

Implementasi *Application Programming Interface Live Location* dan *Payment Gateway* Aplikasi Bank Sampah Berbasis *Mobile*

Wahyutama Fitri Hidayat, Ahmad Setiadi², Yesni Malau³

Universitas Bina Sarana Informatika^{1,2,3}

wahyutama.wfh@bsi.ac.id¹, ahmad.ams@bsi.ac.id², yesni.ymu@bsi.ac.id³

Diterima (02-03-2026)	Direvisi (31-03-2026)	Disetujui (10-04-2026)
--------------------------	--------------------------	---------------------------

Abstrak - Permasalahan pengelolaan sampah tetap menjadi isu strategis di Indonesia seiring dengan meningkatnya volume sampah rumah tangga yang belum diimbangi oleh sistem layanan yang efisien dan transparan. Perkembangan teknologi *mobile* serta layanan berbasis *Application Programming Interface (API)* mendorong transformasi digital berbagai kegiatan publik, tidak terkecuali bank sampah. Penelitian memiliki tujuan untuk merancang dan mengimplementasikan aplikasi bank sampah dengan aplikasi bernama GoTrash dengan fitur proses pemesanan penjemputan sampah, pelacakan lokasi, pengelolaan transaksi. Aplikasi dikembangkan menggunakan arsitektur *RESTful API* untuk menghubungkan aplikasi *mobile* dengan server *backend* sehingga pertukaran data dapat dilakukan secara efisien, aman, dan *real time*. Integrasi layanan pendukung seperti pemetaan lokasi, Firebase untuk sinkronisasi data dan notifikasi, serta Midtrans sebagai *payment gateway* digunakan untuk meningkatkan fungsionalitas aplikasi. Tahapan pengembangan menggunakan metode *waterfall* diawali dengan analisa kebutuhan, perancangan *wireframe* antarmuka, implementasi fitur, kemudian dilanjutkan dengan pengujian menggunakan metode *black-box testing* untuk memastikan kesesuaian fungsi dengan kebutuhan pengguna. Dengan demikian, aplikasi yang dikembangkan mampu meningkatkan efektivitas, transparansi, dan kemudahan layanan bank sampah digital serta berpotensi menjadi solusi pengelolaan sampah yang lebih modern, terstruktur, dan *skalabel*.

Kata Kunci : Aplikasi, Bank Sampah, API, *Waterfall*, Black-Box

Abstract - *Waste management problems remain a strategic issue in Indonesia, along with the increasing volume of household waste that has not been matched by efficient and transparent service systems. The development of mobile technology and services based on Application Programming Interfaces (API) has encouraged the digital transformation of various public activities, including waste banks. This study aims to design and implement a waste bank application named GoTrash with features for scheduling waste pickup, location tracking, and transaction management. The application is developed using a RESTful API architecture to connect the mobile application with the backend server so that data exchange can be carried out efficiently, securely, and in real time. The integration of supporting services such as location mapping, Firebase for data synchronization and notifications, and Midtrans as a payment gateway is utilized to enhance the functionality of the application. The development stages use the waterfall method, beginning with requirements analysis, interface wireframe design, feature implementation, and followed by testing using the black-box testing method to ensure functional conformity with user needs. Therefore, the developed application is able to improve the effectiveness, transparency, and convenience of digital waste bank services and has the potential to become a more modern, structured, and scalable waste management solution..*

Keywords: *Application, Waste Bank, API, Waterfall, Black-Box*

I. PENDAHULUAN

Permasalahan pengelolaan sampah masih menjadi isu strategis di Indonesia seiring meningkatnya volume sampah rumah tangga yang belum diimbangi dengan pendekatan layanan yang efisien dan transparan. Konsep bank sampah sebagai sistem pengelolaan berbasis partisipasi masyarakat telah banyak

dikemukakan dan diterapkan sebagai upaya ekonomi berkelanjutan untuk mengurangi limbah sekaligus memberikan nilai tambah ekonomi (Hs et al., 2025). Penelitian mutakhir menunjukkan bahwa dengan adanya aplikasi *mobile* pada bank sampah dapat memperluas jangkauan layanan dan meningkatkan keterlibatan pengguna, namun masih terdapat

gap dalam integrasi modul pemetaan dan transaksi yang komprehensif dalam satu platform terpadu (Latif et al., 2024). Namun permasalahan muncul ketika kekurangan sistem yang difungsikan untuk membantu

