

Pengembangan Aplikasi Pengelolaan Cuti Pegawai Berbasis Web dengan Metode Agile

Sitti Nur Aisya¹, Nur Aprilianti Aswit², Vina³, Nurul Khofifah Hasyim⁴, Marsya Pratiwi⁵, Ferdinand Murni Hamundu⁶

Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Halu Oleo, Kendari

¹sittinuraisya0911@gmail.com, ²apriliantiaswit@gmail.com, ³vinar0316@gmail.com, ⁴nurulkhofifah076@gmail.com, ⁶ferdinand@uho.ac.id

Abstract

Employee leave management is a crucial aspect of personnel administration, particularly in government institutions such as the Organization Bureau of the Southeast Sulawesi Provincial Secretariat. Currently, the leave request process remains manual, utilizing paper-based forms that frequently lead to approval delays, document misplacement, and inefficient tracking. To address these issues, this study develops a Web-Based Employee Leave Management Application using the Python programming language and SQLite database. The application is designed following the Agile methodology to ensure adaptive development responsive to user needs. The results demonstrate that this system significantly reduces leave approval processing time, decreases paper usage, and enhances transparency through real-time tracking features. The implementation of this application is expected to substantially improve administrative efficiency and employee productivity.

Keywords: Web Application, Python, SQLite, Agile Method, Leave Management

Abstrak

Pengelolaan cuti pegawai merupakan aspek penting dalam administrasi kepegawaian, terutama di instansi pemerintah seperti Biro Organisasi Setda Provinsi Sulawesi Tenggara. Selama ini, proses pengajuan cuti masih dilakukan secara manual menggunakan formulir kertas, yang sering menyebabkan keterlambatan persetujuan, hilangnya dokumen, dan ketidakefisienan dalam pelacakan. Untuk mengatasi masalah ini, penelitian ini mengembangkan Aplikasi Pengelolaan Cuti Pegawai Berbasis Web menggunakan bahasa pemrograman Python dan database SQLite. Aplikasi ini dirancang dengan metode Agile untuk memastikan pengembangan yang adaptif terhadap kebutuhan pengguna. Hasilnya, sistem ini mampu memangkas waktu proses persetujuan cuti, mengurangi penggunaan kertas, dan meningkatkan transparansi melalui fitur pelacakan real-time. Dengan implementasi aplikasi ini, diharapkan efisiensi administrasi dan produktivitas pegawai dapat meningkat secara signifikan.

Kata kunci: Aplikasi Web, Python, SQLite, Metode Agile, Manajemen Cuti.

© 2025 Author
Creative Commons Attribution 4.0 International License



1. Pendahuluan

Era transformasi digital telah menjadi pendorong utama dalam modernisasi tata kelola pemerintahan, termasuk dalam sistem manajemen kepegawaian.

Adopsi teknologi digital dalam birokrasi dinilai mampu meningkatkan efisiensi, transparansi, dan akuntabilitas nilai-nilai fundamental yang sejalan dengan prinsip good governance [1],[2]. Namun,

meskipun perkembangan transformasi digital di sektor publik Indonesia menunjukkan kemajuan signifikan, implementasinya di lapangan masih menghadapi berbagai kendala. Salah satu contoh nyata dapat dilihat pada Biro Organisasi Sekretariat Daerah (Setda) Provinsi Sulawesi Tenggara, di mana mekanisme pengajuan dan persetujuan cuti pegawai masih mengandalkan proses manual. Kondisi ini tidak hanya berpotensi menimbulkan keterlambatan administrasi dan hilangnya dokumen, tetapi juga mengurangi tingkat transparansi. Dampak lebih luas dari permasalahan ini adalah penurunan produktivitas pegawai serta meningkatnya beban kerja staf HRD dalam mengelola administrasi yang seharusnya dapat diotomatisasi.

