

# Sistem *Progressive Web App* Pelaporan Sampah Rumah Tangga dengan Penyesuaian Pembayaran dan Insentif Poin Berdasarkan Volume Sampah

Sidiq Amroni<sup>1</sup>, Dilan Ariandi<sup>2</sup>

<sup>12</sup>Program Studi Teknik Informatika, Sekolah Tinggi Teknologi Texmaco, Indonesia  
Email : [sidiq@stttxmaco.ac.id](mailto:sidiq@stttxmaco.ac.id), [djitue@gmail.com](mailto:djitue@gmail.com)

*Received* 29 Agustus 2025 | *Revised* 15 September 2025 | *Accepted* 25 September 2025

## ABSTRAK

Pengelolaan sampah rumah tangga menjadi tantangan tersendiri dalam menjaga kebersihan dan kesehatan lingkungan di Desa Gempolsari. Pengembangan Sistem PWA (*Progressive Web App*) dirancang guna memudahkan warga dalam melaporkan sampah yang mereka hasilkan. Pengembangan sistem ini menggunakan metode XP (Agile Extreme Programming) dengan pendekatan pemrograman berorientasi objek (OOP) yang memungkinkan tim beradaptasi dengan cepat sesuai kebutuhan pengguna. Beberapa tujuan pengembangan sistem antara lain agar warga dapat mudah melihat laporan sampah dan pengelolaannya secara transparansi, melihat insentif poin untuk mendorong semangat partisipasi mereka, melaporkan jenis dan jumlah sampah secara digital, dapat melihat kontribusi warga terhadap pengelolaan sampah agar tercipta lingkungan yang bersih. Setiap laporan yang dibuat, memberikan poin yang dapat ditukarkan dengan penghargaan tertentu. Hasil evaluasi menunjukkan peningkatan signifikan dalam partisipasi masyarakat setelah aplikasi diluncurkan. Transparansi biaya juga meningkat, sehingga masyarakat lebih memahami cara penghitungan biaya pengelolaan sampah. Sistem aplikasi ini berguna dalam mengelola data, memantau pengangkutan sampah dan membuat laporan berkala.

**Kata kunci:** Sistem, *Progressive Web App*, Agile XP, UML

## ABSTRACT

*Household waste management presents a unique challenge in maintaining environmental cleanliness and health in Gempolsari Village. The PWA (Progressive Web App) system was developed to facilitate residents in reporting their waste. The system was developed using the XP (Agile Extreme Programming) method, enabling the team to quickly adapt to user needs. The system's objectives include enabling residents to easily view waste reports and their management transparently, earning incentive points to encourage their participation, digitally reporting the type and quantity of waste, and viewing residents' contributions to waste management to create a clean environment. Each report generates points that can be exchanged for specific awards. Evaluation results showed a significant increase in community participation after the application was launched. Cost transparency also increased, enabling residents to better understand how waste management costs are calculated. This application system is useful for managing data, monitoring waste collection, and generating periodic reports.*

**Keywords:** System, *Progressive Web App*, Agile XP, UML

## 1. PENDAHULUAN

Sampah adalah masalah yang semakin kompleks dan memerlukan penanganan yang tepat. Di Desa Gempolsari, pelaporan sampah yang masih konvensional menyebabkan ketidakakuratan data dan menurunnya partisipasi masyarakat. Sistem pelaporan berbasis PWA diharapkan dapat memberikan solusi efektif yang mengintegrasikan teknologi informasi untuk melibatkan masyarakat dalam pengelolaan sampah dan meningkatkan kepedulian lingkungan. Masalah sampah tidak hanya menjadi masalah regional, tetapi telah berkembang menjadi masalah global yang perlu segera ditangani. Penerapan aplikasi pelaporan sampah rumah tangga yang terintegrasi, seperti pada sistem aplikasi "*sampahQu*", telah terbukti mampu meningkatkan efektivitas pengelolaan sampah ditingkat masyarakat. melalui digitalisasi proses pelaporan dan penyetoran, masyarakat dapat lebih mudah mencatat jenis serta jumlah sampah rumah tangga yang akan disetor, dan pengelola bank sampah dapat melakukan pencatatan serta pelaporan transaksi secara sistematis.

## 2. METODE

### 2.1 Sistem

Konsep sistem pada dasarnya merupakan sebuah kesatuan yang terdiri atas berbagai komponen atau elemen yang saling berhubungan dan membentuk satu kesatuan demi mencapai tujuan tertentu. Setiap elemen dalam sistem tidak berjalan sendiri-sendiri, melainkan harus terhubung dan berinteraksi sehingga mampu menjalankan fungsi yang telah ditetapkan. Dengan kata lain, sistem adalah sebuah entitas yang bersifat kompleks dan dinamis karena melibatkan banyak unsur yang bekerja sama agar tujuan atau fungsi sistem dapat tercapai secara optimal[1].