Kegiatan kerja sehingga dapat mengakibatkan keterlambatan produktivitas suatu perusahaan atau organisasi (Hidayat et al., 2024). *RESTful API* memiliki berbagai kegunaan yang sangat penting dalam pengembangan aplikasi (Ponggawa et al., 2024). Perkembangan teknologi arsitektur *RESTful API* menjadi fondasi penting dalam pengembangan aplikasi *mobile* modern karena kemampuannya dalam mendukung pertukaran data yang konsisten, ringan, dan *interoperable* antara klien dan server (Calista Chandra et al., 2024). Berbagai penelitian terbaru menggarisbawahi dengan digunakannya *RESTful API* untuk peningkatan hubungan antara front-end dan sistem *backend*, termasuk notifikasi *real time* dan sinkronisasi data lintas platform (Fadhil & Sutarman, 2024). Implementasi arsitektur ini sejalan untuk aplikasi bank sampah karena dimungkinkan pemrosesan data transaksi, *autentikasi* pengguna, dan integrasi layanan eksternal secara terstruktur.

Aspek spasial dan pemetaan menjadi komponen penting dalam layanan bank sampah berbasis *mobile* sebagai upaya mendukung penentuan lokasi bank sampah, perencanaan rute penjemputan, dan visualisasi posisi *real time*. Integrasi *Maps API* terbukti efektif dalam banyak aplikasi layanan publik dan logistik, termasuk sistem yang berkaitan dengan rute terpendek dan pelacakan lokasi pengguna (Susanto & Hariyanto, 2023). Penelitian ini menunjukkan bahwa penggunaan peta digital meningkatkan keakuratan penjadwalan layanan dan pengalaman pengguna secara keseluruhan, namun masih jarang dikombinasikan secara terpadu dengan modul transaksi dalam aplikasi bank sampah.

Di samping itu, kebutuhan akan sistem transaksi pembayaran yang aman, cepat, dan transparan semakin mendesak dalam aplikasi *mobile* layanan publik. Integrasi *payment gateway* seperti Midtrans memungkinkan otomatisasi verifikasi pembayaran, penyimpanan riwayat transaksi, dan notifikasi status transaksi secara *real time* (Fauzi & Fachrie, 2024). Hal ini berkontribusi terhadap kepercayaan pengguna dan ketepatan pencatatan finansial pada aplikasi, dibandingkan pencatatan manual yang rentan kesalahan. Penelitian lain juga menyoroiti pentingnya dukungan *real-time database* untuk notifikasi status layanan dan pembaruan data

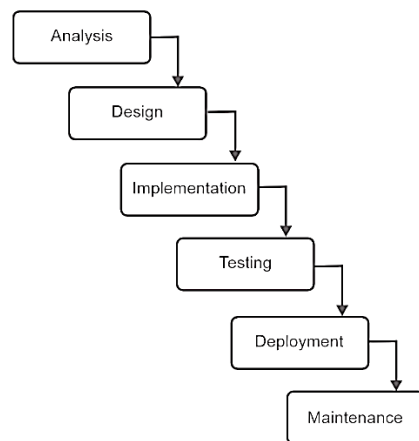
secara instan kepada pengguna (Fadhil & Sutarman, 2024).

Beberapa studi terkait implementasi API dalam layanan *mobile* menunjukkan bagaimana *RESTful API* serta *cloud services* dapat digunakan untuk mengelola berbagai modul layanan, termasuk pemetaan, notifikasi, dan transaksi pembayaran, secara lebih optimal (Bagus Syafiq Faqihuddin et al., 2025). Pendekatan semacam ini memungkinkan sistem aplikasi untuk menjadi lebih modular, skalabel, dan mudah dipelihara dalam jangka panjang daripada sistem monolitik atau terpisah-pisah. Berdasarkan gap riset tersebut, penelitian ini mengusulkan perancangan dan implementasi API peta dan transaksi pembayaran dalam aplikasi bank sampah berbasis *mobile* bernama GoTrash, dikembangkan menggunakan arsitektur *RESTful API* dengan format JSON serta terintegrasi dengan *Google Maps API*, *Midtrans*, *Firebase*, dan *Flutter*. Pendekatan ini dirancang untuk mengatasi keterbatasan sistem konvensional dan digital sebelumnya dengan menyediakan satu platform terpadu yang mampu mengelola data lokasi, transaksi, dan status penjemputan secara sinkron dan *real time* (Fauzi & Fachrie, 2024).

Adapun tujuan utama penelitian ini adalah untuk meningkatkan efisiensi proses transaksi melalui integrasi *payment gateway*, menggabungkan pemetaan lokasi secara digital, memastikan transparansi pembayaran, dan menyediakan pemantauan *real time* terhadap penjemputan sampah dan transaksi keuangan melalui *backend* yang terintegrasi. Diharapkan bahwa penelitian ini memberikan kontribusi konseptual dan praktik dalam pengembangan arsitektur API terintegrasi untuk aplikasi layanan publik berbasis *mobile* dan menjadi acuan untuk penelitian selanjutnya di bidang pengelolaan sampah digital.

