

Eco-Fort : Sistem Informasi Pemantauan Sampah Benteng RotterdamHaerani^{1*}, M. Rizky Kurniawan Tuhulele², Magfirah Aulia³, Muh Afriliawan S⁴^{1,2,3,4} Pendidikan Teknik Informatika dan Komputer, Universitas Negeri Makassar, Makassar, IndonesiaCorresponding e-mail : rani090304@gmail.com**ARTICLE INFO****Keywords:**

Agile
 Benteng Rotterdam
 Eco-Fort
 Pemantauan Sampah
 Sistem Informasi

ABSTRAK

Pada wisata Benteng Rotterdam yang terletak di Jalan Ujung Pandang, Bulu Gading, Ujung Pandang di kota Makassar, Sulawesi Selatan, terdapat permasalahan pengelolaan dan pendataan sampah yang masih dilakukan secara manual. Hal ini menyebabkan pengunjung menjadi terbatas karena lokasi-lokasi pembuangan sampah yang kurang efektif dan efisien. Solusi untuk mengatasi masalah tersebut adalah dengan membuat sistem informasi Eco-Fort. Sistem ini dirancang untuk memberikan informasi yang jelas dan mudah diakses mengenai lokasi-lokasi tempat sampah, sehingga dapat mengurangi perilaku membuang sampah sembarangan oleh pengunjung. Eco-Fort adalah sebuah situs web yang dirancang khusus untuk pengendalian dan pemantauan sampah. Metode pengembangan sistem yang digunakan adalah Agile, karena memungkinkan pengembangan yang bertahap dan berulang. Sistem ini akan membantu dalam pengendalian, pemantauan, dan pendataan lokasi tempat sampah sehingga menjadi lebih terorganisir dan efisien. Dengan adanya Eco-Fort, diharapkan pengelolaan sampah di Benteng Rotterdam dapat ditingkatkan, memberikan pengalaman yang lebih baik bagi pengunjung, serta berkontribusi pada pelestarian lingkungan.

Article History

Received: November 2,
 2024
 Revised: Desember 17.
 2024
 Accepted: Januari 26, 2025

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

To cite this article : Author. (20xx). Title. Information Technology Education Journal, X(X), XX-XX.
 Doi. xxxx

PENDAHULUAN

Wisata adalah kegiatan bepergian atau berkunjung ke suatu tempat untuk tujuan rekreasi, bersantai, atau mendapatkan pengalaman baru. Wisata sering melibatkan perjalanan ke lokasi yang berbeda dari tempat tinggal sehari-hari dan dapat mencakup berbagai aktivitas seperti mengunjungi tempat-tempat bersejarah, menikmati pemandangan alam, berpartisipasi dalam kegiatan budaya, atau sekadar menikmati liburan di tempat yang menyenangkan[2]. Peningkatan jumlah penduduk dan perkembangan aktivitas ekonomi di perkotaan telah menyebabkan peningkatan volume sampah yang dihasilkan. Penanganan sampah yang tidak efektif dapat menimbulkan berbagai masalah lingkungan, kesehatan, dan sosial. Sampah yang tidak dikelola dengan baik dapat mencemari air, tanah, dan udara, serta menjadi tempat berkembang biak bagi penyakit. Oleh karena itu, manajemen sampah yang baik sangat penting untuk menjaga kebersihan lingkungan dan kesehatan masyarakat.

Benteng Rotterdam adalah salah satu tempat wisata yang terkenal dan sering dikunjungi oleh wisatawan lokal maupun internasional. Benteng ini memiliki nilai sejarah yang tinggi dan merupakan salah satu ikon kota Makassar. Namun, dengan peningkatan jumlah pengunjung, volume sampah yang dihasilkan di area tersebut juga meningkat. Sampah-sampah ini terdiri dari berbagai jenis, seperti plastik, kertas, sisa makanan, dan botol minuman, yang jika tidak dikelola dengan baik dapat merusak keindahan dan kenyamanan tempat wisata[4]. Sayangnya, manajemen sampah di Benteng Rotterdam belum sepenuhnya efektif dan terorganisir dengan baik, sehingga menyebabkan dampak negatif terhadap lingkungan sekitar dan pengalaman

pengunjung. Sampah yang menumpuk tidak hanya merusak pemandangan, tetapi juga dapat mengganggu kesehatan pengunjung dan masyarakat sekitar. Selain itu, sampah yang tidak dikelola dengan baik dapat menurunkan kualitas wisata, yang pada akhirnya dapat mengurangi minat wisatawan untuk berkunjung.

Eco-Fort adalah sistem informasi pengendalian dan pemantauan sampah yang dirancang untuk mengoptimalkan proses pengumpulan, pengangkutan, dan pengolahan sampah. Integrasi berbagai komponen teknologi informasi seperti sensor, perangkat lunak, dan jaringan komunikasi memungkinkan sistem ini untuk memantau secara real-time kondisi tempat pembuangan sampah, tingkat pengisian kontainer sampah, dan rute pengangkutan sampah. Tujuan utama dari sistem ini adalah meningkatkan efisiensi operasional dengan mengurangi biaya pengelolaan sampah dan mengoptimalkan penggunaan sumber daya. Selain itu, Eco-Fort juga bertujuan untuk meningkatkan transparansi dan akuntabilitas dalam pengelolaan sampah, memungkinkan partisipasi aktif masyarakat dalam menjaga kebersihan lingkungan [5]. Dalam rangka mengatasi permasalahan manajemen sampah yang ada di Benteng Rotterdam, kami akan merancang dan menyusun langkah-langkah konkret untuk implementasi Sistem Informasi Pemantauan Sampah. Dokumen ini akan menjadi panduan bagi semua pemangku kepentingan untuk memahami peran dan kontribusi mereka dalam keberhasilan implementasi sistem ini, dengan harapan dapat meningkatkan efektivitas pengelolaan sampah, mengurangi dampak negatif terhadap lingkungan, meningkatkan kesadaran akan pentingnya pengelolaan sampah, serta meningkatkan pengalaman pengunjung di tempat wisata tersebut. Dalam era digital ini, penerapan sistem informasi pengendalian dan pemantauan sampah menjadi semakin relevan karena tantangan lingkungan semakin meningkat. Berbagai kota besar di seluruh dunia telah mengadopsi teknologi serupa untuk mengatasi masalah sampah dengan lebih efektif. Dengan memanfaatkan data yang dikumpulkan melalui sistem ini, pemerintah dan pemangku kepentingan dapat membuat keputusan yang lebih tepat dan berbasis data dalam mengelola sampah, serta mengembangkan strategi yang lebih efektif dalam menjaga kebersihan lingkungan.

Pendahuluan ini akan membahas latar belakang dan pentingnya sistem informasi pengendalian dan pemantauan sampah, serta manfaat yang diharapkan dari penerapannya. Fokus utama adalah bagaimana sistem ini dapat berkontribusi terhadap pengelolaan sampah yang lebih berkelanjutan dan efisien di masa depan. Pendekatan ini merupakan langkah awal yang penting dalam memperkenalkan Eco-Fort, sebuah sistem inovatif yang diharapkan dapat menjadi solusi untuk masalah pengelolaan sampah di perkotaan. Dengan pendekatan yang terintegrasi antara teknologi informasi, monitoring real-time, dan partisipasi masyarakat, Eco-Fort bertujuan untuk menciptakan lingkungan yang lebih bersih, sehat, dan berkelanjutan. Dalam konteks era digital saat ini, penerapan teknologi seperti Eco-Fort menjadi semakin relevan dan diperlukan untuk menghadapi tantangan lingkungan yang semakin kompleks. Melalui pendekatan ini, diharapkan bahwa pengelolaan sampah dapat menjadi lebih efisien, transparan, dan responsif terhadap kebutuhan masyarakat serta lingkungan hidup [6].