### 2.2 PWA (*Progressif Web App*)

PWA merupakan salah satu inovasi di bidang pengembangan web yang bertujuan memberikan pengalaman layaknya aplikasi native pada perangkat mobile. PWA merupakan konsep pengembangan aplikasi web yang memanfaatkan teknologi terbaru pada browser seperti web app manifest, service worker, dan app shell. PWA dapat berjalan lintas platform dengan satu kali proses pembuatan, sehingga lebih hemat waktu dan biaya[2].

### 2.2 UML (*Unified Modeling Language*)

UML adalah bahasa pemodelan standar yang digunakan dalam rekayasa *software* untuk memvisualisasikan, menentukan, membangun dan mendokumentasikan artefak sistem *software* [3]. Bahasa ini berakar pada metode pemrograman berorientasi objek yang dikembangkan pada akhir 1980-an dan awal 1990-an, dan diterbitkan sebagai standar yang disetujui oleh Organisasi Internasional untuk Standardisasi (ISO) pada tahun 2005. UML membantu para insinyur *software* dan arsitek sistem dalam pemodelan, desain, dan analisis, memfasilitasi komunikasi dan eksplorasi desain potensial. Beberapa diagram UML, antara lain *use case*, *activity diagram*, *class diagram*, dan *sequence diagram*.

#### 1. *Usecase Diagram*

Diagram ini bersifat statis dan digunakan untuk memperlihatkan hubungan antara aktor (pengguna atau sistem lain) dan berbagai *usecase* (fitur atau fungsi utama) yang terdapat dalam sistem. *Usecase* menjadi gambaran fungsionalitas yang mempresentasikan sebuah interaksi antara aktor dan sistem[4].

#### 2. *Activity Diagram*

Diagram aktivitas termasuk tipe diagram dinamis yang khusus digunakan untuk menggambarkan alur aktivitas atau bagaimana satu proses dalam sistem berpindah dari satu tahap ke tahap berikutnya[5].

### 3. *Sequence Diagram*

Sequence diagram juga bersifat dinamis dan menitikberatkan pada urutan interaksi antar objek dalam sistem. Melalui diagram ini, pengembang dapat melihat bagaimana pesan atau informasi mengalir dari satu objek ke objek lain seiring berjalannya waktu, sehingga sangat membantu dalam memetakan proses-proses yang saling berketertgantungan[6].

### 4. *Class Diagram*

*Class diagram* merupakan diagram statis yang menggambarkan struktur kelas-kelas dalam sistem, termasuk atribut, metode, serta hubungan antar kelas. *Class diagram* membantu memetakan *blueprint* sistem yang akan dibangun sehingga komponen-komponen utamanya dapat diidentifikasi dengan jelas[7].

## 2.3 ERD (*Entity Relationship Diagram*)

ERD dibuat dengan tujuan untuk melihat keterhubungan antar entitas yang nantinya dijadikan sebagai acuan dalam pembuatan tabel pada database. Tujuan dari pengumpulan data pada penelitian ini adalah untuk menambah wawasan terkait desain ERD dan menjawab pertanyaan-pertanyaan yang dibutuhkan oleh user yang mengalami masalah dalam pembuatan database. ERD alat bantu dalam pembuatan database dan memberikan gambaran bagaimana kerja database yang akan dibuat[8].

## 2.4 *Agile XP (Extreme Programming)*

*Agile XP* merupakan metode pengembangan perangkat lunak yang lahir dari kebutuhan akan fleksibilitas dan respon cepat dalam menghadapi perubahan di dunia teknologi yang dinamis. *XP* membantu memastikan bahwa perangkat lunak yang dibangun benar-benar sesuai kebutuhan dan lebih mudah menyesuaikan diri dengan perubahan dimasa mendatang[9]. Tahapan-tahapan dalam metode *Agile XP* yang diterapkan:

### 1. *Planning* (Perencanaan)

- a. Melakukan diskusi langsung dengan calon pengguna untuk mengidentifikasi kebutuhan, keinginan, dan permasalahan terkait pelaporan sampah.
- b. Mengumpulkan ide, saran, serta kendala yang dihadapi pengguna sebagai dasar pengembangan sistem.
- c. Menyusun daftar prioritas fitur dan menentukan alur kerja aplikasi berdasarkan hasil diskusi.

### 2. *Design* (Perancangan)

- a. Menyusun struktur basis data dan membangun *flowchart* atau bagan alir proses aplikasi.
- b. Memetakan relasi antar objek dalam sistem sehingga seluruh fitur dapat saling terhubung dengan baik.

