

IMPLEMENTASI SISTEM MONITORING BANK SAMPAH ANORGANIK BERBASIS WEB (E-MONSTER)

Rendy Christian^[1];Muhamat Andre^[2];Eva Meilinda^[3]; Tarina Dashela^[4]

^[1,2,3,4]Program Studi Sistem Informasi, Fakultas Teknik dan Informatika, Universitas Bina Sarana Informatika
19221146@bsi.ac.id, 19221030@bsi.ac.id, eva.emd@bsi.ac.id, tarina.tdl@bsi.ac.id

INFO ARTIKEL	INTISARI
Diajukan : 09-05-2026	Pengelolaan sampah anorganik pada Bank Sampah di bawah naungan Dinas Perumahan Rakyat dan Kawasan Permukiman, Pertanahan dan Lingkungan Hidup Kabupaten Bengkayang saat ini masih menghadapi kendala efisiensi akibat pencatatan data nasabah dan transaksi setoran yang bersifat manual. Proses konvensional tersebut rentan terhadap risiko kehilangan dokumen, kesalahan kalkulasi nominal saldo, serta keterbatasan akses informasi bagi nasabah. Penelitian ini bertujuan untuk merancang dan membangun sebuah aplikasi monitoring terpadu berbasis web yang dinamakan E-MONSTER (Elektronik Monitoring Sampah Terpadu). Metode pengembangan sistem yang digunakan dalam penelitian ini adalah metode Waterfall, meliputi tahap analisis kebutuhan, perancangan sistem, penulisan kode program, hingga pengujian aplikasi. Hasil dari penelitian ini adalah sebuah sistem informasi terintegrasi dengan arsitektur Model-View-Controller (MVC) berbasis framework CodeIgniter 4. Sistem ini berhasil memfasilitasi Admin Dinas dalam memantau grafik volume sampah, membantu Petugas Lapangan dalam mendigitalisasi transaksi penimbangan secara real-time, serta mempermudah Nasabah dalam memonitor saldo tabungan secara transparan. Implementasi E-MONSTER terbukti mampu meningkatkan akurasi data, meminimalisir human error, serta mewujudkan transparansi manajemen bank sampah yang efektif.
Diterima : 20-05-2026	
Diterbitkan: 01-06-2026	
Kata Kunci : <i>Bank Sampah, Bengkayang, CodeIgniter 4, Sampah, Sistem Monitoring</i>	

I. PENDAHULUAN

Pengelolaan sampah anorganik pada unit bank sampah di bawah naungan Dinas Perumahan Rakyat dan Kawasan Permukiman, Pertanahan dan Lingkungan Hidup Kabupaten Bengkayang saat ini masih menghadapi kendala efisiensi. Berdasarkan hasil observasi lapangan dan wawancara langsung dengan Kepala Bidang Lingkungan Hidup, ditemukan bahwa pencatatan data transaksi setoran sampah anorganik masih dilakukan secara manual menggunakan buku *ledger*. Proses konvensional tersebut rentan terhadap risiko kesalahan perhitungan (*human error*), lambat dalam proses pelaporan, serta minim transparansi bagi masyarakat selaku nasabah bank sampah. Untuk mengatasi permasalahan tata kelola tersebut, diperlukan sebuah sistem informasi monitoring terintegrasi yang mampu mendigitalisasi seluruh proses bisnis secara tepat dan *real-time*. Implementasi platform berbasis web diharapkan dapat meminimalisir kendala administratif tersebut melalui fitur otomatisasi pencatatan transaksi, akses data yang sistematis,

serta penyediaan laporan yang akurat bagi pihak pengelola maupun nasabah (Prabowo 2022).

Dalam merencanakan solusi digital ini, analisis kebutuhan merupakan salah satu fase paling esensial dalam proses pengembangan perangkat lunak. Pada fase ini, pengembang melakukan identifikasi komprehensif terhadap seluruh kebutuhan yang wajib dipenuhi oleh sistem yang akan dibangun, meliputi aspek fungsional maupun non-fungsional agar terhindar dari risiko gagal memenuhi ekspektasi pengguna akhir. Pendekatan ini mencakup tahapan metodologis seperti analisis kebutuhan, perancangan sistem, implementasi kode program, serta pengujian berkala untuk memastikan reliabilitas platform sebelum diterapkan secara penuh (Ni Made Mila Rosa Desmayani , Luh Gede Bevi Libraeni , S.E., M.Si 2024).Proses ini berfungsi sebagai dasar pijakan dalam perancangan arsitektur sistem, penentuan teknologi, serta penetapan batasan sistem secara keseluruhan (Dewi et al. 2023).Selain itu, tahapan analisis ini secara spesifik memetakan interaksi antara nasabah dan petugas, mulai dari pendaftaran anggota, penimbangan sampah, hingga pelaporan

aktivitas transaksi yang terekam secara otomatis dalam basis data terpusat (Atin et al. 2022). Spesifikasi kebutuhan ini menjabarkan definisi sistem secara detail, meliputi identifikasi dan deskripsi abstraksi sistem perangkat lunak yang mendasar beserta hubungannya (Ahmad Herdinal Muttaqin, Adian fatchur rochim 2022).