METODE

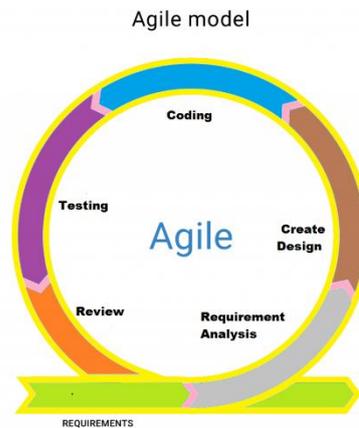
Waterfall

Metode Agile adalah metode yang digunakan untuk mengembangkan website Eco-Fort, yang menyediakan pendekatan pengembangan perangkat lunak yang iteratif dan fleksibel. Metode ini disebut sebagai model iteratif inkremental atau alur hidup dinamis, yang menekankan pada siklus pembangunan yang berulang, dikenal sebagai iterasi atau sprint. Setiap iterasi melibatkan serangkaian tahap, dimulai dari perencanaan, analisis, dan desain, dilanjutkan dengan pengembangan, pengujian, dan pengiriman perangkat lunak. Pada tahap perencanaan, tim

mengidentifikasi dan memprioritaskan fitur atau user stories yang akan dikerjakan dalam sprint tersebut. Ini termasuk menentukan tujuan, cakupan, dan timeline untuk sprint yang akan dijalankan. Perencanaan yang matang memastikan bahwa tim memiliki panduan yang jelas dan target yang realistis untuk dicapai [1].

Tahap analisis dan desain melibatkan kolaborasi antara tim pengembangan dan pemangku kepentingan untuk memahami kebutuhan pengguna dan merancang solusi yang sesuai. Pendekatan seperti user stories, prototyping, dan desain iteratif digunakan untuk memastikan bahwa solusi yang dirancang benar-benar memenuhi kebutuhan pengguna. Selama tahap pengembangan (coding), tim mengimplementasikan fitur-fitur yang telah direncanakan dalam sprint. Pengembangan dilakukan dengan pendekatan inkremental, di mana fitur-fitur ditambahkan secara bertahap. Kode yang dibuat harus terstruktur, bersih, dan sesuai dengan standar yang telah ditetapkan. Penggunaan alat-alat otomatisasi dan integrasi berkelanjutan (CI/CD) juga umum dalam tahap ini untuk meningkatkan efisiensi dan kualitas kode. Tahap pengujian dilakukan secara terintegrasi selama siklus pengembangan. Setiap fitur yang dikembangkan diuji secara menyeluruh untuk memastikan kesesuaian dengan kebutuhan dan kualitas yang diharapkan. Pengujian meliputi uji unit, uji integrasi, uji sistem, dan uji penerimaan pengguna (UAT). Automasi pengujian juga sering digunakan untuk mempercepat proses dan memastikan konsistensi. Setelah setiap sprint selesai, tahap pengiriman dan umpan balik memastikan bahwa inkrement produk disampaikan kepada pengguna untuk mendapatkan umpan balik. Umpan balik ini sangat penting untuk merancang sprint berikutnya dan memastikan bahwa produk selalu berkembang sesuai dengan kebutuhan pengguna.

Tahap ini juga melibatkan evaluasi kinerja tim dan proses untuk perbaikan berkelanjutan. Salah satu keunggulan utama dari metode Agile adalah kemampuannya untuk merespons perubahan kebutuhan pasar dengan cepat. Dalam konteks pengembangan website Eco-Fort, metode Agile memungkinkan tim untuk beradaptasi dengan perubahan kebutuhan pengguna atau permintaan baru dari pemangku kepentingan dengan mudah. Hal ini memastikan bahwa produk yang dihasilkan tidak hanya memiliki nilai yang tinggi, tetapi juga relevan dengan kebutuhan pengguna. Umpan balik dari pengguna atau pelanggan menjadi inti dari pendekatan Agile ini, memungkinkan perbaikan terus-menerus dan pengembangan produk yang lebih sesuai dengan kebutuhan dan harapan pengguna. Dengan pendekatan ini, tim dapat merespons lebih cepat terhadap perubahan pasar dan memastikan bahwa produk yang dihasilkan memiliki nilai yang tinggi serta relevan dengan kebutuhan pengguna[3]. Dengan mengadopsi metode Agile, pengembangan website Eco-Fort dapat berjalan lebih efisien dan efektif, memberikan solusi yang tepat waktu dan berkualitas tinggi kepada pengguna. Pendekatan iteratif dan inkremental ini memastikan bahwa setiap iterasi menghasilkan peningkatan yang nyata dan sesuai dengan harapan pengguna, menjadikan Eco-Fort sebagai platform yang adaptif dan responsif terhadap kebutuhan pengelolaan sampah di Benteng Rotterdam.



Gambar. 1 Metode Agile

SDLC (Software Development Life Cycle) adalah proses yang sistematis untuk mengubah suatu sistem dengan menggunakan perangkat lunak, serta mengembangkan sistem perangkat lunak berdasarkan praktik terbaik dan metode pengujian yang baik. SDLC adalah pendekatan yang sangat terstruktur dan berfokus pada kolaborasi tim, responsivitas terhadap perubahan, dan pengiriman nilai yang cepat kepada pengguna. Dalam SDLC, tahapan-tahapan pengembangan perangkat lunak biasanya mencakup analisis kebutuhan, desain sistem, pengembangan, pengujian, implementasi, dan pemeliharaan.[15] Metode Agile adalah salah satu pendekatan dalam SDLC yang menggabungkan siklus iteratif dan inkremental untuk pengembangan perangkat lunak yang lebih adaptif dan fleksibel. Metode ini sangat cocok untuk menghadapi perubahan kebutuhan pasar dan pengguna. Agile mendorong transparansi dan akuntabilitas dalam proses pengembangan dengan memastikan setiap iterasi diuji dan dievaluasi secara menyeluruh sebelum disampaikan kepada pengguna akhir. Metode ini telah menjadi sangat populer dalam pengembangan perangkat lunak modern, membantu organisasi mencapai tujuan bisnis mereka dengan lebih efisien dan efektif.[15]

Tahapan dalam Metode Agile:

1. Perencanaan: Setiap iterasi atau sprint dimulai dengan sesi perencanaan. Tim mengidentifikasi dan memprioritaskan fitur atau user stories yang akan dikerjakan dalam sprint tersebut. Tujuannya adalah menentukan sasaran dan cakupan dari sprint yang akan dijalankan, serta menetapkan timeline dan pembagian tugas yang jelas.
2. Analisis dan Desain: Pada tahap ini, dilakukan analisis kebutuhan dan desain solusi secara kolaboratif. Tim menggunakan pendekatan seperti user stories, prototyping, dan desain iteratif untuk memahami kebutuhan pengguna dan merancang solusi yang sesuai. Kegiatan ini melibatkan semua pemangku kepentingan untuk memastikan bahwa semua aspek kebutuhan dan keinginan pengguna dipahami dengan baik.
3. Pengembangan (Coding): Tahap ini melibatkan implementasi fitur-fitur yang telah direncanakan dalam sprint. Pengembangan dilakukan dengan pendekatan inkremental, di mana fitur-fitur ditambahkan secara bertahap. Kode yang dibuat harus terstruktur, bersih, dan sesuai dengan standar yang telah ditetapkan. Penggunaan alat-alat otomatisasi dan integrasi berkelanjutan (CI/CD) juga umum dalam tahap ini untuk meningkatkan efisiensi dan kualitas kode.
4. Pengujian: Pengujian dilakukan secara terintegrasi selama siklus pengembangan. Setiap fitur yang dikembangkan diuji secara menyeluruh untuk memastikan kesesuaian dengan kebutuhan dan kualitas yang diharapkan. Pengujian meliputi uji unit, uji integrasi, uji sistem, dan uji penerimaan pengguna (UAT). Automasi

pengujian juga sering digunakan untuk mempercepat proses dan memastikan konsistensi.