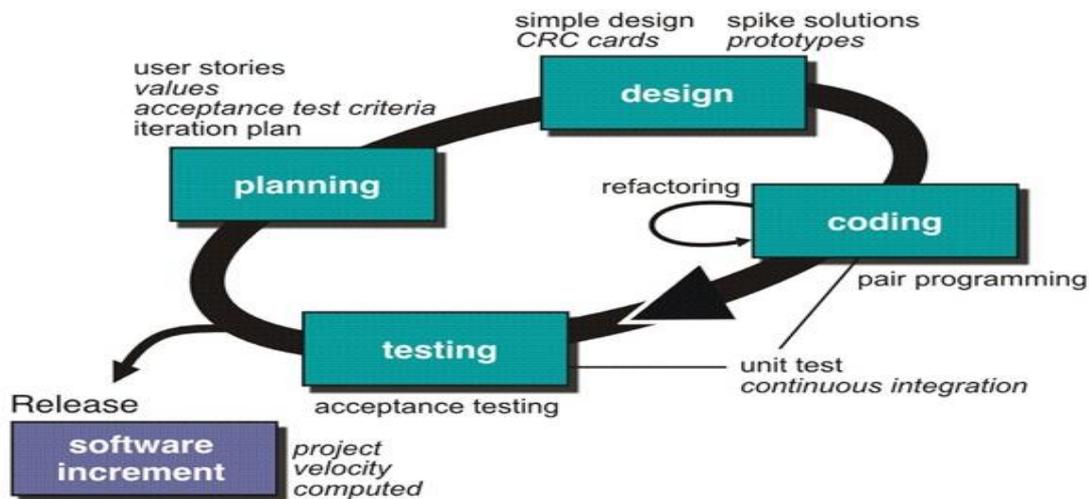
### 3. *Coding* (Pengkodean)

- a. Menerjemahkan desain dan perancangan ke dalam kode program sesuai prinsip OOP (*Object Oriented Programming*).
- b. Mengembangkan fitur-fitur utama aplikasi secara bertahap dan terstruktur.
- c. Melakukan penyesuaian atau perbaikan kode jika terdapat masukan atau perubahan kebutuhan dari pengguna.

### 4. *Testing* (Pengujian)

- a. Melakukan uji coba aplikasi secara internal untuk menemukan *bug* atau kesalahan sistem.

Berikut Gambar 1 menunjukkan ilustrasi tahapan metode *Agile XP* [10]



**Gambar 1 Tahapan-tahapan Metode *Agile XP* (*Extreme Programming*)**  
(Sumber: L. Suryani, E. Murniyasih, M. P. Saptono, R. F. Waliulu, and ..., "Pengembangan aplikasi bank sampah dengan metode extreme programming,").

### 3. HASIL DAN PEMBAHASAN

#### 3.1 *Planning* (Perencanaan)

Tahap perencanaan merupakan fase awal dalam pengembangan sistem yang sangat penting untuk menentukan arah dan keberhasilan proyek. Pada tahap ini dilakukan analisis mendalam terhadap kebutuhan sistem, identifikasi masalah yang akan diselesaikan, serta perencanaan strategi pengembangan yang tepat. Perencanaan dimulai dengan studi kelayakan untuk menentukan apakah sistem layak dikembangkan dari segi teknis, ekonomis, dan operasional. Dilakukan juga analisis kebutuhan fungsional dan non-fungsional yang harus dipenuhi oleh sistem, termasuk identifikasi stakeholder dan pengguna akhir yang akan menggunakan aplikasi.