Oleh karena itu, penelitian ini bertujuan untuk mengimplementasikan aplikasi E-MONSTER (Elektronik Monitoring Sampah Terpadu) berbasis web. Analisa kebutuhan sistem E-MONSTER mencakup empat komponen utama, yaitu pemilihan *framework* pengembangan yang tepat, penerapan arsitektur perangkat lunak *Model-View-Controller* (MVC), identifikasi kebutuhan fungsional sistem, serta identifikasi kebutuhan non-fungsional sebagai tolok ukur kualitas layanan sistem. Melalui pendekatan sistematis ini, diharapkan aplikasi E-MONSTER dapat mentransformasikan manajemen bank sampah konvensional menjadi ekosistem digital yang akurat dan transparan. Integrasi fitur-fitur tersebut dirancang untuk mengatasi hambatan operasional administratif dan meningkatkan partisipasi aktif masyarakat dalam pengelolaan sampah berkelanjutan (Erwin, Ni'matullah Muin, Marni 2025).

II. BAHAN DAN METODE

Penelitian ini diawali dengan tahap pengumpulan data melalui observasi langsung dan wawancara terkait pengelolaan alur administrasi bank sampah anorganik pada Dinas Perumahan Rakyat dan Kawasan Permukiman, Pertanahan dan Lingkungan Hidup Kabupaten Bengkayang. Bahan utama yang digunakan dalam pengembangan ini meliputi data master kategori sampah, data nasabah, serta skema tarif konversi rupiah per kilogram. Selanjutnya, metodologi pengembangan perangkat lunak menggunakan model *waterfall* diadopsi untuk menjamin alur kerja yang sistematis melalui tahapan analisis kebutuhan, desain sistem, implementasi kode, serta pengujian fungsionalitas aplikasi (Zumrotuz Zakiyah, Tri Listyorini 2024).

Dalam konteks penelitian ini, sistem dirancang untuk mengintegrasikan manajemen bank sampah dari hulu ke hilir. Platform web dipilih sebagai media implementasi aplikasi E-MONSTER karena sifatnya yang dinamis, mudah diakses tanpa batasan waktu dan ruang, serta mendukung integrasi data secara terpusat menggunakan arsitektur *framework* CodeIgniter 4 yang memungkinkan pengelolaan transaksi secara lebih efisien dibandingkan metode konvensional (Putra 2025). Pemilihan *framework*

ini didukung oleh penggunaan basis data MySQL sebagai sistem manajemen basis data yang andal untuk menjamin konsistensi serta integritas data operasional. Selain itu, arsitektur ini memungkinkan pembagian hak akses pengguna secara hierarkis guna memastikan keamanan serta akuntabilitas dalam pengelolaan data di setiap unit terkait, baik untuk Admin Dinas, Petugas Lapangan, maupun Nasabah.

Secara konseptual, sistem merupakan sekumpulan komponen atau elemen yang saling berinteraksi secara terintegrasi untuk mencapai tujuan tertentu, yang dalam konteks sistem informasi, diwujudkan melalui integrasi komponen data dan prosedur terstruktur untuk mengolah informasi secara efisien. Dengan demikian, penerapan teknologi web ini mampu menanggulangi kendala klasik seperti risiko kehilangan data, duplikasi entri, serta penumpukan laporan manual yang sering menghambat efektivitas operasional bank sampah (Nicho Yuda Mahendra, Lukman Nulhakim, Andri Irawan 2024).

dengan pendekatan ini, sistem mampu menyajikan pangkalan data yang lengkap dan akurat, sehingga meminimalisir kendala manual dalam proses akademik serta meningkatkan efektivitas koordinasi antara mahasiswa dan dosen pembimbing. Selain itu, penggunaan metode *waterfall* dalam pengembangan sistem ini memastikan alur kerja yang terstruktur dan terukur guna meminimalisir risiko kegagalan pada setiap tahapan implementasi (Rahmah 2022).

Metode pengembangan perangkat lunak yang diterapkan dalam membangun aplikasi E-MONSTER adalah metode *Waterfall*. Metode ini dipilih karena kemampuannya dalam memberikan alur kerja yang terstruktur dan sistematis melalui tahapan analisis kebutuhan, desain, implementasi, hingga pengujian.

Penerapan tahapan *Waterfall* dalam penelitian ini dijabarkan sebagai berikut:

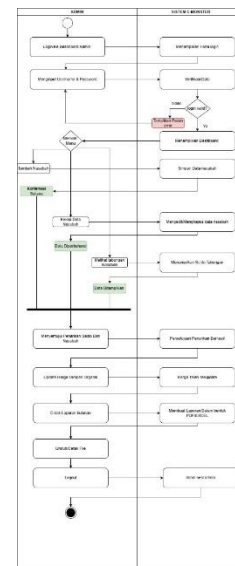
1. Analisis Kebutuhan Sistem: Melakukan bedah alur berjalan untuk memetakan kebutuhan fungsional bagi ketiga aktor (Admin, Petugas, dan Nasabah).
2. Desain Sistem (*Design*): Merancang pemodelan sistem menggunakan diagram *Unified Modeling Language* (UML) guna menspesifikasikan, menggambarkan, dan mendokumentasikan artefak dari sistem berorientasi objek. Tahap perancangan ini dimodelkan secara terpadu melalui *Use Case Diagram*, *Activity Diagram*, dan *Sequence Diagram*.