5. Pengiriman dan Umpan Balik: Setelah setiap sprint selesai, inkrement produk disampaikan kepada pengguna untuk mendapatkan umpan balik. Umpan balik ini sangat penting untuk merancang sprint berikutnya dan memastikan bahwa produk selalu berkembang sesuai dengan kebutuhan pengguna. Tahap ini juga melibatkan evaluasi kinerja tim dan proses untuk perbaikan berkelanjutan.

HASIL DAN PEMBAHASAN

Analisis Kebutuhan

a. Kebutuhan Fungsional

Kebutuhan fungsional merujuk pada spesifikasi atau persyaratan yang menjelaskan fungsionalitas sistem atau perangkat lunak yang harus dipenuhi untuk memenuhi kebutuhan pengguna. Berbeda dengan kebutuhan non-fungsional yang lebih menitikberatkan pada aspek-aspek seperti kinerja, keamanan, dan ergonomi, kebutuhan fungsional menentukan apa yang sebenarnya harus dilakukan oleh sistem. Ini mencakup fitur-fitur yang sistem harus sediakan, operasi-operasi yang harus dijalankan, dan tanggapan-tanggapan yang diharapkan dari sistem terhadap masukan pengguna tertentu.

Tabel 1. Kebutuhan Fungsional

ID	Kebutuhan Fungsional	Penjelasan
	Fitur Login/Sign In	Pengguna dapat masuk ke dalam sistem (login) dengan menggunakan nama pengguna dan kata sandi yang tepat. Fitur ini memastikan bahwa hanya pengguna yang terdaftar yang dapat mengakses fitur-fitur tertentu dari sistem. Apabila pengguna belum memiliki akun, mereka dapat melakukan pendaftaran (sign up) terlebih dahulu dengan mengisi informasi yang diperlukan seperti nama, alamat email, dan kata sandi. Sistem akan memverifikasi informasi yang dimasukkan dan membuat akun baru untuk pengguna tersebut. Setelah berhasil login, pengguna dapat mengakses profil mereka dan fitur lainnya yang memerlukan autentikasi.
	Fitur Pemetaan Lokasi	Pengguna dapat melihat peta dengan lokasi-lokasi sampah di sekitar tempat wisata. Fitur ini menggunakan teknologi peta interaktif yang memungkinkan pengguna untuk mencari dan menemukan tempat-tempat di mana tempat sampah tersedia atau di mana sampah sering ditemukan. Pengguna dapat memperbesar atau memperkecil peta, serta mendapatkan petunjuk arah ke lokasi yang diinginkan. Informasi ini membantu pengguna untuk lebih sadar akan kebersihan lingkungan sekitar mereka dan mempermudah dalam membuang sampah pada tempatnya.
	Fitur Pemantauan	Pengguna dapat melihat informasi lengkap tentang keadaan sampah di tempat wisata, termasuk tingkat kebersihan dan lokasi sampah. Fitur ini menyediakan data real-time mengenai kebersihan di berbagai lokasi wisata, yang diperoleh melalui laporan pengguna

		atau sensor kebersihan yang dipasang di lokasi tersebut. Pengguna dapat melihat tingkat kebersihan, frekuensi pengosongan tempat sampah, dan area-area yang membutuhkan perhatian khusus. Informasi ini membantu dalam meningkatkan kesadaran lingkungan dan mendorong pengguna untuk berperan aktif dalam menjaga kebersihan tempat wisata.
	Fitur Edukasi	Fitur edukasi Eco-Fort adalah sumber informasi yang menyajikan materi tentang lingkungan dan pengelolaan sampah. Pengguna dapat mengakses artikel, video, dan infografis yang menjelaskan pentingnya menjaga kebersihan lingkungan, cara mendaur ulang, dan berbagai tips untuk mengurangi sampah. Fitur ini bertujuan untuk meningkatkan kesadaran dan pengetahuan pengguna tentang isu-isu lingkungan serta mendorong mereka untuk mengadopsi praktik ramah lingkungan dalam kehidupan sehari-hari.
	Fitur Pelaporan	Pengguna dapat membuat laporan tentang masalah sampah yang mereka temui, seperti menumpuknya sampah di suatu tempat. Laporan ini dapat mencakup deskripsi masalah, lokasi, dan foto yang relevan. Admin dapat mengelola laporan ini dengan memberikan tanggapan, mengubah status laporan (misalnya dari "Diajukan" menjadi "Sedang Diproses" atau "Selesai"), dan mengambil tindakan yang diperlukan untuk menyelesaikan masalah. Fitur ini memastikan bahwa masalah sampah dapat diidentifikasi dan ditangani dengan cepat dan efisien, serta meningkatkan keterlibatan komunitas dalam menjaga kebersihan lingkungan.
	Fitur Keluhan & Saran	Fitur keluhan dan saran Eco-Fort memungkinkan pengguna untuk menyampaikan umpan balik tentang kunjungan mereka ke lokasi tertentu. Pengguna dapat memberikan keluhan mengenai fasilitas kebersihan, pelayanan, atau kondisi lingkungan, serta memberikan saran untuk perbaikan. Admin akan meninjau dan merespons umpan balik ini, yang dapat digunakan untuk meningkatkan kualitas layanan dan kebersihan di tempat wisata. Fitur ini membantu dalam menciptakan lingkungan yang lebih bersih dan nyaman bagi pengunjung, serta membangun komunikasi yang baik antara pengelola dan pengguna.

b. Kebutuhan Non Fungsional

Kebutuhan non-fungsional mencakup segala hal di dalam sistem yang tidak secara langsung terkait dengan operasi dasarnya tetapi mempengaruhi pengalaman pengguna secara keseluruhan. Ini termasuk aspek-aspek seperti keamanan sistem, kinerja, skalabilitas, dan keandalan. Keamanan memastikan bahwa data sensitif dilindungi dari akses yang tidak sah, sementara kinerja mengacu pada seberapa cepat dan responsif sistem tersebut dalam menanggapi permintaan pengguna. Skalabilitas mengukur sejauh mana sistem dapat menangani lonjakan lalu lintas atau peningkatan pengguna tanpa mengalami penurunan kinerja yang signifikan, sementara keandalan mencakup seberapa sering sistem mengalami kegagalan dan seberapa cepat dapat dipulihkan dari kegagalan tersebut. Kebutuhan non-fungsional ini merupakan komponen kunci dalam merancang sistem yang dapat memberikan pengalaman pengguna yang memuaskan dan dapat diandalkan.