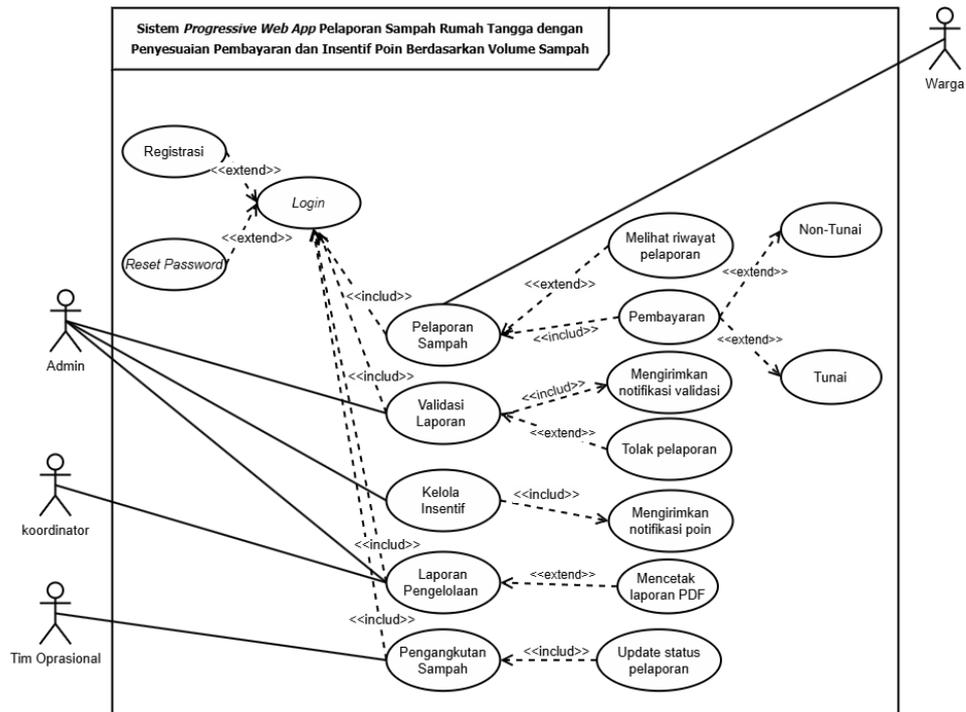
#### 3.2 *Design* (Perancangan)

Pada tahap ini dilakukan perancangan dengan beberapa model diagram untuk mempermudah merancang sekaligus analisis keterhubungan antar komponen yang terlibat. Beberapa diagram yang dibuat diantaranya :

1. *Usecase Diagram* Sistem Usulan

Gambar 2 ditampilkan seluruh *usecase* yang ada dalam sistem lapor sampah beserta aktor-aktor yang terlibat. Sistem ini melibatkan tiga aktor utama yaitu Warga, Koordinator, Admin dan Tim Operasional, dimana masing-masing memiliki hak akses dan fungsi yang berbeda sesuai peran dalam pengelolaan sampah. Terlihat terdapat tujuh belas *usecase* pada gambar tersebut dan terdiri dari empat aktor.

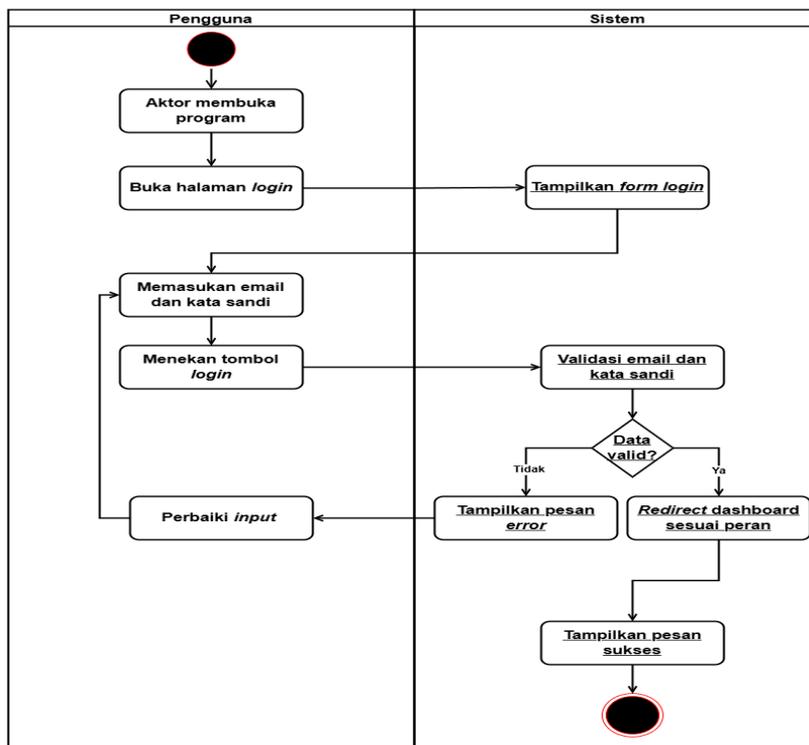
Sistem *Progressive Web App* Pelaporan Sampah Rumah Tangga dengan Penyesuaian Pembayaran dan Insentif Poin berdasarkan Volume Sampah



**Gambar 2 Usecase Sistem yang diusulkan**

2. *Activity Diagram Sistem Usulan*

*Activity Diagram* yang diusulkan berfungsi untuk menggambarkan alur kegiatan dalam sistem pelaporan sampah PWA. Diagram ini memberikan pandangan yang jelas tentang bagaimana pengguna akan berinteraksi dengan sistem dalam berbagai situasi, serta tahapan yang diperlukan untuk setiap aktivitas.