- ## 2. Activity Diagram

1. Use Case Diagram



Gambar 2. *Activity Diagram Admin*
Sumber: Christian & Andre (2026)

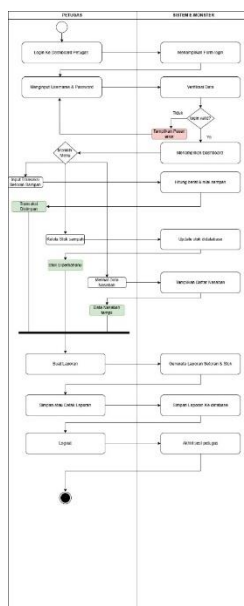


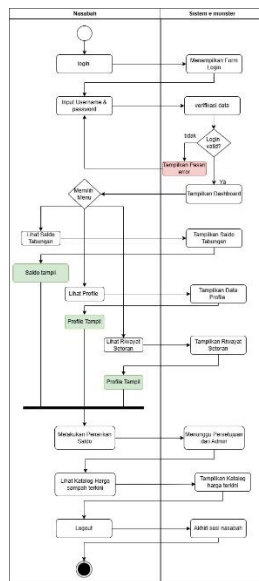
- a. Proses Autentikasi: Admin memulai aktivitas dengan mengakses halaman *login*, kemudian sistem menampilkan form *login*. Setelah Admin menginput *username* dan *password*, sistem melakukan verifikasi data. Jika data tidak valid, sistem menampilkan pesan *error* dan mengarahkan Admin kembali ke menu input. Jika valid, sistem menampilkan halaman utama berupa *dashboard* Admin.

- Implementasi Sistem Monitoring

- Gambar 3. *Activity Diagram* Petugas Lapangan
Sumber: Christian & Andre (2026)

- a. Proses Autentikasi: Petugas memulai aktivitas dengan mengakses halaman *login* ke *dashboard* Petugas, kemudian sistem merespons dengan menampilkan form *login*. Setelah Petugas menginput *username* dan *password*, sistem melakukan verifikasi data.
- b. Jika data tidak valid, sistem menampilkan pesan *error* dan mengarahkan kembali ke menu input. Jika valid, sistem menampilkan halaman utama berupa *dashboard* Petugas.
- c. Manajemen Transaksi dan Operasional Lapangan: Melalui pilihan menu yang tersedia, Petugas dapat menjalankan tiga fungsi utama. Jika memilih menu Input Transaksi Setoran Sampah, sistem akan menghitung berat serta nilai konversi rupiah dari sampah tersebut hingga memunculkan konfirmasi transaksi disimpan. Jika memilih menu Kelola Stok Sampah, sistem akan melakukan *update* stok di *database* hingga muncul status stok diperbaharui. Petugas juga dapat memilih menu Melihat Data Nasabah untuk memicu sistem menampilkan daftar nasabah yang terdaftar.
- d. Pelaporan dan Terminasi Sesi: Setelah aktivitas transaksi lapangan selesai, Petugas dapat memilih menu Buat Laporan. Sistem akan merespons dengan melakukan *generate* laporan setoran dan stok. Petugas kemudian dapat memilih opsi Simpan atau Cetak Laporan, di mana sistem akan menyimpan dokumen laporan tersebut ke dalam *database*. Aktivitas ditutup secara aman ketika Petugas memilih tombol *logout* dan sistem mengakhiri sesi Petugas.





Gambar 4. Activity Diagram Nasabah
Sumber: Christian & Andre (2026)

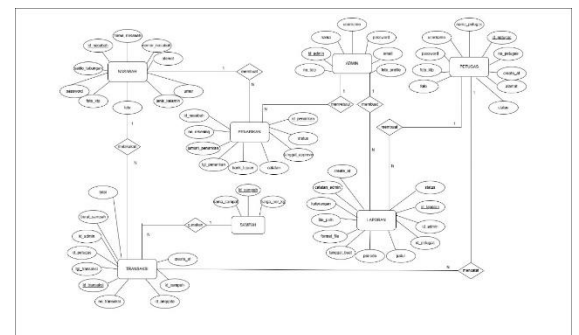
Berdasarkan Gambar 4, alur aktivitas yang dilakukan oleh aktor Nasabah pada sistem E-MONSTER dijabarkan sebagai berikut:

- Proses Autentikasi: Nasabah memulai aktivitas dengan mengakses menu *login*, kemudian sistem merespons dengan menampilkan form *login*. Setelah Nasabah memasukkan *username* dan *password*, sistem melakukan verifikasi data. Jika data tidak valid, sistem menampilkan pesan *error* dan mengarahkan kembali ke form input. Jika valid, sistem menampilkan halaman utama berupa *dashboard* Nasabah.
- Monitoring Akun dan Informasi Saldo: Melalui menu utama, Nasabah dapat memilih tiga opsi pemantauan data. Jika memilih menu *Lihat Saldo Tabungan*, sistem akan merespons dengan menampilkan saldo tabungan milik nasabah hingga status saldo tampil. Jika memilih menu *Lihat Profil*, sistem akan memicu respons untuk menampilkan data profil pribadi nasabah. Nasabah juga dapat memilih menu *Lihat Riwayat Setoran* untuk menampilkan data riwayat transaksi setoran sampah secara berkala.
- Transaksi Finansial dan Katalog Harga: Nasabah dapat memilih menu *Melakukan Penarikan Saldo*, di mana sistem akan merespons dengan memproses status menunggu persetujuan dari Admin. Selain

itu, Nasabah dapat mengakses informasi nilai ekonomi sampah terkini melalui menu *Lihat Katalog Harga Sampah Terkini*, yang direspons sistem dengan menampilkan katalog harga terupdate.

- Terminasi Sesi: Seluruh rangkaian aktivitas ditutup secara aman ketika Nasabah menekan opsi *logout*, sehingga sistem akan merespons dengan mengakhiri sesi nasabah secara otomatis.