Tabel 2. Kebutuhan Non Fungsional

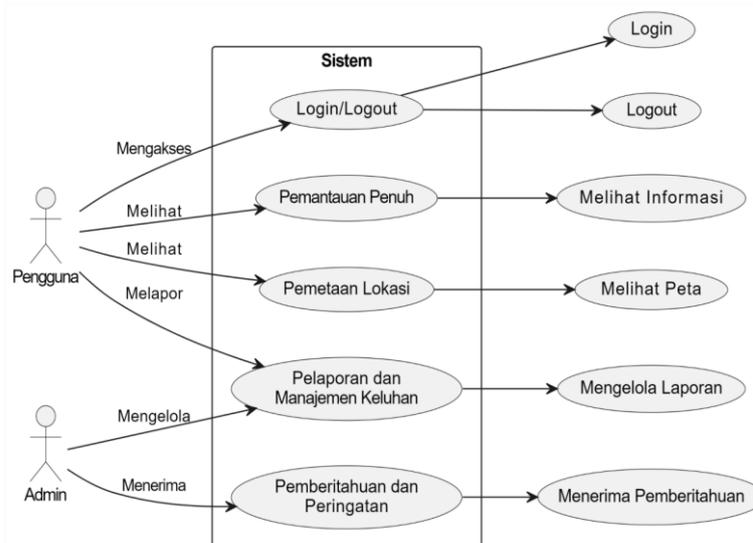
ID	Parameter	Kebutuhan
NFR1	Ketersediaan	Sistem Eco-Fort harus tersedia 24/7 tanpa adanya waktu downtime yang tidak terjadwal. Ini berarti sistem harus dapat diakses oleh pengguna kapan pun dibutuhkan tanpa gangguan dari waktu henti yang tidak terencana. Ketersediaan yang tinggi menjadi kunci untuk memastikan bahwa layanan yang diberikan oleh Sistem Eco-Fort selalu dapat diandalkan.
NFR2	Keandalan	Sistem Eco-Fort harus dapat beroperasi secara konsisten tanpa kegagalan yang signifikan. Keandalan mengacu pada kemampuan sistem untuk berfungsi tanpa gangguan atau kerusakan yang menyebabkan penurunan kualitas layanan. Dalam konteks Eco-Fort, keandalan yang tinggi penting untuk memastikan bahwa proses pengelolaan sampah berlangsung lancar dan terpercaya.
NFR3	Ergonomi	Antarmuka pengguna Eco-Fort haruslah intuitif dan mudah digunakan oleh berbagai kalangan. Ergonomi antarmuka pengguna mencakup desain yang ramah pengguna, navigasi yang mudah dipahami, dan pengalaman pengguna yang menyenangkan secara keseluruhan. Hal ini memastikan bahwa pengguna dari berbagai latar belakang dan tingkat keahlian dapat dengan mudah berinteraksi dengan sistem.
NFR4	Portabilitas	Sistem Eco-Fort harus dapat diakses dari berbagai perangkat dan platform, termasuk perangkat mobile. Portabilitas mengacu pada kemampuan sistem untuk beroperasi di berbagai lingkungan teknologi tanpa perubahan yang signifikan. Dengan demikian, pengguna dapat mengakses Sistem Eco-Fort melalui berbagai perangkat, meningkatkan fleksibilitas dan kenyamanan pengguna.
NFR5	Memori	Penggunaan memori sistem Eco-Fort harus dioptimalkan untuk memastikan kinerja yang optimal. Pengoptimalan memori penting untuk meminimalkan penggunaan sumber daya dan memastikan respons sistem yang cepat. Ini juga membantu dalam mengurangi biaya infrastruktur dan meningkatkan efisiensi operasional.
NFR6	Waktu Respon	Sistem Eco-Fort harus memberikan respons dalam waktu kurang dari 2 detik untuk setiap permintaan pengguna. Waktu respons yang cepat merupakan faktor penting dalam pengalaman pengguna yang memuaskan. Dengan memberikan respons yang cepat, pengguna akan merasa sistem responsif dan efisien dalam menanggapi kebutuhan mereka.
NFR7	Keamanan	Akses ke sistem Eco-Fort harus terbatas hanya untuk pengguna yang memiliki otorisasi, dengan autentikasi ganda. Data harus disimpan dan ditransmisikan secara terenkripsi untuk melindungi kerahasiaan dan integritasnya. Keamanan merupakan aspek krusial dalam pengelolaan data pengguna dan informasi sensitif terkait dengan pengelolaan sampah. Melalui praktik keamanan yang ketat, Sistem Eco-Fort dapat mencegah akses tidak sah dan melindungi informasi pengguna.
NFR8	Keselamatan	Sistem Eco-Fort harus memiliki mekanisme keselamatan yang memadai untuk melindungi pengguna dan lingkungan dari potensi

		bahaya terkait dengan pengelolaan sampah. Keselamatan dalam konteks ini mencakup perlindungan terhadap risiko fisik maupun lingkungan yang mungkin timbul dari proses pengelolaan sampah. Sistem Eco-Fort harus dilengkapi dengan fitur-fitur keselamatan yang dirancang untuk mencegah kecelakaan dan mengurangi dampak negatif pada lingkungan.
NFR9	Bahasa Komunikasi	Semua interaksi pengguna dengan sistem Eco-Fort, termasuk tanya jawab dan pesan, harus menggunakan bahasa Indonesia. Penggunaan bahasa yang konsisten dan sesuai dengan preferensi pengguna merupakan faktor penting dalam memastikan pengalaman pengguna yang optimal. Dengan menggunakan bahasa Indonesia, Sistem Eco-Fort dapat lebih mudah diakses dan dipahami oleh pengguna yang berbicara dalam bahasa tersebut.

Tahap Desain

a. Use Case Diagram

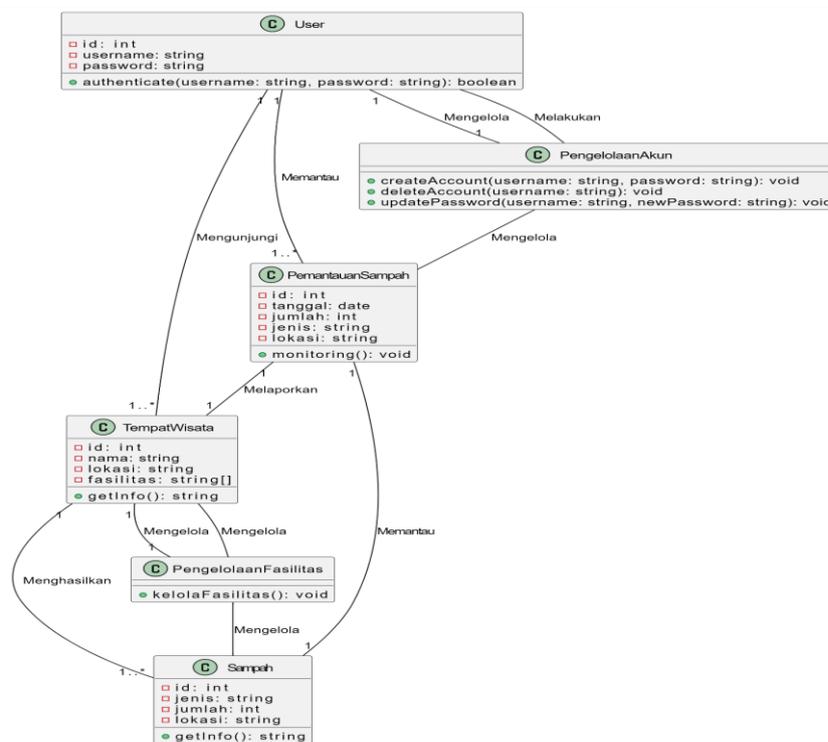
Use case adalah representasi grafis dari interaksi antara sistem dan aktor-aktor yang terlibat dalam suatu proses atau sistem. Diagram ini membantu dalam pemodelan dan pemahaman tentang bagaimana pengguna atau aktor-aktor lainnya berinteraksi dengan sistem untuk mencapai tujuan tertentu. Dalam diagram use case, terdapat dua komponen utama: aktor dan use case. Aktor adalah entitas eksternal yang berinteraksi dengan sistem, seperti pengguna manusia, sistem lain, atau perangkat keras. Use case adalah tindakan atau fungsi yang dilakukan oleh sistem untuk memenuhi kebutuhan atau tujuan tertentu dari aktor. Dengan menggunakan simbol-simbol grafis yang khas, diagram use case menggambarkan skenario-skenario atau alur-alur interaksi antara aktor dan use case dalam sistem. Hal ini membantu para pengembang perangkat lunak dalam merancang sistem yang responsif terhadap kebutuhan pengguna dan mengidentifikasi kebutuhan fungsional yang harus dipenuhi oleh sistem. Diagram use case juga berguna dalam komunikasi antara tim pengembang, pemangku kepentingan, dan pihak lain yang terlibat dalam pengembangan sistem, karena menyajikan gambaran visual yang jelas tentang bagaimana sistem akan digunakan dan berinteraksi dengan lingkungan eksternalnya.[7]