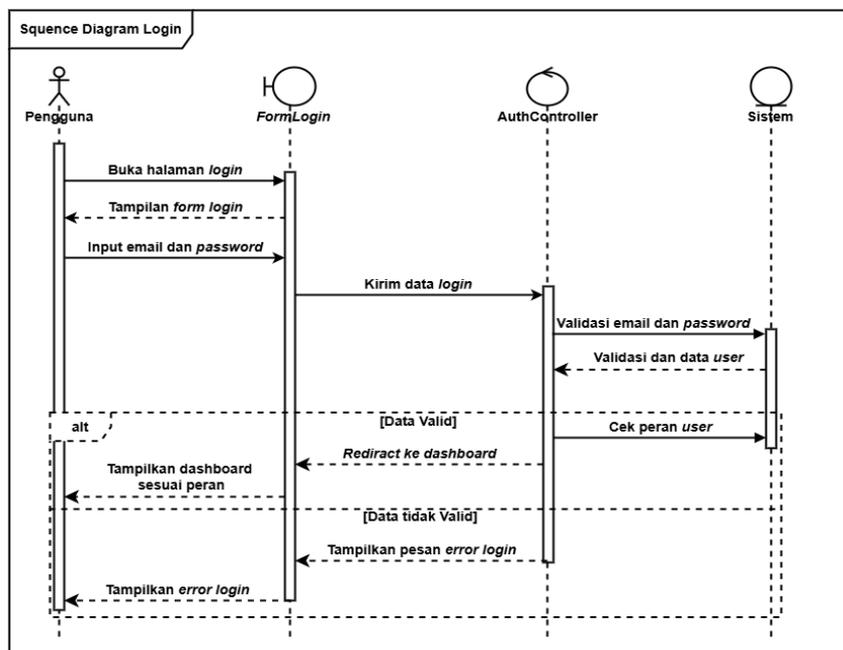


**Gambar 3 Activity Diagram Login sistem yang diusulkan**

Gambar 3 merupakan salah satu contoh *activity diagram* sistem yang diusulkan yaitu *activity diagram login*. Selain *activity diagram login* pada tahap perancangan ini juga membuat beberapa *activity diagram* yang lain diantaranya *activity diagram* registrasi, reset password, pelaporan sampah, pembayaran, validasi laporan sampah, penagihan tunai, kelola insentif, kelola biaya dan pembuatan laporan pengelolaan sampah. Pada artikel ini tidak semua gambar *activity diagram* ditampilkan karena membutuhkan banyak halaman untuk menampilkan semuanya. Sebagai gambaran penulis hanya menyertakan salah satu contoh *activity diagram* dari sistem yang diusulkan berupa *activity diagram login* sesuai Gambar 3 tersebut.

### 3. *Sequence Diagram* Sistem Usulan

*Sequence Diagram* adalah komponen kunci dari *Unified Modeling Language* (UML) yang digunakan untuk menggambarkan bagaimana objek berinteraksi secara berurutan. *Sequence diagram* yang telah dibuat pada perancangan sistem ini diantaranya *sequence diagram login*, pelaporan sampah, pembayaran, pengangkutan sampah, kelola insentif dan laporan pengelolaan.



