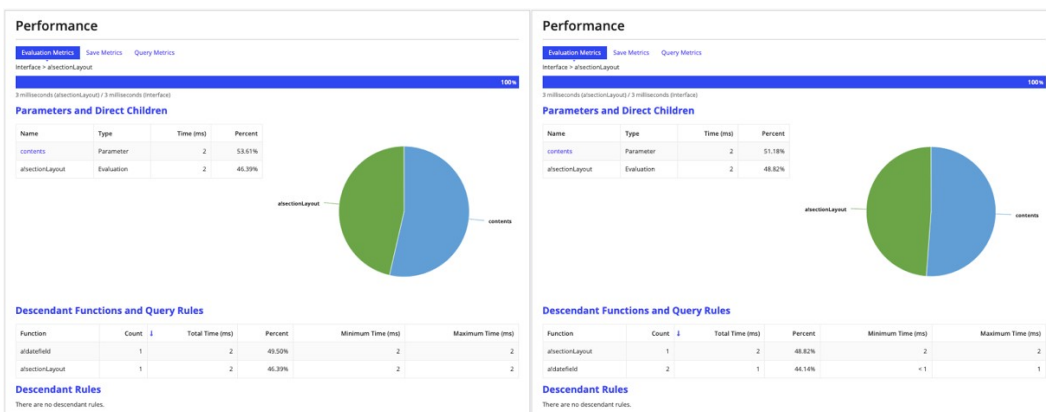
Pengujian performa pada aplikasi Bank Sampah yang dikembangkan menggunakan Appian Designer dilakukan untuk memastikan sistem mampu berjalan secara optimal dalam kondisi penggunaan nyata, khususnya saat menangani banyak transaksi atau pengguna secara bersamaan. Melalui fitur monitoring dan load testing yang disediakan oleh Appian, pengujian ini difokuskan pada kecepatan respon aplikasi, efisiensi proses model, serta kestabilan sistem saat digunakan oleh sejumlah pengguna sekaligus. Hasil pengujian ini diharapkan dapat memberikan gambaran terhadap kemampuan aplikasi dalam mendukung aktivitas operasional Bank Sampah secara berkelanjutan tanpa mengurangi kenyamanan pengguna.



Gambar 10. Performance Test Login dan Setoran Tabungan Sampah



Gambar 11. Performance Test Registrasi dan Menu Warga



Gambar 12. Performance Test Penarikan Saldo dan Laporan

4. KESIMPULAN

Penelitian ini membuktikan bahwa penggunaan Low-Code Development Platform (LCDP) Appian Designer mampu mendukung proses digitalisasi pengelolaan Bank Sampah secara efektif dan efisien. Dengan menerapkan metode RAD, maka proses perancangan dan pembangunan aplikasi dapat dilakukan secara cepat, iteratif, dan fleksibel. Hasil pengujian performa menunjukkan bahwa aplikasi memiliki waktu render yang sangat rendah, yaitu hanya 3 milidetik untuk form login, setoran sampah, dan penarikan saldo; 14 milidetik untuk form registrasi; serta 4 milidetik untuk form laporan dan menu warga. Seluruh form menunjukkan performa yang responsif tanpa adanya bottleneck atau keterlambatan dalam proses loading. Hal ini menandakan bahwa aplikasi yang dibangun melalui Appian sangat cocok digunakan dalam lingkungan sistem real-time maupun sistem dengan banyak pengguna secara bersamaan. Dengan demikian, platform Appian dapat menjadi solusi yang handal dalam pengembangan aplikasi pengelolaan Bank Sampah yang ringan, cepat, dan berkelanjutan.

REFERENSI

- [1] U. Bock, A. C., & Frank, "In search of the essence of low-code: An exploratory study of seven development platforms," in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion*, 2021, pp. 57–66.
- [2] R. Waszkowski, "Low-code platform for automating business processes in manufacturing," *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 376–381, 2019.
- [3] T. Arora, R., Ghosh, N., & Mondal, "Sagitec software studio (S3) – a low code application development platform," in *International Conference on Industry 4.0 Technology*, 2020, pp. 13–17.
- [4] A. Selomo, M., Birawida, A. B., & Mallongi, "Bank Sampah Sebagai Salah Satu Solusi Penanganan Sampah Di Kota Makassar," *J. MKMI*, vol. 12, no. 4, pp. 232-24-, 2016.
- [5] J. Kourouklidis, P., Kolovos, D., Matragkas, N., & Noppen, "Towards a low-code solution for monitoring machine learning model performance," in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2020, pp. 423–430.
- [6] M. Daniel, G., Cabot, J., Deruelle, L., & Derras, "Xatkit: A multimodal low-code chatbot development framework," in *IEEE Access*, 2020, pp. 15332–15346.
- [7] K. R. and M. Kirikova, "Exploring Low-Code Development: A Comprehensive Literature Review," *Complex Syst. Informatics Model. Q.*, vol. 1, no. 36, pp. 68–86, 2023.
- [8] M. Horváth, B., Horváth, Á., & Wimmer, "Towards the next generation of reactive model transformations on low-code platforms: Three research lines," in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2020, pp. 1–10.
- [9] et al. Iyer, C. V. K., "Trinity: A no-code AI platform for complex spatial datasets. ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery," in *GeoAI 2021*, 2021, pp. 33–42.
- [10] J. C. Bexiga, M., Garbatov, S., & Seco, "Closing the gap between designers and developers in a low-code ecosystem," in *ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, 2020, p. Article No.: 61, Pages 1-10.
- [11] Kissflow, "What is Rapid Application Development (RAD) Model?" Accessed: Mar. 21, 2025. [Online]. Available: <https://kissflow.com/application-development/rad/rapid-application-development-model>
- [12] M. K. Karlis Rokis, "Exploring Low-Code Development: A Comprehensive Literature Review," *Complex Syst. Informatics Model. Q.*, vol. 36, pp. 68–86, 2023.
- [13] A. I. Md Abdullah Al Alamin, Sanjay Malakar, Gias Uddin, Sadia Afroz, Tameem Bin Haider, "An Empirical Study of Developer Discussions on Low-Code Software Development Challenges," in *arXiv:2103.11429*, 2021, pp. 46–57. [Online]. Available: <http://dx.doi.org/10.1109/MSR52588.2021.00018>
- [14] T. P. and Zulfachmi, "Perbandingan Metode Pengembangan Perangkat Lunak (Waterfall, Prototype, RAD)," *J. Bangkit Indones.*, vol. 10, no. 1, pp. 6–12, 2021, doi: doi: 10.52771/bangkitindonesia.v10i1.153.
- [15] E. Commission, "Low-Code Engineering Repository Architecture Specification." Accessed: Mar. 27, 2025. [Online]. Available: <https://ec.europa.eu/research/participants/documents/downloadPublic?appId=PPGMS&documentIds=080166e5d671be31>