II. METODOLOGI PENELITIAN

Berdasarkan uraian permasalahan, maka digunakan metode pengembangan perangkat lunak *waterfall* (air terjun). Dengan menggunakan model *waterfall* yang dikemukakan oleh (Fitria et al., 2020) yang terbagi menjadi enam tahapan, yaitu :



Sumber : Fitria et al (2020)

Gambar 1. Model *Waterfall*

1. *Analysis*

Pengumpulan kebutuhan secara menyeluruh dilanjutkan analisa dan mendefinisikan kebutuhan yang akan dipenuhi oleh aplikasi yang akan dibangun. Proses ini mendefinisikan secara rinci mengenai fungsi-fungsi, batasan dan tujuan dari perangkat lunak (Oktaviana Pamudji & Kamisutara, 2024).

2. *Design*

Tahap *design* dibuat dengan tujuan menghasilkan gambaran aplikasi yang detail. Tahapan ini merupakan tahap perancangan sistem. Sistem yang akan dibuat harus sesuai dengan kebutuhan (Putri et al., 2023).

3. *Implementation*

Tahap ini dilakukan implementasi ke dalam kode program setelah menyelesaikan tahap analisis dan desain sistem (Lina Nurlaili & Mustika Rizki, 2025). Tujuan tahapan ini mengubah desain menjadi kode program yang integrasikan menjadi aplikasi yang lengkap.

4. *Testing*

Di tahap ini dilakukan pengujian aplikasi yang sudah selesai dan dilakukan pengujian menggunakan metode *blackbox* testing dengan tujuan untuk uji secara keseluruhan fungsinya apakah berjalan dengan baik atau tidak (Ghani et al., 2023).

5. *Deployment*

Pengguna menguji apakah sistem tersebut telah sesuai dengan yang disetujui. Perangkat lunak yang telah digunakan oleh pengguna suatu saat bisa saja mengalami perubahan

(Septiawan et al., 2022).

6. *Maintenance*

Instalasi dan proses perbaikan sistem sesuai yang disetujui. Pada tahap ini, sistem dipantau secara terus-menerus untuk memastikan kerjanya tetap optimal (Satria et al., 2024).

III. HASIL DAN PEMBAHASAN

Pada bagian ini, dijabarkan hasil penelitian dan pembahasan yang disajikan dengan angka, tabel, grafik, dan bentuk lain yang dapat dipahami dengan mudah.

1. *Analysis*

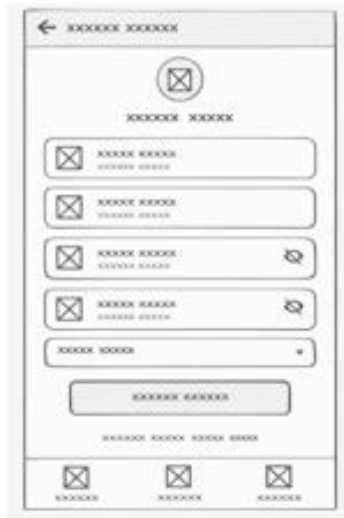
Hasil analisa kebutuhan menghasilkan, di mana analisa disesuaikan dengan kebutuhan pengguna dibagi berdasarkan pengguna yang terdiri dari dua pengguna yaitu pengguna dan petugas. Pengguna dapat melakukan *registrasi* dan *login*, pemesanan pengangkutan sampah, penentuan lokasi penjemputan, penjadwalan waktu pengambilan, pemantauan status pesanan, riwayat dan estimasi biaya. Sedangkan untuk petugas dapat melakukan pilihan menerima atau menolak pesanan, akses petunjuk arah pengguna, *update* status pengangkutan.

2. *Design*

Desain aplikasi didasarkan pada hasil analisis kebutuhan pada tahap sebelumnya sehingga dihasilkan alur kerja sistem sebagai berikut:

- Pengguna dapat mendaftar atau masuk ke aplikasi.
- Pengguna membuat pengangkutan sampah.
- Sistem mencari petugas terdekat.
- Petugas menerima pesanan.
- Sampah diambil di lokasi pengguna
- Petugas melakukan perhitungan estimasi harga dan melakukan perubahan status pesanan.
- Petugas melakukan pengantaran ke TPS dan menimbang berat pasti.
- Sistem melakukan kalkulasi berat dan harga.
- Pengguna membayar melalui sistem pembayaran yang sudah terintegrasi.
- Selesai.