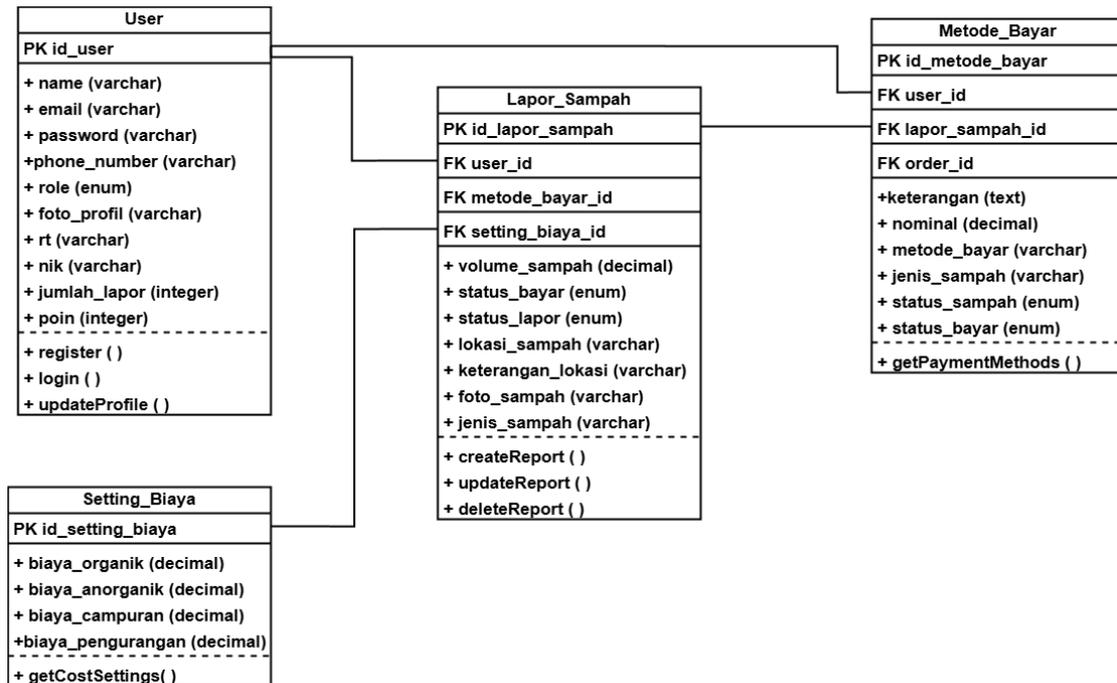
**Gambar 4 *Sequence Diagram Login***

Gambar 4 merupakan salah satu contoh *sequence diagram* yang telah dibuat dalam perancangan sistem ini.

### 4. *Class Diagram* Sistem Usulan

Pada Gambar 5 ditampilkan *class diagram* yang menggambarkan struktur utama dari sistem. Diagram ini memperlihatkan berbagai kelas (*class*) yang digunakan dalam aplikasi beserta atribut dan relasi antar kelas tersebut. Misalnya, terdapat class *User* yang memiliki atribut seperti nama, email, dan *password*, serta relasi dengan class lain seperti *LaporSampah*, *MetodeBayar*, dan *Notifikasi*.

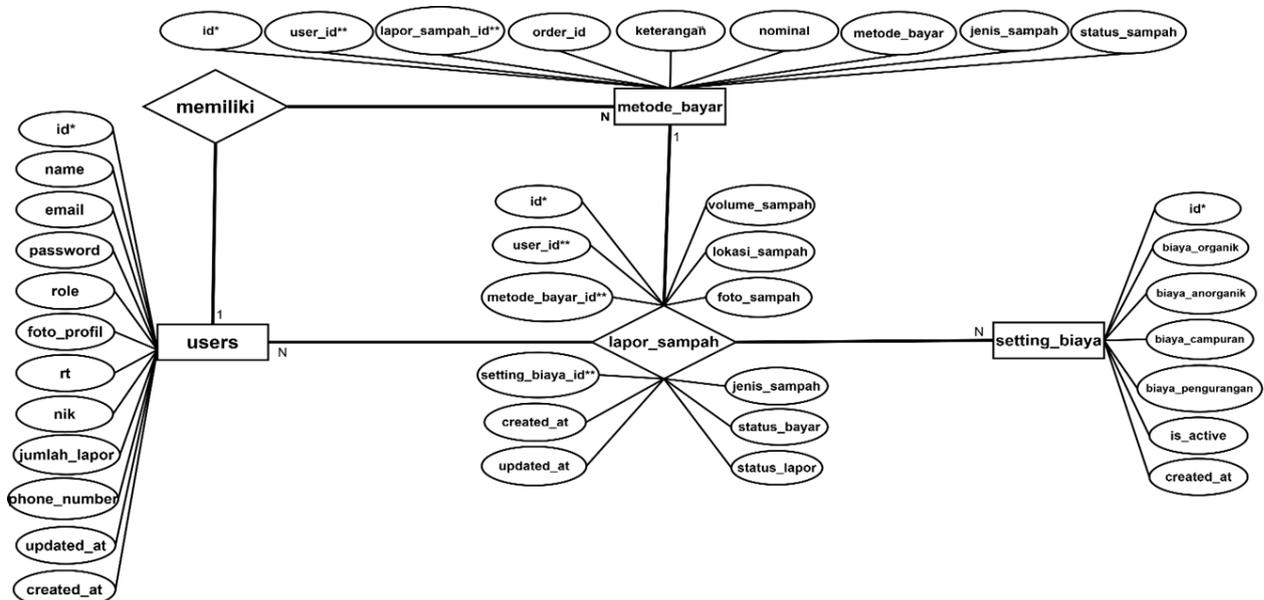
Sistem *Progressive Web App* Pelaporan Sampah Rumah Tangga dengan Penyesuaian Pembayaran dan Insentif Poin berdasarkan Volume Sampah



Gambar 5 Class Diagram Sistem Usulan

5. ERD (*Entity Relationship Diagram*) Sistem Usulan

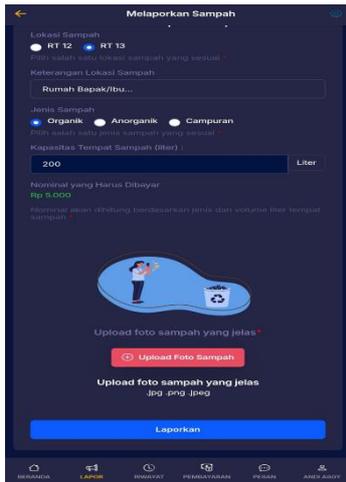
Gambar 6 menunjukkan pemetaan serta hubungan entitas kunci, seperti pengguna, laporan sampah, metode pembayaran, dan pengaturan biaya.