Gambar 2. Use Case Diagram

b. Class Diagram

Diagram kelas adalah representasi visual dari struktur statis atau arsitektur kelas dalam sistem berorientasi objek. Diagram ini menggambarkan kelas-kelas dalam sistem, beserta atribut-atribut dan metode-metode yang dimiliki oleh masing-masing kelas, serta hubungan antar kelas tersebut. Dalam diagram kelas, kelas-kelas direpresentasikan sebagai kotak dengan tiga bagian: bagian atas berisi nama kelas, bagian tengah berisi atribut-atribut dari kelas tersebut, dan bagian bawah berisi metode-metode atau perilaku yang dimiliki oleh kelas tersebut. Hubungan antar kelas ditunjukkan dengan panah yang menghubungkan kelas-kelas, menunjukkan hubungan seperti pewarisan (inheritance), asosiasi, agregasi, atau komposisi. Diagram kelas membantu dalam memodelkan struktur statis dari sistem berorientasi objek, dengan memberikan pandangan yang jelas tentang kelas-kelas yang terlibat dalam sistem dan hubungan antar mereka. Hal ini berguna dalam fase perancangan sistem, karena membantu dalam mengorganisir dan mengelompokkan kelas-kelas berdasarkan tanggung jawab dan hubungan antar mereka, sehingga memfasilitasi implementasi sistem yang terstruktur dan mudah dipahami.

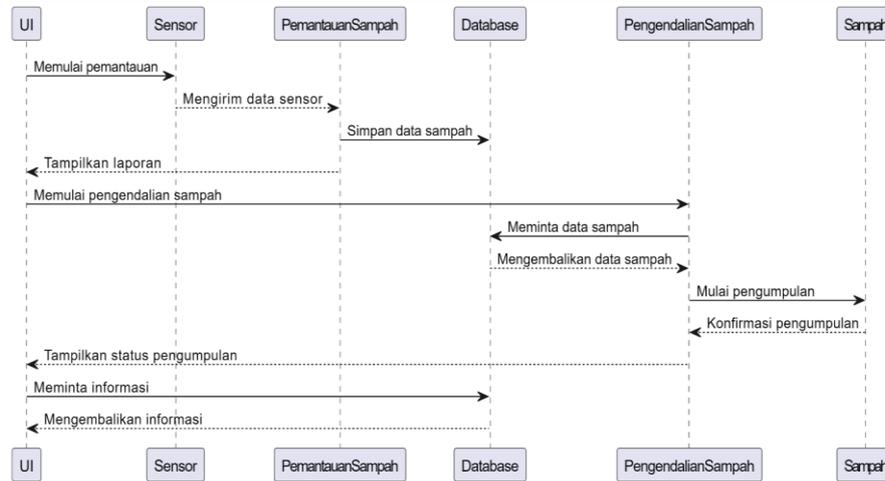


Gambar 3. Class Diagram

c. Sequence Diagram

Sequence diagram adalah representasi grafis dari interaksi antara objek dalam sebuah sistem atau proses, menunjukkan bagaimana pesan atau panggilan metode ditransmisikan dari satu objek ke objek lainnya seiring waktu. Diagram ini memberikan gambaran tentang urutan peristiwa yang terjadi dalam suatu skenario atau proses, dengan menampilkan objek-objek yang terlibat dan pesan-pesan yang dikirim antara objek-objek tersebut. Dalam sequence diagram, objek-objek direpresentasikan sebagai kotak dengan nama objek di bagian atasnya, dan urutan pesan atau panggilan metode direpresentasikan sebagai panah yang menghubungkan objek-objek tersebut. Pesan-pesan tersebut diberi label untuk menunjukkan jenis pesan atau metode

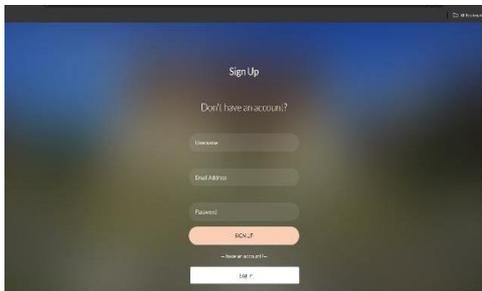
yang dipanggil, serta urutan waktu ketika pesan tersebut dikirim. Sequence diagram membantu dalam memodelkan interaksi antar objek dalam sistem secara terperinci, dengan menunjukkan bagaimana objek-objek berkomunikasi satu sama lain untuk mencapai tujuan tertentu. Hal ini berguna dalam pemahaman tentang aliran kerja sistem, identifikasi ketergantungan antar objek, dan analisis kinerja sistem dalam menanggapi permintaan pengguna atau peristiwa tertentu..



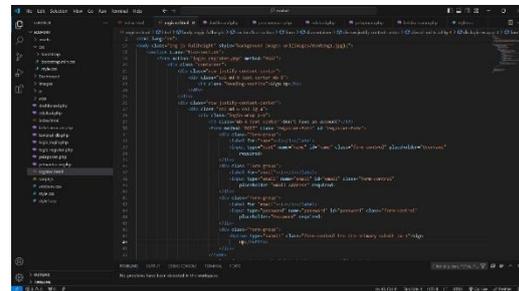
Gambar 4. Sequence Diagram

Tahap Coding

a. Fitur Registrasi



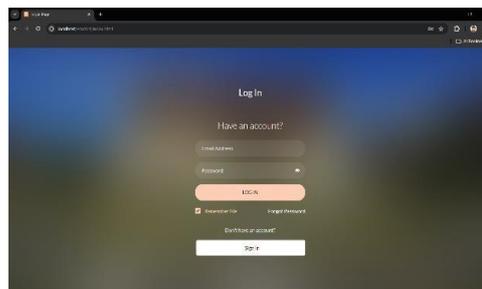
(a)



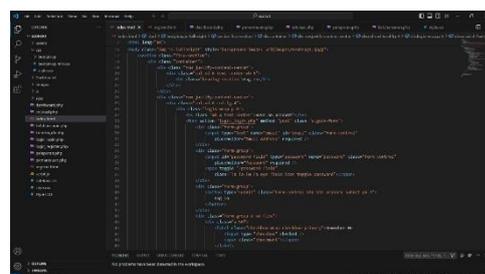
(b)

Gambar 5. (a) Tampilan fitur registrasi, (b) Kode program registrasi

b. Fitur Login



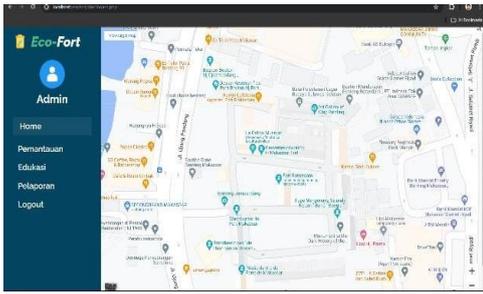
(a)



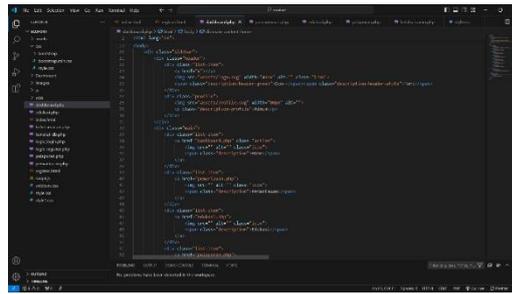
(b)