Berdasarkan kebutuhan tersebut maka digambarkan rancangan antar muka menggunakan desain *wireframe* berikut:



Sumber : Penulis (2026)

Gambar 2. Wireframe Registrasi

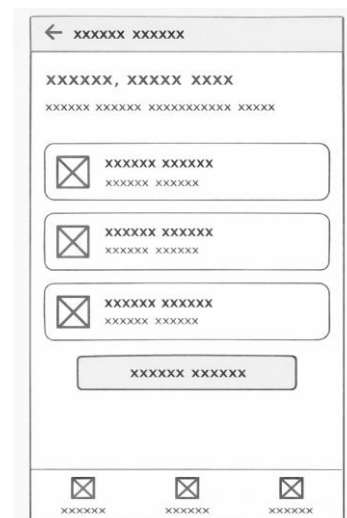
Gambar 2 merupakan halaman formulir pendaftaran akun baru pada aplikasi *mobile*. Bagian atas berisi *header* dengan tombol kembali dan judul halaman. Di bawahnya terdapat ikon profil sebagai ilustrasi pengguna, diikuti beberapa *field input* yang disusun vertikal untuk mengisi data seperti identitas, kontak, kata sandi, serta pilihan peran melalui *dropdown*. Setiap *field* dilengkapi ikon *placeholder* di sisi kiri. Pada bagian bawah tersedia tombol aksi utama untuk mengirimkan pendaftaran serta teks informasi tambahan. Navigasi bawah tetap ditampilkan untuk perpindahan halaman. *Layout* dibuat sederhana, terstruktur, dan fokus pada kemudahan pengisian *form*.



Sumber : Penulis (2026)

Gambar 3. Wireframe Login

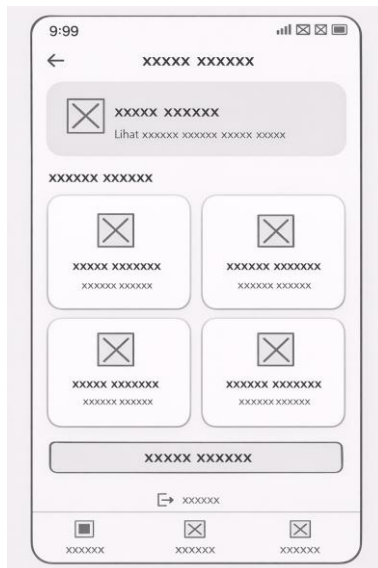
Sedangkan Gambar 3. merupakan *wireframe* halaman *login* aplikasi *mobile* dengan tata letak terpusat berbentuk kartu. Bagian atas menampilkan logo sebagai identitas aplikasi, diikuti judul dan teks instruksi. Di bawahnya terdapat dua *field input* untuk email dan kata sandi yang masing-masing dilengkapi ikon *placeholder*. Selanjutnya tersedia tombol aksi utama untuk masuk, serta teks navigasi tambahan di bagian bawah untuk menuju halaman pendaftaran. Desain *wireframe* dibuat sederhana, fokus, dan memudahkan proses *otentikasi* pengguna.



Sumber : Penulis (2026)

Gambar 4. Wireframe Menu Utama Pengguna

Gambar 4 Tampilan ini merupakan halaman beranda aplikasi *mobile* dengan struktur vertikal dan fokus pada kemudahan navigasi. Bagian atas menampilkan status bar dan *header* dengan tombol kembali serta judul halaman. Di bawahnya terdapat teks sapaan dan deskripsi singkat sebagai pengantar. Konten utama terdiri dari tiga kartu menu yang disusun bertumpuk, masing-masing dilengkapi ikon *placeholder* dan teks untuk mewakili fitur utama. Selanjutnya terdapat satu tombol aksi utama berukuran penuh sebagai *call-to-action*. Pada bagian paling bawah tersedia bilah navigasi dengan tiga ikon untuk perpindahan antar halaman. Desain *wireframe* dibuat sederhana, monokrom, dan menekankan tata letak tanpa detail visual.



Sumber : Penulis (2026)

Gambar 5. Wireframe Halaman Utama Petugas

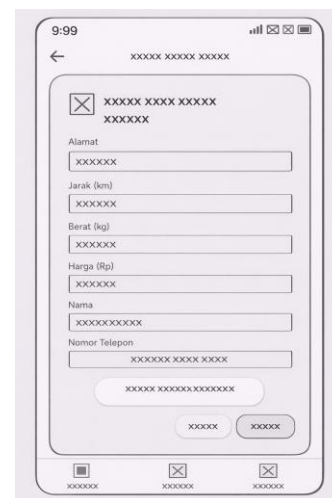
Gambar 5 merupakan halaman *dashboard* utama pengguna dengan struktur ringkas dan berfokus pada akses cepat fitur. Bagian atas menampilkan *header* dengan tombol kembali dan judul halaman. Di bawahnya terdapat satu kartu informasi sebagai ringkasan atau sapaan pengguna. Selanjutnya tersedia bagian menu utama yang disusun dalam *grid* dua kolom berisi empat kartu fitur, masing-masing dilengkapi ikon *placeholder* dan teks. Pada bagian bawah terdapat tombol aksi tambahan serta bilah navigasi untuk perpindahan halaman. *Wireframe* dibuat monokrom dan sederhana untuk menekankan tata letak dan hierarki konten.