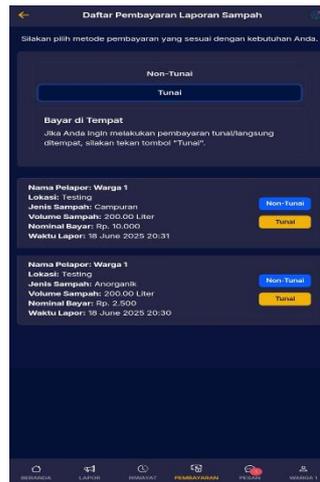
Gambar 6 ERD Sistem Usulan

3.3 Coding

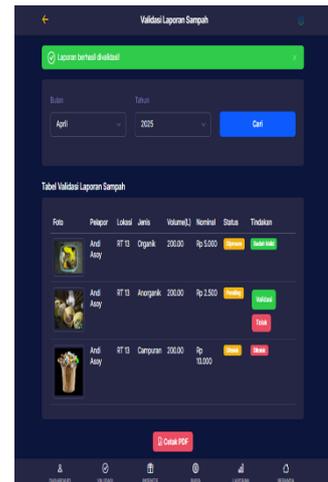
Setelah merancang sistem dengan diagram dan spesifikasi yang jelas, langkah selanjutnya adalah pengkodean. Dalam konteks sistem pelaporan sampah ini, mengutamakan penggunaan teknologi *web* modern untuk memastikan aplikasi tidak hanya fungsional, tetapi juga responsif dan ramah pengguna. Berikut Gambar 7-9 merupakan tampilan setelah dilakukan pengkodean.



**Gambar 7**  
Implementasi Halaman  
Lapor Sampah



**Gambar 8**  
Implementasi Halaman  
Pembayaran



**Gambar 9**  
Implementasi Halaman  
Validasi Laporan

### 3.4 Testing (pengujian)

Pengujian digunakan metode *White Box Testing*, yang berfokus pada internal struktur dan logika aplikasi. Metode ini memungkinkan kami untuk mengevaluasi arsitektur kode dan memastikan bahwa semua jalur eksekusi dalam aplikasi diuji secara menyeluruh. Tabel 1 berikut menunjukkan hasil pengujian menggunakan *white box testing*.

**Tabel 1 Hasil Testing (pengujian)**

| No. | Fitur yang Diuji     | Skenario Pengujian  | Input                                | Output yang Diharapkan                               |
|-----|----------------------|---|--------------------------------------|--|
| 1.  | Login                | Pengguna login dengan kombinasi email dan kata sandi benar/salah          | Email & Kata Sandi                   | Berhasil masuk atau muncul pesan <i>error</i>        |
| 2.  | Registrasi           | Pengguna mengisi <i>form</i> registrasi dengan data valid dan tidak valid | Nama, email, kata sandi, NIK,        | Registrasi berhasil/gagal sesuai validasi            |
| 3.  | Reset Password       | Pengguna melakukan <i>reset</i> kata sandi melalui email                  | Email, kata sandi baru, <i>token</i> | Kata sandi berhasil direset, bisa <i>login</i> ulang |
| 4.  | Pelaporan Sampah     | Pengguna mengisi <i>form</i> dan pelaporan mengunggah foto                | Lokasi, jenis, volume, foto          | Laporan tersimpan dan diarahkan ke pembayaran        |
| 5.  | Pembayaran Non-Tunai | Pengguna memilih pembayaran non-tunai dan menyelesaikan transaksi         | Pilih metode, lakukan pembayaran     | Status pembayaran berubah menjadi lunas              |
| 6.  | Pembayaran Tunai     | Pengguna memilih pembayaran tunai, tim operasional menandai lunas         | Pilih tunai, konfirmasi pembayaran   | Status pembayaran berubah menjadi lunas              |
| 7.  | Riwayat Laporan      | Pengguna melihat riwayat laporan dan detail status                        | Tekan tombol menu riwayat            | Daftar laporan tampil sesuai data pengguna           |
| 8.  | Validasi Laporan     | Admin memvalidasi atau menolak laporan                                    | Pilih laporan, validasi/tolak        | Status laporan berubah, notifikasi terkirim          |