3. Entity Relationship Diagram (ERD)



Gambar 5. ERD
Sumber: Christian & Andre (2026)

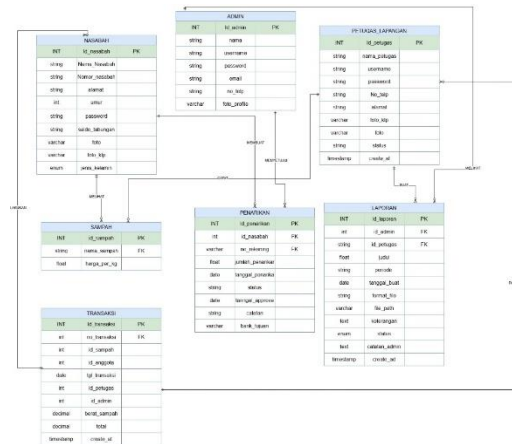
Berdasarkan Gambar 5, rancangan basis data sistem E-MONSTER terdiri dari 6 entitas utama yang saling terintegrasi dengan hubungan relasi sebagai berikut:

- Entitas Pengguna: Terdiri dari entitas ADMIN (kunci *id_admin*), PETUGAS (kunci *id_petugas*), dan NASABAH (kunci *id_nasabah*) yang menyimpan seluruh data profil dan hak akses pengguna sistem.
- Entitas Operasional: Terdiri dari entitas SAMPAH (*id_sampah*) untuk master harga komoditas, TRANSAKSI (*id_transaksi*) untuk mencatat setoran sampah harian, PENARIKAN (*id_penarikan*) untuk pengajuan dana nasabah, dan LAPORAN (*id_laporan*) untuk rekapitulasi data.
- Kardinalitas Relasi (*One-to-Many*): Satu Nasabah dapat melakukan banyak transaksi setoran sampah dan mengajukan banyak penarikan saldo. Satu jenis Sampah dapat digunakan di banyak transaksi yang dicatat oleh Petugas. Sementara itu, Admin dan Petugas memiliki wewenang untuk

membuat serta mengelola banyak dokumen Laporan berkala melalui sistem.

dokumentasi dan rekapitulasi laporan berkala.

4. Logical Record Structure (LRS)

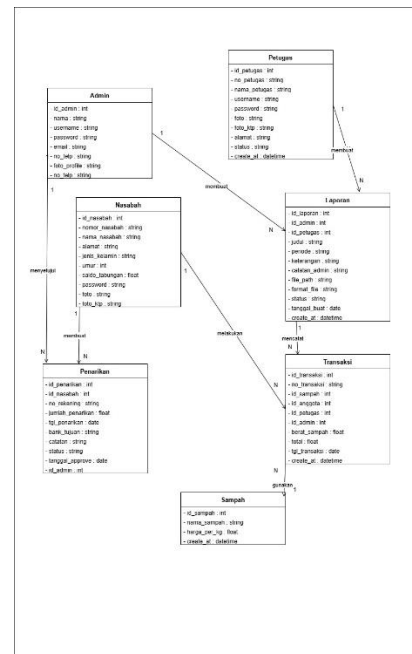


Gambar 6.LRS

Sumber: Christian & Andre (2026)
Berdasarkan Gambar 6, struktur relasi antartabel pada *database* E-MONSTER dijabarkan sebagai berikut:

- Tabel Master Pengguna: Terdiri dari tabel ADMIN (PK:id_admin),PETUGAS_LAPANGAN (PK: id_petugas), dan NASABAH (PK: id_nasabah). Ketiga tabel ini berdiri sendiri sebagai master data entitas pengguna sistem.
- Tabel Master Komoditas: Tabel SAMPAH menggunakan id_sampah sebagai *Primary Key* (PK) untuk menyimpan data jenis dan harga sampah per kilogram.
- Tabel Transaksi dan Relasi *Foreign Key*:
 - Tabel TRANSAKSI (PK: id_transaksi) terelasi dengan tabel NASABAH, SAMPAH, PETUGAS_LAPANGAN, dan ADMIN melalui *Foreign Key* (FK) yang tertanam di dalamnya untuk mencatat setoran.
 - Tabel PENARIKAN (PK: id_penarikan) memuat FK id_nasabah yang menghubungkan riwayat pengajuan saldo nasabah serta terelasi dengan tindakan persetujuan oleh ADMIN.
 - Tabel LAPORAN (PK: id_laporan) memuat FK id_admin dan id_petugas untuk mencatat aktor yang melakukan pembuatan

5. Class Model/Class Diagram



Gambar 7.Class Model/Class Diagram
Sumber: Christian & Andre (2026)

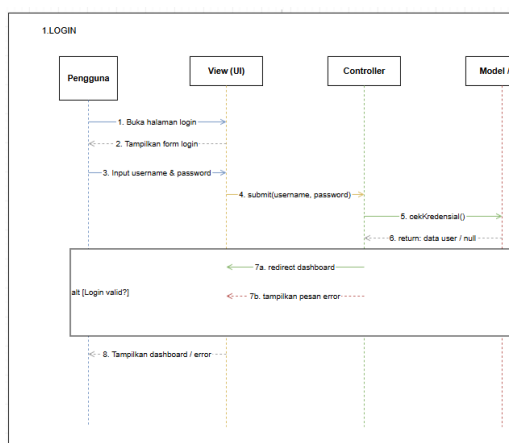
Berdasarkan Gambar 7, struktur kelas (*class*) beserta hubungan multiplisitas objek pada arsitektur sistem E-MONSTER dijabarkan sebagai berikut:

- Komponen Kelas Pengguna: Terdiri dari kelas Admin, Petugas, dan Nasabah. Ketiga kelas ini merangkum atribut data pribadi, kredensial akun (*username* dan *password*), serta status hak akses masing-masing aktor di dalam aplikasi.
- Komponen Kelas Operasional: Terdiri dari kelas Sampah (mengelola data komoditas bank sampah), Transaksi (mencatat rekapitulasi setoran sampah harian), Penarikan (mengelola pengajuan saldo), dan Laporan (memproses dokumentasi rekapitulasi periodik).
- Multiplisitas Hubungan Antar-Kelas:
 - Satu (1) objek Nasabah dapat berinteraksi dengan banyak (N) objek Transaksi setoran sampah dan memicu banyak (N) objek Penarikan saldo.

- 2) Banyak (N) objek Transaksi terikat dengan satu (1) kelas Sampah yang digunakan sebagai acuan nilai harga per kilogram.
- 3) Satu (1) objek Admin dapat memvalidasi banyak (N) objek Penarikan saldo nasabah, serta bersama satu (1) objek Petugas memiliki kemampuan untuk membuat banyak (N) objek Laporan harian maupun bulanan pada sistem.

- d. Percabangan Kondisi (*Alternative Loop*):
 - 1) Jika *login* valid (data *user* ditemukan), *Controller* melakukan *redirect dashboard* ke *View (UI)*, lalu memunculkan tampilan *dashboard* utama kepada Pengguna.
 - 2) Jika *login* tidak valid (data *user* bernilai *null*), *Controller* mengirimkan instruksi untuk menampilkan pesan *error* pada *View (UI)* agar dapat dilihat oleh Pengguna.

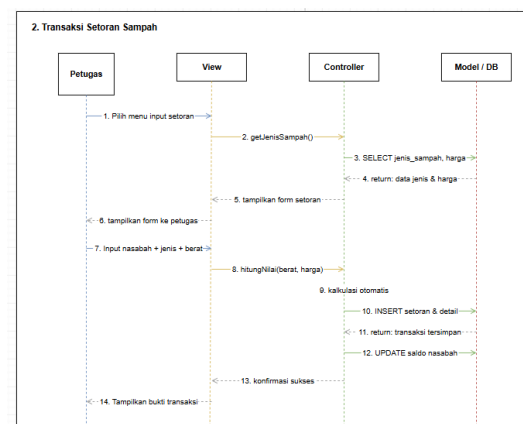
6. Sequence Diagram



Gambar 8. Sequence Diagram Login
Sumber: Christian & Andre (2026)

Berdasarkan Gambar 8, urutan interaksi antar-komponen dalam proses autentikasi sistem E-MONSTER dijabarkan sebagai berikut:

- a. Proses Awal: Pengguna membuka halaman *login* pada komponen *View (UI)*, kemudian sistem memberikan respons dengan menampilkan form *login* kembali kepada Pengguna.
- b. Pengiriman Data: Pengguna melakukan input *username* dan *password* pada *View (UI)*, lalu menekan tombol *submit*. Data tersebut dikirimkan oleh *View (UI)* menuju komponen *Controller*.
- c. Validasi Basis Data: *Controller* meneruskan data dengan memanggil fungsi *cekKredensial()* pada komponen *Model / DB*. Komponen *Model / DB* kemudian memberikan nilai balik (*return*) berupa data *user* jika ditemukan atau *null* jika tidak ditemukan.

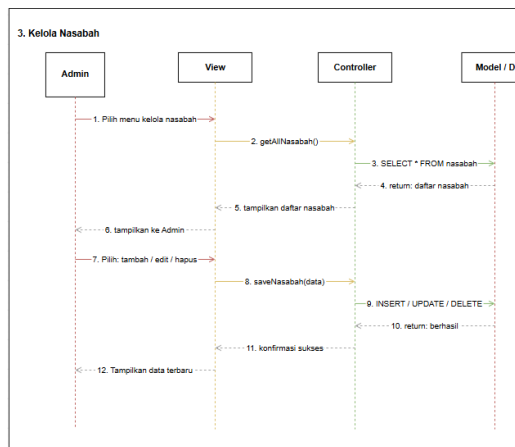


Gambar 9. Sequence Diagram Transaksi Setoran Sampah

Sumber: Christian & Andre (2026)

Berdasarkan Gambar 9, urutan interaksi komponen dalam memproses setoran sampah dijabarkan sebagai berikut:

- a. Tahap Awal: Petugas mengakses menu input setoran pada *View*, lalu *Controller* mengambil data jenis dan harga sampah dari *Model / DB* untuk menampilkan form transaksi kepada Petugas.
- b. Proses Kalkulasi: Petugas menginput data nasabah, jenis, dan berat sampah pada *View*. Data dikirim ke *Controller* untuk dilakukan kalkulasi total nominal secara otomatis.
- c. Penyimpanan & Output: *Controller* menyimpan data transaksi ke *Model / DB*, melakukan pembaruan saldo pada data nasabah terkait, dan mengirimkan konfirmasi sukses ke *View* untuk menampilkan bukti transaksi kepada Petugas.