Sumber : Penulis (2026)

Gambar 6. Wireframe Pemilihan Lokasi

Gambar 6 *Wireframe* halaman ini menampilkan *layout mobile* sederhana untuk pemilihan lokasi penjemputan. Bagian atas terdapat *header* dengan ikon kembali dan judul bertuliskan "xxxxx xxxxx xxxxxxx". Bagian tengah didominasi area peta berupa blok dan garis abu-abu dengan beberapa *marker* simbol "X" serta satu pin utama sebagai titik lokasi. Di bagian bawah terdapat kartu informasi alamat berisi teks tersamarkan (xxxxx/999999), lalu tombol aksi utama lebar penuh bertuliskan "xxxxx xxxxxxx" untuk konfirmasi lokasi.



Sumber : Penulis (2026)

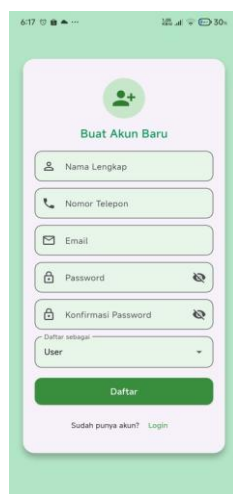
Gambar 7. Wireframe Pemesanan

Gambar 7 *Wireframe* ini menampilkan halaman formulir pembuatan pesanan dalam bentuk modal/kartu tengah layar. Bagian atas terdapat *header* dengan ikon X (logo) dan judul teks tersamarkan "xxxxxx xxxxx xxxxxxx". Di bawahnya tersedia beberapa *field input* berurutan (alamat, jarak, berat, harga, nama, nomor telepon) dengan isi berupa "xxxxxx/999999". Terdapat tombol pilihan tanggal, lalu di bagian bawah dua tombol aksi untuk batal dan simpan. Tata letak fokus pada pengisian data secara vertikal dan sederhana.

3. Implementation

Tahap implementasi sistem dilakukan dengan mentransformasikan rancangan *wireframe* ke dalam bentuk antarmuka pengguna aplikasi *mobile* GoTrash menggunakan *framework* Flutter. *Wireframe* berfungsi sebagai acuan visual dalam menentukan struktur navigasi, hierarki informasi, serta penempatan komponen interaksi sehingga

proses pengembangan antarmuka dapat berlangsung secara sistematis dan konsisten dengan kebutuhan pengguna. Setiap halaman utama, seperti autentikasi, beranda, peta lokasi bank sampah, transaksi tabungan, riwayat pembayaran, dan fitur penjemputan, diimplementasikan menjadi komponen *widget* Flutter dengan pendekatan *single codebase* untuk mendukung kinerja lintas platform. Antarmuka aplikasi terintegrasi dengan *backend* melalui *RESTful* API menggunakan format JSON, sehingga data pengguna, transaksi, dan lokasi dapat diperbarui secara dinamis. Modul peta direalisasikan menggunakan Google Maps API untuk menampilkan lokasi bank sampah dan pelacakan penjemputan secara *real time*, sedangkan modul transaksi terhubung dengan *payment gateway* Midtrans untuk memproses pembayaran secara otomatis dan transparan. Selain itu, Firebase dimanfaatkan untuk notifikasi dan sinkronisasi data secara langsung. Implementasi berbasis *wireframe* ini menghasilkan tampilan aplikasi yang responsif, terstruktur, serta mendukung efisiensi interaksi pengguna dalam mengakses seluruh layanan GoTrash.

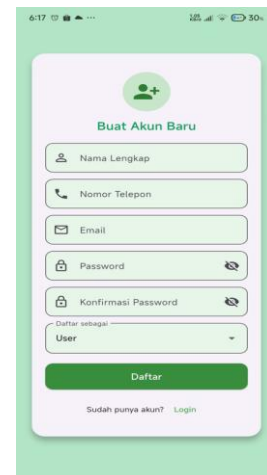


Sumber : Penulis (2026)

Gambar 8. Implementasi Tampilan Register

Gambar 8 merupakan tampilan halaman pendaftaran akun dengan desain modern dan ramah pengguna. *Form* ditempatkan pada kartu terpusat dengan ikon ilustrasi di bagian atas sebagai identitas visual. Tersedia beberapa *field input* seperti nama, lengkap, nomor telepon, email, *password*, konfirmasi *password*, serta pilihan peran pengguna melalui dropdown. Di bagian bawah terdapat tombol utama *Daftar* sebagai aksi *submit* dan tautan kecil

untuk masuk bagi pengguna yang sudah memiliki akun.