Gambar 6. (a) Tampilan fitur login, (b) Kode program login

c. Fitur Dashboard



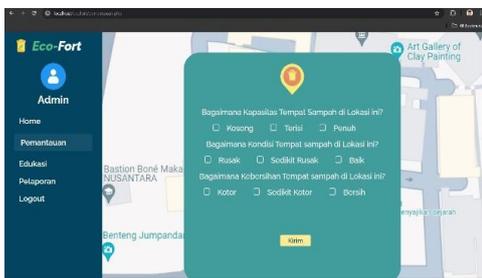
(a)



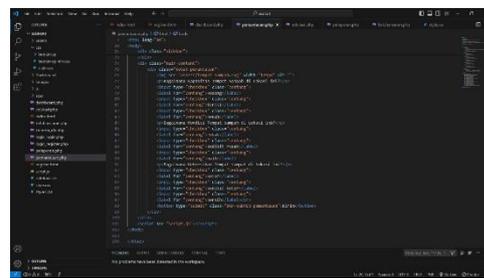
(b)

Gambar 7.(a) Tampilan fitur dashboard, (b) Kode program dashboard

d. Fitur Pemantauan



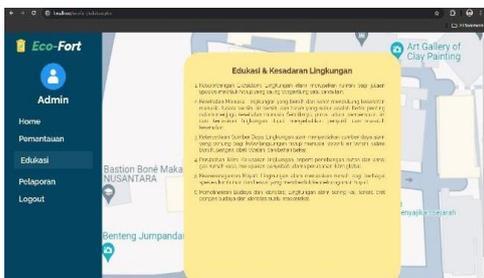
(a)



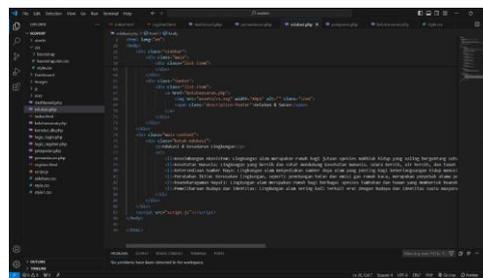
(b)

Gambar 8.(a) Tampilan fitur Pemantauan, (b) Kode program Pemantauan

e. Fitur Edukasi



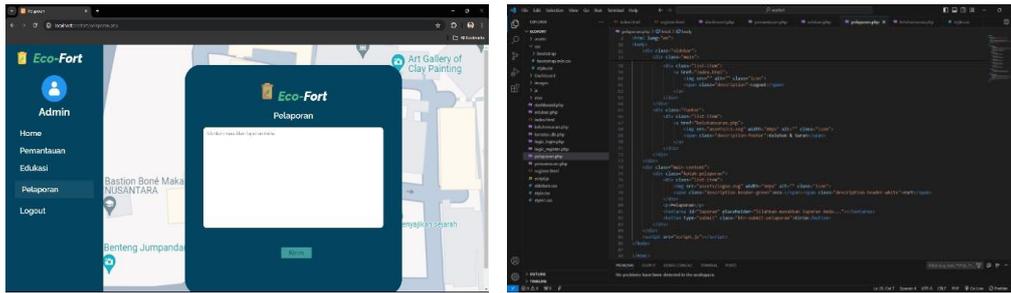
(a)



(b)

Gambar. 9 (a) Tampilan Fitur Edukasi, (b) Kode Program Edukasi

f. Pelaporan

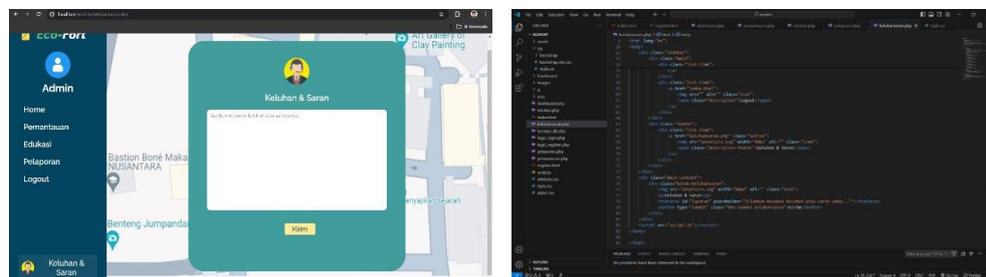


(a)

(b)

Gambar. 10 (a) Tampilan Fitur Pelaporan, (b) Kode Program Pelaporan

g. Fitur Keluhan dan saran



(a)

(b)

Gambar. 11 (a) Tampilan Fitur Keluhan & Saran, (b) Kode Program Keluhan & Saran

Tahap Pengujian

a. Whitebox Testing

Whitebox testing adalah serangkaian pengujian struktural di mana penguji memiliki akses ke internal struktur perangkat lunak seperti source code, database, dan API. Ini memerlukan pemahaman dan keterampilan dalam bahasa pemrograman yang digunakan untuk mengembangkan perangkat lunak. Berbagai teknik whitebox testing termasuk Basis Path Testing, Branch Coverage, Loop Testing, dan Statement Coverage. Dalam pengujian whitebox kali ini, kami fokus pada Teknik Basis Path Testing yang menggunakan Teknik Path Coverage untuk memastikan bahwa semua jalur eksekusi program telah diuji secara menyeluruh. Whitebox testing, khususnya Basis Path Testing, memberikan pendekatan yang komprehensif untuk memastikan bahwa semua jalur eksekusi dalam program telah diuji. Dengan mengidentifikasi jalur basis dan mengembangkan kasus uji yang sesuai, penguji dapat meningkatkan cakupan pengujian dan mengidentifikasi kesalahan yang mungkin tidak terdeteksi dengan metode pengujian lainnya. Whitebox testing memainkan peran penting dalam pengembangan perangkat lunak yang berkualitas tinggi dengan memastikan bahwa kode sumber diuji secara menyeluruh dari dalam.[9]

Basis Path Testing

Basis Path Testing adalah metode whitebox testing yang digunakan untuk menguji semua jalur yang mungkin dalam program dengan mengeksplorasi setiap jalur eksekusi yang mungkin melalui kode sumber. Dalam Basis Path Testing, program diwakili sebagai graf alir kontrol, dan

setiap jalur yang mungkin dari simpul awal ke simpul akhir dianalisis. Tujuan utamanya adalah untuk memastikan bahwa setiap pernyataan dalam program dieksekusi setidaknya satu kali selama pengujian. Teknik ini membantu dalam mengidentifikasi kesalahan logika dan bug yang terkait dengan aliran program [11].

Merancang flowgraph dapat dilakukan program dan flowchart dari suatu fitur system telah dibuat, flowgraph ini sendiri akan menjadi representasi dari flowchart yang telah dibuat terlebih dahulu. Tahap selanjutnya ialah menghitung cyclomatic complexity, nilai cyclomatic complexity ini sendiri didapat melalui nilai rumus $EDGE - NODE + 2$ yang terdapat pada flowgraph, hasil dari nilai $EDGE - NODE + 2$ nantinya akan menghasilkan nilai serangkaian path, misalkan ada 2 path yang didapatkan maka ada 2 kemungkinan jalur pula yang dapat diuji pada tahap pengujian[10].

Branch Coverage

Teknik ini memastikan bahwa setiap cabang dari setiap titik keputusan (seperti if-statements dan switch-cases) dalam program dieksekusi setidaknya sekali. Ini membantu dalam memastikan bahwa semua kondisi benar dan salah dari setiap keputusan diuji.

Loop Testing

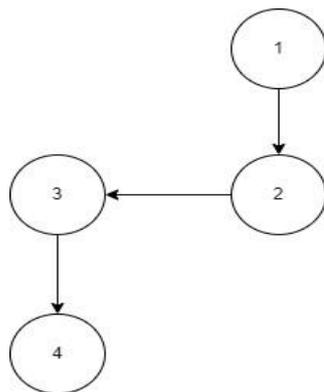
Teknik ini fokus pada pengujian struktur loop dalam program. Ini mencakup pengujian loop dengan nol iterasi, satu iterasi, beberapa iterasi, dan iterasi maksimum untuk memastikan bahwa loop berfungsi dengan benar dalam semua kondisi yang mungkin.