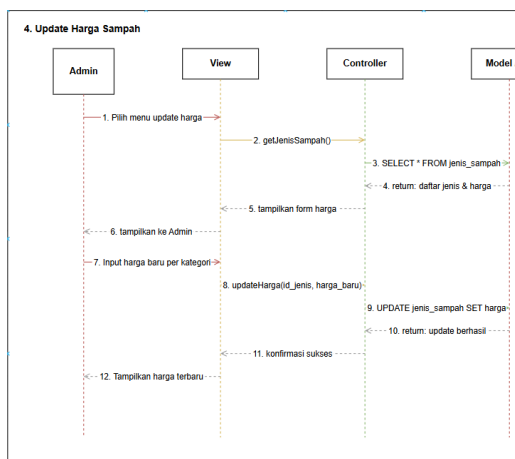


Gambar 10. *Sequence Diagram* Kelola Nasabah

Sumber: Christian & Andre (2026)

Berdasarkan Gambar 10, urutan interaksi komponen dalam proses pengelolaan data nasabah dijabarkan sebagai berikut:

- Menampilkan Data:** Admin mengakses menu kelola nasabah pada *View*, lalu *Controller* memanggil fungsi `getAllNasabah()` untuk mengambil daftar nasabah dari *Model / DB* dan menampilkannya kepada Admin.
- Proses Eksekusi (CRUD):** Admin memilih aksi tambah, edit, atau hapus data pada *View*. Perubahan data tersebut diteruskan ke *Controller* melalui fungsi `saveNasabah(data)`.
- Penyimpanan & Output:** *Controller* mengeksekusi perintah manipulasi data (*INSERT/UPDATE/DELETE*) ke *Model / DB*, menerima nilai balik sukses, dan mengirimkan konfirmasi ke *View* untuk menampilkan data nasabah terbaru kepada Admin.

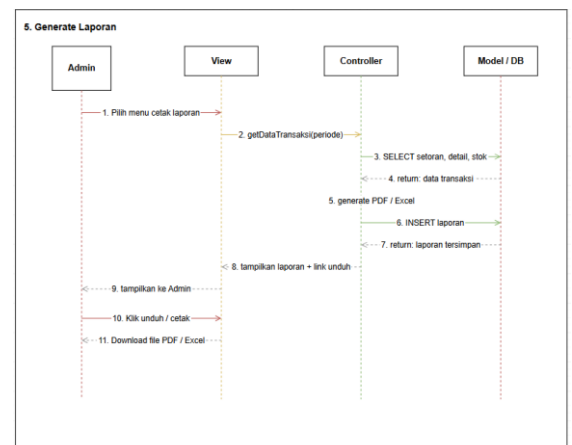


Gambar 11. *Sequence Diagram* Update Harga Sampah

Sumber: Christian & Andre (2026)

Berdasarkan Gambar 11, urutan interaksi komponen dalam proses pembaruan harga sampah oleh Admin dijabarkan sebagai berikut:

- Menampilkan Data Awal:** Admin mengakses menu *update* harga pada *View*, lalu *Controller* memanggil fungsi `getJenisSampah()` untuk mengambil daftar jenis dan harga sampah dari *Model / DB* guna menampilkan form harga kepada Admin.
- Proses Pembaruan:** Admin menginput harga baru per kategori pada *View*. Data tersebut diteruskan ke *Controller* melalui pemanggilan fungsi `updateHarga(id_jenis, harga_baru)`.
- Penyimpanan & Output:** *Controller* mengeksekusi perintah kueri pembaruan data (*UPDATE*) ke *Model / DB*, menerima nilai balik sukses, dan meneruskan konfirmasi ke *View* untuk menampilkan data harga terbaru kepada Admin.



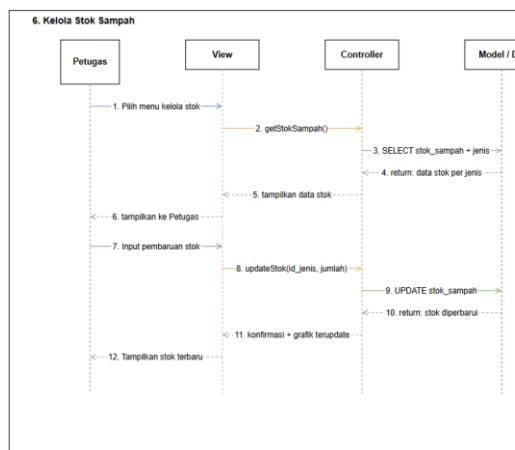
Gambar 12. *Sequence Diagram* Generate Laporan

Sumber: Christian & Andre (2026)

Berdasarkan Gambar 12, urutan interaksi komponen dalam proses pembuatan laporan oleh Admin dijabarkan sebagai berikut:

- Pencarian Data:** Admin memilih menu cetak laporan pada *View*. Komponen *View* mengirim parameter ke *Controller* melalui fungsi `getDataTransaksi(periode)` untuk mengambil rekapitulasi data setoran, detail, dan stok dari *Model / DB*.

- b. Proses Generate & Simpan: Setelah menerima data transaksi, *Controller* memproses *generate* dokumen ke format PDF/Excel, lalu mengeksekusi perintah *INSERT* untuk menyimpan rekam log laporan tersebut ke dalam *Model / DB*.
- c. Unduh File: *Controller* mengirimkan tampilan laporan beserta tautan unduh ke *View* untuk disajikan kepada Admin. Proses selesai saat Admin menekan tombol unduh/cetak dan menerima berkas fisik PDF/Excel yang diinginkan.