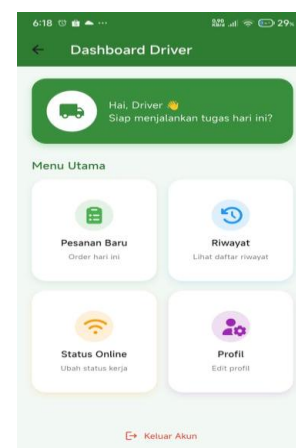


Sumber : Penulis (2026)

Gambar 9. Implementasi Tampilan Login

Sedangkan pada Gambar 9 tampilan *login* tata letak vertikal, rapi, dan memudahkan proses registrasi. Tampilan ini merupakan halaman *login* pengguna dengan desain sederhana dan terpusat. Sebuah kartu *form* berada di tengah layar dengan ikon aplikasi di bagian atas, diikuti teks sambutan dan instruksi singkat. Tersedia dua kolom *input* untuk email dan *password*, dilengkapi ikon pendukung serta tombol tampil/sembunyikan sandi. Di bawahnya terdapat tombol aksi utama *Login* dan tautan *Daftar* bagi pengguna yang belum memiliki akun. Tata letak dibuat ringkas agar proses masuk cepat dan mudah.

Sumber : Penulis (2026)



Gambar 10. Implementasi Tampilan Beranda Pengguna

Gambar 10 merupakan halaman beranda

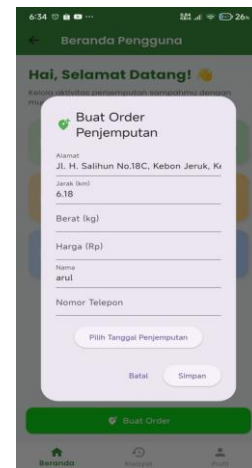
pengguna yang berfungsi sebagai pusat navigasi utama aplikasi. Bagian atas menampilkan *header* dengan judul halaman dan sapaan singkat kepada pengguna. Di bawahnya terdapat tiga kartu menu berwarna yang disusun vertikal, masing-masing mewakili fitur utama seperti jadwal penjemputan, poin dan *reward*, serta edukasi daur ulang. Pada bagian bawah tersedia tombol aksi utama Buat Order untuk melakukan pemesanan, serta *bottom navigation* dengan beberapa ikon untuk berpindah halaman. Desain dibuat bersih, sederhana, dan mudah diakses.



Sumber : Penulis (2026)

Gambar 11. Implementasi Tampilan Titik Lokasi Penjemputan

Gambar 11 merupakan merupakan halaman penentuan lokasi penjemputan yang menampilkan peta sebagai fokus utama layar. Di bagian atas terdapat *header* dengan tombol kembali dan judul halaman. Area tengah didominasi peta interaktif dengan penanda lokasi untuk menunjukkan titik penjemputan. Pada bagian bawah tersedia kartu informasi alamat yang menampilkan detail lokasi terpilih, serta tombol aksi utama Konfirmasi Lokasi Ini untuk menyimpan pilihan. Desain menekankan kemudahan memilih dan memastikan lokasi secara cepat dan akurat.



Sumber : Penulis (2026)

Gambar 12. Implementasi Tampilan Buat Order Penjemputan

Gambar 12 menampilkan *form* pembuatan order penjemputan dalam bentuk modal yang muncul di atas halaman beranda. Di bagian atas terdapat judul *form*, diikuti informasi alamat lokasi. Selanjutnya tersedia beberapa kolom *input* seperti jarak, berat, harga, nama, dan nomor telepon, serta tombol untuk memilih tanggal penjemputan. Pada bagian bawah terdapat dua tombol aksi, yaitu Batal dan Simpan. Desain dibuat terpusat dan terstruktur agar pengguna dapat mengisi data pesanan dengan mudah.