Statement Coverage

Teknik ini memastikan bahwa setiap pernyataan dalam kode sumber dieksekusi setidaknya sekali selama pengujian. Ini membantu dalam memastikan bahwa tidak ada pernyataan yang dilewatkan atau tidak diuji.

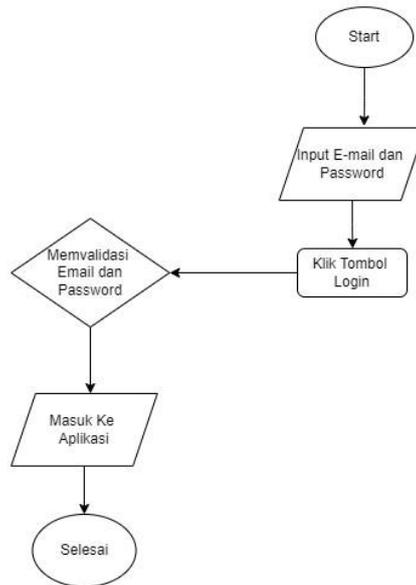
1. Fitur Login

Flowgraph

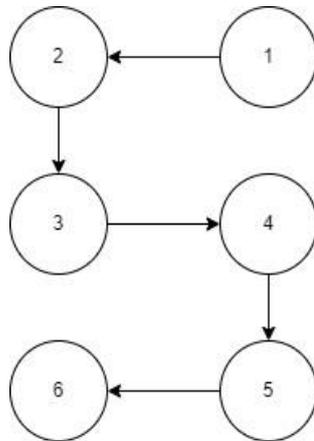


- Menghitung cyclomatic complexity
Node = 4
Edge = 4
Jumlah Region: $E - N + 2 = 2$ (Rangkaian Path)
- Menguji serangkaian path
Path 1: 1-2-3-4 (Pass)

Flowchart

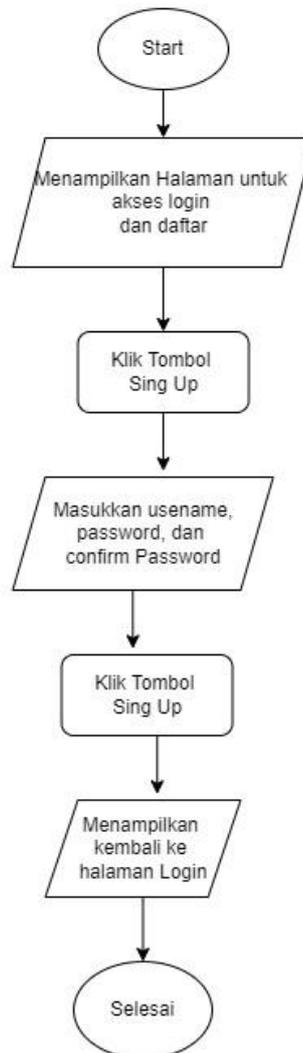


2. Fitur Register
Flowgraph



- Menghitung cyclomatic complexity
Node = 6
Edge = 5
Jumlah Region: $E - N + 2 = 1$ (Rangkaian Path)
- Menguji serangkaian path
Path 1: 1-2-3-4-5-6 (Pass)

Flowchart

b. *Blackbox Testing*

Blackbox Testing adalah metode pengujian perangkat lunak yang dilakukan tanpa memperhatikan struktur internal atau kode sumber dari program yang diuji. Dalam blackbox testing, penguji hanya fokus pada input dan output yang dihasilkan oleh perangkat lunak tanpa memperhatikan bagaimana proses tersebut terjadi di dalamnya. Penguji menganggap perangkat lunak sebagai sebuah "kotak hitam" di mana ia hanya dapat melihat apa yang terjadi di luar, tanpa memperhatikan detail internalnya. Teknik-teknik blackbox testing biasanya melibatkan pengujian fungsional dan non-fungsional. Pengujian fungsional menguji apakah perangkat lunak berperilaku sesuai dengan spesifikasi fungsional yang telah ditentukan, sedangkan pengujian non-fungsional menguji karakteristik seperti kinerja, keamanan, dan kegunaan perangkat lunak. Keuntungan utama dari blackbox testing adalah bahwa tidak diperlukan pengetahuan tentang struktur internal perangkat lunak, sehingga pengujian dapat dilakukan oleh orang yang tidak terlibat dalam pengembangan perangkat lunak. Namun, kelemahannya adalah bahwa pengujian mungkin tidak mencakup semua kemungkinan jalur atau skenario yang mungkin terjadi di dalam program[12].

Teknik Blackbox Testing

1. Equivalence Partitioning: Teknik ini membagi domain input menjadi beberapa kelas ekivalen. Penguji kemudian memilih satu atau lebih nilai dari setiap kelas untuk diuji. Tujuannya adalah untuk mengurangi jumlah kasus uji dengan menganggap bahwa semua nilai dalam satu kelas ekivalen akan diproses dengan cara yang sama.
2. Boundary Value Analysis: Teknik ini menguji batas-batas dari input domain. Penguji memfokuskan pada nilai-nilai yang berada pada batas terendah, batas tertinggi, dan di sekitar batas tersebut. Hal ini dilakukan karena kesalahan sering terjadi pada batas nilai input.
3. Decision Table Testing: Teknik ini menggunakan tabel keputusan untuk menangkap berbagai kombinasi input dan tindakan yang dihasilkan. Setiap baris dalam tabel keputusan mewakili satu kombinasi unik dari kondisi yang diuji dan tindakan yang diharapkan.
4. State Transition Testing: Teknik ini digunakan ketika perangkat lunak memiliki status yang berbeda. Penguji membuat diagram transisi status dan menguji perangkat lunak untuk memastikan bahwa transisi antar status terjadi dengan benar.
5. Use Case Testing: Teknik ini didasarkan pada skenario yang menggambarkan interaksi antara pengguna dan sistem. Penguji mengembangkan kasus uji dari use case yang menjelaskan bagaimana sistem seharusnya berperilaku dalam skenario dunia nyata.

Blackbox testing adalah pendekatan penting dalam siklus pengembangan perangkat lunak yang membantu memastikan bahwa aplikasi berfungsi sesuai dengan spesifikasi yang telah ditentukan. Dengan memfokuskan pada input dan output, penguji dapat mengidentifikasi cacat atau kesalahan dalam fungsi perangkat lunak tanpa memerlukan pemahaman mendalam tentang kode sumber atau struktur internalnya. Namun, penting juga untuk mengkombinasikan metode ini dengan teknik pengujian lain seperti whitebox testing untuk memastikan cakupan pengujian yang lebih komprehensif [13]

Tabel 2. Tabel Kasus Uji

Kode	Kasus Uji	Hasil yang diharapkan	Hasil yang diperoleh	Status
TC1	User memasukkan email dan password yang valid	User berhasil login dan diarahkan ke halaman home page	User berhasil login dan diarahkan ke halaman home page	Pass
TC2	User memasukan data berupa username, email, password, dan konfirmasi password	User berhasil mendaftarkan akun dan data tersimpan ke dalam database	User berhasil mendaftarkan akun dan data tersimpan ke dalam database	Pass
TC3	User melakukan Reset Password	Password berhasil diubah	Password berhasil diubah	Faill