Gambar 13. Sequence Diagram Kelola Stok Sampah

Sumber: Christian & Andre (2026)

Berdasarkan Gambar 13, urutan interaksi komponen dalam proses pengelolaan stok sampah oleh Petugas dijabarkan sebagai berikut:

- a. Menampilkan Data Awal: Petugas mengakses menu kelola stok pada *View*, lalu *Controller* memanggil fungsi *getStokSampah()* untuk mengambil data stok per jenis dari *Model / DB* guna menampilkan data tersebut kepada Petugas.
- b. Proses Pembaruan: Petugas memasukkan input pembaruan stok pada *View*. Perubahan data tersebut diteruskan ke *Controller* melalui pemanggilan fungsi *updateStok(id_jenis, jumlah)*.
- c. Penyimpanan & Output: *Controller* mengeksekusi perintah kueri pembaruan (*UPDATE*) ke *Model / DB*, menerima nilai balik sukses, dan meneruskan konfirmasi beserta grafik ter-update ke *View* untuk menampilkan informasi stok terbaru kepada Petugas.

III. HASIL DAN PEMBAHASAN

3.1 Implementasi Antarmuka Sistem

Hasil dari penelitian ini adalah sebuah sistem informasi berbasis web terintegrasi bernama E-MONSTER. Pembahasan implementasi sistem difokuskan pada pemenuhan kebutuhan fungsional dari tiga aktor utama melalui arsitektur *Model-View-Controller* (MVC). Integrasi sistem dibuktikan melalui keterhubungan komponen MVC pada modul transaksi. Proses dimulai dari antarmuka pengguna (*View*) di mana petugas menginput parameter berat sampah. Data tersebut kemudian diteruskan ke komponen pengendali (*Controller*) untuk dikalkulasikan secara otomatis berdasarkan harga pasar terkini, dan tahap akhir dilakukan penyimpanan ke *database* melalui objek data (*Model*) ke dalam tabel transaksi (*tbl_transaksi*).

Implementasi sistem juga mencakup penyediaan *dashboard* khusus untuk memfasilitasi fungsionalitas Admin dalam menyetujui penarikan dana nasabah secara transparan, serta menu monitoring untuk memantau fluktuasi grafik ketersediaan stok sampah anorganik di gudang penyimpanan secara *real-time*.

3.2 Pengujian Sistem (Black-Box Testing)

Untuk memastikan bahwa seluruh modul dan fungsionalitas aplikasi E-MONSTER telah berhasil diimplementasikan tanpa adanya kesalahan teknis (*error*) pada antarmuka, dilakukan pengujian sistem menggunakan metode *Black-Box Testing*. Pengujian ini difokuskan pada pengamatan hasil eksekusi data masukan (*input*) dan kesesuaian luaran (*output*) pada setiap fitur utama.

Berdasarkan hasil uji coba fungsional yang telah dilaksanakan, rincian hasil pengujian *Black-Box Testing* dipaparkan pada Tabel 1.

Tabel 1 Blackbox Testing

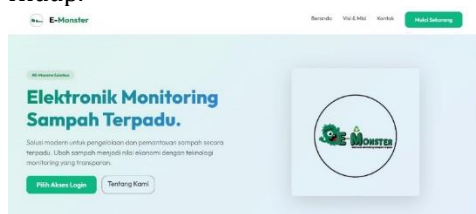
N O	Fitur/Fungs i Sistem	Skenario Pengujian	Hasil yang diharapkan	Hasil pengujian	Statu s
--------	-------------------------	-----------------------	--------------------------	--------------------	------------

1	Autentikasi Pengguna (Login)	Menginput <i>username</i> & <i>password</i> yang valid.	Sistem mengarahkan pengguna ke halaman <i>dashboard</i> .	Pengguna berhasil masuk ke halaman <i>dashboard</i> .	Valid
2	Input Transaksi Setoran	Petugas memasukkan data nasabah, jenis sampah, dan berat sampah.	Sistem otomatis menghitung total rupiah dan memperbarui saldo nasabah.	Total nominal terhitung otomatis dan saldo nasabah langsung bertambah.	Valid
3	Kelola Data Nasabah	Admin menambah, mengedit, atau menghapus data profil nasabah.	Data nasabah pada basis data berhasil dimanipulasi (<i>CRUD</i>).	Data nasabah berhasil tersimpan, terupdate, dan terhapus di <i>database</i> .	Valid
4	Update Harga Sampah	Admin mengubah tarif konversi rupiah per kilogram dari sampah anorganik.	Parameter harga sampah terbaru terupdate di sistem secara global.	Katalog harga pada akun nasabah langsung berubah sesuai <i>input</i> terbaru.	Valid
5	Generate Laporan	Admin memilih periode tanggal dan mencetak rekap laporan.	Sistem memproses file dan memunculkan tombol unduh PDF/Excel.	File laporan berhasil digenerate dan dapat diunduh tanpa kerusakan format.	Valid
6	Kelola Stok Sampah	Petugas memperbarui data jumlah masuk/keluar stok sampah di gudang.	Data kuantitas stok terupdate dan grafik <i>dashboard</i> berubah.	Jumlah stok berubah di <i>database</i> dan visualisasi grafik terupdate tepat.	Valid

Sumber: Christian & Andre (2026)

Berdasarkan hasil uji coba fungsional, fitur-fitur utama telah berhasil diimplementasikan dengan rincian sebagai berikut:

1. *Halaman Homepage & Dashboard Admin*: Menampilkan grafik volume sampah secara *real-time* untuk kebutuhan monitoring pihak Dinas Lingkungan Hidup.