4. Testing

Pada tahap pengujian aplikasi yang berkaitan dengan interaksi *user* menggunakan *Blac-kbox* Testing. Pelaksanaan pengujian ditujukan untuk mengetahui apakah fungsi masukan dan keluaran aplikasi untuk berinteraksi sesuai dengan spesifikasi kebutuhan. Pengujian dilakukan dengan berbagai *testcase* kemungkinan yang akan terjadi baik benar atau salah. Berikut hasil pengujian menggunakan metode Black-Box Testing:

Tabel 1. Pengujian Black-Box

Skenario	Test Case	Harapan	Hasil	Kesimpulan
Pengujian registrasi	Seluruh masukan teks diisikan dengan data yang	Aplikasi dapat menyimpan data pendaftar	Sesuai	Valid

Skenario	Test Case	Harapan	Hasil	Kesimpulan	Skenario	Test Case	Harapan	Hasil	Kesimpulan
Penguji n login	sesuai kemudian tekan tombol Daftar Seluruh masuk teks tidak diisi kemudian tekan tombol Daftar Satu bagian atau lebih masuk teks tidak diisi kemudian tekan tombol Daftar Email dan password diisi dengan benar Email diisi dengan benar dan password diisi dengan salah Email diisi dengan salah dan password diisi	Aplikasi dapat menampilkan pesan bagian harus diisi	Sesuai	Valid	Pengujian Order Penjempuan	dengan benar Seluruh masuk teks diisi dengan data yang sesuai kemudian tekan tombol Simpan Seluruh masuk teks tidak diisi kemudian tekan tombol Simpan Satu bagian atau lebih masuk teks tidak diisi kemudian tekan tombol Simpan	Aplikasi dapat menyimpan data pendaftar	Sesuai	Valid
		Aplikasi dapat menampilkan pesan bagian harus diisi	Sesuai	Valid			Aplikasi dapat menampilkan pesan bagian harus diisi	Sesuai	Valid
		Aplikasi menampilkan tampilan dashboard	Sesuai	Valid			Aplikasi dapat menampilkan pesan bagian harus diisi	Sesuai	Valid
		Aplikasi menampilkan pesan salah, kembali tampil halaman login	Sesuai	Valid					
		Aplikasi menampilkan pesan salah, kembali tampil halaman login	Sesuai	Valid					

Sumber : Penulis (2026)

Berdasarkan Tabel 1 hasil pengujian tampilan utama aplikasi dengan menggunakan metode *black-box* testing seluruh *testcase* hasilnya adalah sesuai dan dikatakan valid.

5. Deployment

Tahapan *deployment* ditujukan untuk melakukan pengujian berdasarkan penggunaan oleh *user* pengguna memastikan apakah sistem tersebut telah sesuai dengan yang disetujui.

6. Maintenance

Tahap *maintenance* dilakukan untuk menjaga aplikasi tetap berjalan stabil, aman, dan sesuai kebutuhan pengguna. Kegiatan meliputi perbaikan *bug*, pembaruan fitur, peningkatan performa, serta penyesuaian kompatibilitas dengan sistem operasi atau perangkat terbaru. Proses *maintenance* dilaksanakan secara periodik.

IV. KESIMPULAN

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan, penelitian ini berhasil mengembangkan aplikasi bank sampah berbasis *mobile* bernama GoTrash dengan pendekatan arsitektur *RESTful API* yang terintegrasi. Sistem mampu menggabungkan modul autentikasi pengguna, pemetaan lokasi menggunakan *Maps API*, pengelolaan transaksi pembayaran melalui Midtrans, serta sinkronisasi data dan notifikasi *real time* dengan Firebase dalam satu platform terpadu. Integrasi tersebut memungkinkan proses pemesanan penjemputan, penentuan lokasi, pemantauan status layanan, hingga transaksi pembayaran dilakukan secara lebih efisien, transparan, dan terstruktur.

Hasil pengujian menggunakan metode *black-box* testing menunjukkan bahwa seluruh fungsi utama aplikasi berjalan sesuai dengan kebutuhan dan spesifikasi yang telah ditetapkan. Selain itu, penggunaan *wireframe* pada tahap perancangan membantu menghasilkan antarmuka yang sistematis, responsif, dan mudah digunakan. Dengan demikian, aplikasi GoTrash mampu meningkatkan efektivitas layanan bank sampah digital serta memberikan kontribusi praktis dan konseptual dalam pengembangan arsitektur API terintegrasi untuk layanan publik berbasis *mobile*. Penelitian ini diharapkan dapat menjadi referensi bagi pengembangan sistem pengelolaan sampah digital yang lebih modern dan skalabel di masa mendatang.