KESIMPULAN

Eco-Fort adalah sistem informasi inovatif yang dirancang untuk mengoptimalkan pengelolaan sampah di Benteng Rotterdam, sebuah tempat wisata terkenal yang memiliki sejarah dan daya tarik budaya yang signifikan. Sistem ini mengintegrasikan teknologi informasi canggih seperti sensor, perangkat lunak, dan jaringan komunikasi untuk memantau secara real-time

kondisi tempat pembuangan sampah, tingkat pengisian kontainer, dan rute pengangkutan sampah. Melalui pemantauan yang terus menerus, Eco-Fort bertujuan untuk meningkatkan efisiensi operasional dalam pengelolaan sampah, mengurangi biaya operasional, serta meningkatkan transparansi dan akuntabilitas dalam pengelolaan lingkungan. Eco-Fort menggunakan sensor yang ditempatkan di berbagai titik tempat pembuangan sampah untuk mendeteksi tingkat pengisian kontainer sampah secara real-time. Data dari sensor ini dikirim ke sistem pusat yang kemudian menganalisis dan memberikan laporan mengenai status kebersihan dan kebutuhan pengosongan kontainer. Selain itu, sistem ini juga menyediakan informasi rute pengangkutan sampah yang optimal bagi petugas kebersihan, sehingga waktu dan sumber daya dapat digunakan lebih efisien. Dalam pengembangan sistem Eco-Fort, metode Agile digunakan untuk memastikan bahwa setiap tahap dalam siklus pengembangan perangkat lunak, mulai dari perencanaan, analisis, desain, pengembangan, pengujian, hingga pengiriman, dilakukan secara iteratif dan fleksibel.

Pendekatan ini memungkinkan tim pengembang untuk beradaptasi dengan perubahan kebutuhan dan harapan pengguna secara cepat dan efisien. Setiap iterasi atau sprint melibatkan penilaian dan umpan balik dari pengguna, yang digunakan untuk melakukan perbaikan dan penyesuaian yang diperlukan. Hal ini memastikan bahwa sistem yang dihasilkan selalu relevan dan mampu memenuhi kebutuhan pengelolaan sampah di Benteng Rotterdam. Eco-Fort juga dirancang untuk meningkatkan kesadaran dan partisipasi masyarakat dalam menjaga kebersihan lingkungan. Melalui fitur edukasi yang ada dalam sistem, pengguna dapat mengakses informasi dan materi edukatif mengenai pentingnya pengelolaan sampah yang baik dan praktik ramah lingkungan. Selain itu, fitur pelaporan dan umpan balik memungkinkan pengunjung untuk melaporkan masalah kebersihan dan memberikan saran untuk perbaikan, yang kemudian dapat ditindaklanjuti oleh pengelola. Sistem ini tidak hanya berfokus pada peningkatan efisiensi operasional, tetapi juga pada keberlanjutan lingkungan. Dengan pemantauan dan pengelolaan sampah yang lebih baik, Eco-Fort membantu mengurangi dampak negatif terhadap lingkungan dan mendukung upaya konservasi di Benteng Rotterdam. Selain itu, sistem ini juga memberikan transparansi dalam pengelolaan sampah, memungkinkan pemangku kepentingan untuk memantau dan mengevaluasi kinerja pengelolaan sampah secara lebih efektif. Dengan mengadopsi teknologi informasi dan metode Agile, Eco-Fort mampu menghadapi tantangan lingkungan yang semakin kompleks dengan solusi yang lebih efisien dan berkelanjutan. Sistem ini merupakan contoh bagaimana inovasi teknologi dapat digunakan untuk mendukung pengelolaan lingkungan yang lebih baik, memberikan manfaat tidak hanya bagi pengelola tetapi juga bagi masyarakat dan lingkungan secara keseluruhan.

REFERENSI

- [1] Andi Rozzi, Y., Fredericka, J., Qurniati, N., Raden Wijaya, H., & Dehasen Bengkulu, U. (2024). Sosialisasi Sistem Pemantauan Tempat Sampah Pintar Secara Real-Time Berbasis IoT di SMAN 1 Kepahiang. *Jurnal Dehasen Untuk Negeri*, 3(1), 41–44.
- [2] Aziz, R., Dewilda, Y., & Putri, B. E. (2020). Kajian Awal Pengolahan Sampah. *Jurnal Sains Dan Teknologi*, 20(1), 77–85.
- [3] Dewi, R. P. (2017). Perancangan Sistem Pengelolaan Sampah Untuk Mendukung Perkembangan Industri Kreatif Di Daerah Pariwisata. *Prosiding Seminar Nasional Multi Disiplin Ilmu & Papers Unisbank Ke-3*, 3(Sendi_U 3), 217–221.
- [4] Ermawati, E. A., Amalia, F. R., & Mukti, M. (2018). Analisis Strategi Pengelolaan Sampah di

- Tiga Lokasi Wisata Kabupaten Banyuwangi. *Journal of Tourism and Creativity*, 2(1), 25.
- [5] Fane, P., Agus, I. P., & Pratama, E. (2023). Jurnal Indonesia : Manajemen Informatika dan Komunikasi SAMPAH BERBASIS ANDROID MENGGUNAKAN MODEL EXTREME PROGRAMMING DAN BLACK BOX *Jurnal Indonesia : Manajemen Informatika dan Komunikasi. Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 4(2), 381-398, 4(2), 381–398.
- [6] Fitri, W. Y., Wibowo, A. W., & Ariyanto, D. B. (2020). Kebijakan Pengelolaan Sampah Di Daerah Utama Tujuan Wisata. *Jurnal Kebijakan Publik*, 11(2), 105.
- [7] Mahrus Satriadi Trisnu, A. (2014). *Issn 1978-8096. EnviroScienteeae*, 10, 80–87.
- [8] Naltaru, M. (2014). Perencanaan Sistem Pengelolaan Sampah Di Kawasan Wisata Bukit Kelam Kabupaten Sintang. *Jurnal Teknologi Lingkungan Lahan Basah*, 2(1), 1–10.
- [9] Nuroji, N. (2023). Penerapan Metode Agile Dalam Permodelan Sistem Informasi Inventory Barang. *Journal of Artificial Intelligence and Technology Information*, 1(4), 138–145.
- [10] Rahmansa, R., Kalsum, T. U., Alamsyah, H., Komputer, R. S., Komputer, I., Dehasen Bengkulu, U., & Artikel, H. (2023). Sistem Monitoring Kapasitas Sampah Pada Bak Sampah Secara Real-Time Berbasis Internet of Things. *Digital Transformation Technology (Digitech) | E*, 3(1), 74–82.
- [11] Ramadhan, J. A., Haniva, D. T., & Suharso, A. (2023). Systematic Literature Review Penggunaan Metodologi Pengembangan Sistem Informasi Waterfall, Agile, dan Hybrid. *JIEET : Journal Information Engineering and Educational Technology*, 07(01), 36–42.
- [12] Saputra, I., Sukmasetya, P., & Primadewi, A. (2023). Implementasi Agile Software Development dalam Perancangan Sistem Pengelolaan Limbah Sampah. 4(3), 1930–1942.
- [13] Wafi, A., Setyawan, H., & Ariyani, S. (2020). Prototipe Sistem Smart Trash Berbasis IOT (Internet Of Things) dengan Aplikasi Android. *Jurnal Teknik Elektro Dan Komputasi (ELKOM)*, 2(1), 20–29.
- [14] Widyowanti, R. A., Kifli, F. W., Moruk, A., Oktaviany, H., Putri, A. G., & Renjani, A. R. (2022). Tangga Didukung Sistem Pemantau-. *Wikrama Parahita: Jurnal Pengabdian Masyarakat*, 6(2), 207–212.
- [15] Handayani, H., Faizah, K. U., Mutiara Ayulya, A., Rozan, M. F., Wulan, D., & Hamzah, M. L. (2023). Perancangan Sistem Informasi Inventory Barang Berbasis Web Menggunakan Metode Agile Software Development Designing a Web-Based Inventory Information System Using the Agile Software Development Method. *Jurnal Testing Dan Implementasi Sistem Informasi*, 1(1), 29–40.