| No. | Fitur yang Diuji    | Skenario Pengujian  | Input   | Output yang Diharapkan                                |
|-----|---------------------|---|---|---|
| 9.  | Pengangkutan        | Tim operasional menandai laporan sebagai sudah diangkut                             | Pilih laporan, tekan tombol "Tandai Diangkut" | Status laporan berubah menjadi "diangkut"             |
| 10. | Penagihan Tunai     | Tim operasional melakukan penagihan dan menandai pembayaran sebagai lunas           | Pilih laporan, tekan tombol "Tandai Lunas"    | Status pembayaran berubah menjadi lunas               |
| 11. | Kelola Insentif     | Admin menambah poin ke pengguna   | Pilih <i>user</i> , <i>input</i> poin         | Poin bertambah dan notifikasi terkirim                |
| 12. | Kelola Biaya        | Admin mengubah biaya sampah dan menyimpan pengaturan baru                           | <i>Input</i> biaya baru                       | Biaya <i>terupdate</i> dan berlaku untuk laporan baru |
| 13. | Laporan Pengelolaan | Admin dan Koordinator dapat melihat laporan statistik dan grafik pengelolaan sampah | Pilih filter, lihat grafik/tabel              | Data statistik dan grafik tampil sesuai filter        |

#### 4. KESIMPULAN

Pengembangan sistem *Progressive Web App* untuk pelaporan sampah rumah tangga di Desa Gempolsari berhasil mengoptimalkan pengelolaan sampah yang lebih efisien dan meningkatkan partisipasi masyarakat. Penelitian selanjutnya dapat mengembangkan fitur analisis data untuk mendukung keputusan pengelolaan sampah yang lebih baik di tingkat lokal.

#### 5. DAFTAR RUJUKAN

- [1] S. Amroni and N. Astuti, "Sistem Pendaftaran Puskesmas Berbasis Web: Studi kasus: Puskesmas Tegalwaru," *INFOTEX J. Ilm. Bid. Ilmu Tek.*, vol. 2, no. 2, pp. 325–337, 2024.
- [2] A. Samsudin and A. I. Sanjaya, "Sistem Pemesanan Katering Berbasis Website," vol. 3, no. 1, pp. 274–285, 2024.
- [3] S. W. Ramdany, S. Aulia Kaidar, B. Aguchino, C. Amelia, A. Putri, and R. Anggie, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web." *Journal of Industrial and Engineering System (JIES)* vol. 5 no.1 pp. 30-31, 2024
- [4] S. W. Ramdany, S. Aulia Kaidar, B. Aguchino, C. Amelia, A. Putri, and R. Anggie, "Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web."
- [5] Kurniawan. Hamid, et al, "Penerapan Metode Waterfall Dalam Perancangan Sistem Informasi Penggajian Pada SMK Bina Karya Karawang", *Jurnal Interkom: Jurnal Publikasi Ilmiah Bidang Teknologi Informasi dan Komunikasi* Vol 14 No 04 Bulan Januari, Tahun 2020.
- [6] A. H. Hendri and Mochammad Arief Sutisna, "Article Desktop Based National Police

Commission Activities Information System," *Jurnal CoSciTech (Computer Science and Information Technology)*, vol. 2, no. 1, pp. 14–23, Jun. 2021, doi: 10.37859/coscitech.v2i1.2393

- [7] Sinambela, M. R., Waidah, D. F., Susilo, T., Jaya, N. A., & Friansyah, I. G. (2024). Rancang Bangun Perpustakaan Digital Berbasis Website pada SD Swasta 001 PT. KG Meral Barat di Kabupaten Karimun. *Tikar: Jurnal Teknik Informatika Karimun*, 5(1), 12–23. [https://doi.org/https://doi.org/10.51742/teknik\\_informatika.v5i1.1231](https://doi.org/https://doi.org/10.51742/teknik_informatika.v5i1.1231)
- [8] K. Afifah, Z. F. Azzahra, and A. D. Anggoro, " Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review" , *Jurnal Intech* vol. 3, no. 1 2022
- [9] Y. Irawan Chandra, D. Ruri Irawati, M. Riastuti, and S. Jakarta STI, "Penerapan Model Agile-Extreme Programming (XP) Dalam Membuat Aplikasi Pengenalan Daerah Wisata di Wonogiri Berbasis Web", doi: 10.37817/ikraith-informatika.v8i1.
- [10] A. Samsudin and H. Hamdalah Islami, "Sistem Pengaduan Masyarakat Menggunakan Metode Agile Extreme Programming." *Jurnal Infotex* vol. 2 no. 1 pp.214-226, 2023