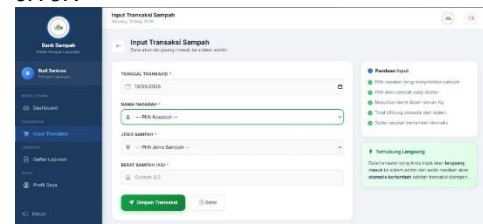


Sumber: Christian & Andre (2026)

Gambar 14. *Homepage* E-Monster

2. *Modul Transaksi Petugas*: Memfasilitasi pencatatan setoran sampah di lapangan secara digital untuk meminimalisir *human*

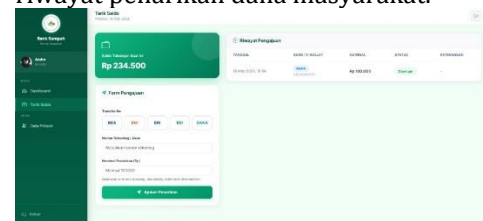
error.



Sumber: Christian & Andre (2026)

Gambar 15. *Transaksi Sampah E-Monster*

3. *Panel Nasabah & Tarik Saldo*: Menyediakan transparansi informasi mengenai akumulasi saldo tabungan dan riwayat penarikan dana masyarakat.



Sumber: Christian & Andre (2026)

Gambar 16. *Dashboard Tarik Saldo E-Monster*

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, penelitian ini berhasil membangun sistem informasi E-MONSTER yang mendigitalisasi seluruh alur administrasi bank sampah anorganik pada Dinas Perumahan Rakyat dan Kawasan Permukiman, Pertanahan dan Lingkungan Hidup Kabupaten Bengkayang menjadi lebih terintegrasi. Hasil pengujian fungsional menggunakan metode *Black-Box Testing* menunjukkan tingkat keberhasilan mencapai 100% dengan status valid, yang membuktikan bahwa sistem ini mampu menyajikan transparansi saldo tabungan nasabah secara tepat, mengotomatisasi kalkulasi nominal setoran secara akurat, serta mempermudah pihak instansi dalam memonitor perkembangan volume sampah secara *real-time*. Untuk pengembangan penelitian selanjutnya, disarankan agar platform ini dikembangkan ke dalam versi *mobile application* (Android/iOS) guna memperluas aksesibilitas nasabah serta diintegrasikan dengan fitur *payment gateway* untuk mempermudah proses pencairan saldo secara otomatis.

V. REFERENSI

Ahmad Herdinal Muttaqin, Adian fatchur rochim, Eko Didik Widiyanto. 2022. "Sistem

- Auntentikasi Hotspot Menggunakan LDAP Dan Radius Pada Jaringan Internet Wireless Prodi Teknik Sistem Komputer.” *Jurnal Teknologi Dan Sistem Komputer* Vol 4,no 2.
- Atin, S., S. Mutia², A. Widayanti³, H. S. Yatawa⁴, A. A. Rafdhi⁵, and I Afrianto⁶. 2022. “PERANCANGAN SISTEM INFORMASI BANK SAMPAH BERBASIS WEBSITE.”
- Dewi, Indah, Murti Suyoto, Surya Wedi, and Kurnia Setiawan. 2023. “MENGUNAKAN TOGAF ADM (STUDI KASUS KANTOR PERTANAHAN ABC) INFORMATION SYSTEM AND INFORMATION TECHNOLOGY ARCHITECTURE DESIGN (ABC LAND OFFICE CASE STUDY).” 10(4):909-18. doi: 10.25126/jtik.2023106220.
- Erwin, Ni'matullah Muin, Marni, Anugrah Batari Gading Salsabilah⁴. 2025. “RANCANG BANGUN E-COMMERCE BERBASIS WEB UNTUK MENDORONG PARTISIPASI MASYARAKAT DALAM PENGELOLAAN SAMPAH BERKELANJUTAN.” 10(2):498-506.
- Ni Made Mila Rosa Desmayani , Luh Gede Bevi Libraeni , S.E., M.Si, Aniek Suryanti Kusuma. 2024. “Pkm: Implementasi Sistem Informasi Bank Sampah Banjarangkan Asri.” 4(2):152-59.
- Nicho Yuda Mahendra, Lukman Nulhakim, Andri Irawan, DedyPrasetya Kristiadi. 2024. “PENERAPAN SISTEM INFORMASI PENGELOLAAN BANK SAMPAH BERBASIS WEB PADA STMIK KUWERA.” IV(02):36-41.
- Prabowo, Rizky. 2022. “Jurnal Pepadun Aplikasi Multi Bank Sampah Berbasis Web Jurnal Pepadun.” 3(1):64-73.
- Putra, Yeviki Maisya. 2025. “IMPLEMENTASISISTEM INFORMASI TRANSPARANSI DAN EFISIENSI PENGELOLAAN PENDATAAN SAMPAH DI PROVINSI SUMATERA BARAT (STUDI KASUS DINAS LINGKUNGAN HIDUP).” 4(1):493-503.
- Rahmah, Fauzia Hayati. 2022. “Aplikasi Bank Sampah Berkah Melimpah Berbasis Website Pada Kelurahan Nanggewer 1 Pendahuluan JURNAL INFORMATIK Edisi Ke-18 , Nomor 2 , Agustus 2022 2 Studi Literatur.” 4221:131-42.
- Zumrotuz Zakiyah, Tri Listyorini, Endang Supriyati¹. 2024. “Implementasi Aplikasi Basamka Di Bank Sampah Mugi Barokah Desa Getassrabi Kudus 1.” 9(April):39-49.