V. REFERENSI

Bagus Syafiq Faqihuddin, Acep Taryana, & Mulki Indana Zulfa. (2025). PENGEMBANGAN APPLICATION PROGRAMMING INTERFACE (API) APLIKASI PENDETEKSI PENYAKIT TANAMAN MENGGUNAKAN METODE DEVOPS. *Jurnal SINTA: Sistem Informasi Dan Teknologi Komputasi*, 2(2).
<https://doi.org/10.61124/sinta.v2i2.42>

Calista Chandra, Wijaya, F., Gunawan, J. A., Lee, J. R., & Maulana, A. (2024). Perancangan dan Implementasi RESTful API untuk Aplikasi Mobile Pembelajaran Flora dan Fauna pada Google Cloud Platform. *SATESI: Jurnal Sains Teknologi Dan Sistem Informasi*, 4(1), 58–69.
<https://doi.org/10.54259/satesi.v4i1.2850>

Fadhil, R. H., & Sutarnan. (2024). Implementasi RESTful API dan Push Notifikasi pada Sistem Informasi Acara Kebudayaan di Indonesia Berbasis Android. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 5(1), 417–426.
<https://doi.org/10.35870/jimik.v5i1.503>

Fauzi, B. N., & Fachrie, M. (2024). Implementasi API Payment Gateway Midtrans pada Sistem Reservasi Dokter Gigi Berbasis Mobile. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 5(1), 662–671.
<https://doi.org/10.35870/jimik.v5i1.535>

Fitria, O., Hasanah, N., Pd, M., & Untari, R. S. (2020). *BUKU AJAR REKAYASA PERANGKAT LUNAK Diterbitkan oleh UMSIDA PRESS UNIVERSITAS MUHAMMADIYAH SIDOARJO 2020*.

Ghani, M. A., Zuama, R. A., Gunawan, D., & Matihudin, A. L. (2023). Implementasi Metode Waterfall Dalam Mengembangkan Sistem Informasi Ujian Online Dengan Fitur Proctoring. *Informatics for Educators And Professionals : Journal of Informatics*, 7(2), 218–225.

Hidayat, W. F., Malau, Y., Purnama, R. A., & Setiadi, A. (2024). Penerapan Model Design Thinking Pada Perancangan Aplikasi Informasi Desa Wisata Kabupaten Bantul. In *Computer Science (CO-SCIENCE)* (Vol. 4, Number 2).
<http://jurnal.bsi.ac.id/index.php/co-science>

Hs, A. A., Sunusi, P. M., Nurwahyuni, A., Reskita, D., Mustari, M. L. M., & Sonni, A. F. (2025). *Analisis Bibliometrik: Penerapan Mobile Application "Bank Sampah" sebagai Layanan Pengelolaan Sampah di Era Digital*. 5(2), 137–147.

- <https://doi.org/10.51135/kambotivol5issu e2p>
- Latif, N., Arifin, R., & Ratnawati. (2024). Aplikasi Pencarian Lokasi Bank Sampah Aktif... ZONAsi. In *Jurnal Sistem Informasi* (Vol. 6, Number 3).
- Lina Nurlaili, A., & Mustika Rizki, A. (2025). Implementasi Metode Waterfall dalam Pengembangan Sistem Informasi Praktik Kerja Lapangan. *ILKOMNIKA: Journal of Computer Science and Applied Informatics E*, 7(2), 211–223. <https://doi.org/10.28926/ilkomnika.v7i2.781>
- Oktaviana Pamudji, P., & Kamisutara, M. (2024). *Implementasi Metode Waterfall dalam Pengembangan Sistem Informasi Narotama Career Center*.
- Ponggawa, V. V, Loleh, S. J., Sendiang, M., Simanjuntak, C. H., Lumentut, V. M. A., Lumunon, E. S. A., & Kimbal, A. A. (2024). *Perancangan RESTful API Polimdo Dengan Metode Object Relational Mapping (ORM)*.
- Putri, D. S., Voutama, A., & Heryana, N. (2023). Analisis Aksesibilitas Fasilitas Kesehatan di Kota Pekanbaru, Indonesia. *Journal Information System Development (ISD)*, 8(1), 1–6. <https://doi.org/10.19166/isd.v8i1.581>
- Satria, P. W., Ligasi, S. P., Yani, L. I., Azzahra, S., Rahayu, S., Islam, U., Raden, N., & Palembang, F. (2024). *Manajemen Proyek Sistem Informasi Menggunakan Metode Waterfall Untuk Meningkatkan Efisiensi Pada Usaha Catering*. 2(11), 792–799. <https://doi.org/10.5281/zenodo.14489913>
- Septiawan, E., Sakethi Dan, D., Andrian, R., Komputer, J. I., Matematika, F., Ilmu, D., & Alam, P. (2022). *Penerapan Metode Waterfall Pada Pengembangan Sistem Informasi Monitoring Proses Bimbingan Skripsi di Jurusan Ilmu Komputer Universitas Lampung* (Vol. 3, Number 1).
- Susanto, B. M., & Hariyanto, A. (2023). Aplikasi Penentuan Rute Terpendek Pengambilan Sampah Berbasis Internet of Things Terintegrasi Dengan Google Maps API. *Angkasa: Jurnal Ilmiah Bidang Teknologi*, 15(1). <https://doi.org/10.28989/angkasa.v15i1.1651>