Sesuai PermenPAN RB No. 7 Tahun 2022, instansi pemerintah didorong untuk mengadopsi digitalisasi administrasi demi mempercepat pengambilan keputusan dan meningkatkan pelayanan publik [3]. Oleh karena itu, penelitian ini bertujuan mengembangkan Aplikasi Pengelolaan Cuti Pegawai Berbasis Web dengan metode Agile di Biro Organisasi Setda Provinsi Sultra. Metode Agile dipilih karena pendekatannya yang iteratif dan adaptif, memungkinkan pengembangan sistem yang responsif terhadap kebutuhan pengguna.

Aplikasi ini dirancang untuk menyederhanakan proses pengajuan cuti secara online, validasi kuota otomatis, persetujuan digital, serta pencatatan dan pelaporan terpusat. Sistem ini diharapkan meningkatkan efisiensi, transparansi, mengurangi human error, dan memudahkan akses bagi pegawai. Manfaatnya meliputi peningkatan tata kelola administrasi kepegawaian, kemudahan pengajuan cuti secara real-time bagi pegawai, serta menjadi studi kasus integrasi metode Agile dalam pengembangan aplikasi sektor publik.

Implementasi aplikasi E-Cuti serupa telah terbukti meningkatkan efisiensi dan akuntabilitas di berbagai instansi pemerintahan Indonesia, meski tantangan seperti resistensi perubahan dan keterbatasan infrastruktur masih perlu diatasi [4]. Metode Agile juga efektif dalam pengembangan aplikasi pemerintah, meningkatkan kinerja dan kepuasan pengguna melalui pendekatan kolaboratif [5]. Dengan demikian, aplikasi ini diharapkan memberikan kontribusi signifikan dalam mendukung percepatan digitalisasi pemerintahan dan memperkuat prinsip good governance di tingkat daerah.

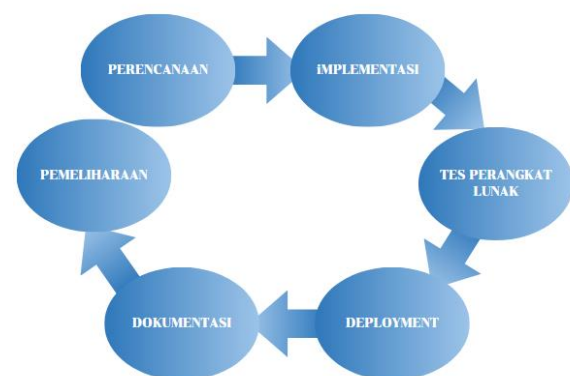
2. Metode Penelitian

2.1. Penerapan Metode Agile

Penelitian ini mengimplementasikan pendekatan Agile sebagai landasan pengembangan sistem digitalisasi manajemen cuti karyawan berbasis web. Pemilihan metodologi ini didasarkan pada kemampuannya dalam mengakomodasi perubahan

kebutuhan pengguna secara fleksibel sepanjang proses pengembangan. Berbeda dengan metode konvensional yang bersifat linear, Agile memungkinkan pengembangan sistem dilakukan secara iteratif melalui siklus-siklus pendek (*sprint*), sehingga setiap tahap dapat dievaluasi dan disempurnakan berdasarkan masukan pengguna secara berkala.

Salah satu keunggulan utama pendekatan Agile dalam penelitian ini adalah kemampuannya dalam meminimalisir risiko ketidaksesuaian produk akhir dengan kebutuhan aktual pengguna. Dengan menerapkan mekanisme umpan balik (*feedback loop*) yang berkelanjutan, tim pengembang dapat melakukan penyesuaian fitur, antarmuka, maupun alur kerja sistem secara dinamis. Selain itu, pendekatan iteratif dalam Agile memastikan bahwa setiap peningkatan fungsionalitas dapat diuji dan divalidasi sebelum masuk ke tahap pengembangan berikutnya.



Gambar 1. Tahap Metode Agile

Tahapan dari metode agile adalah sebagai berikut:

2.1.1. Perencanaan (Planning)

Perencanaan dalam Agile berfokus pada identifikasi kebutuhan awal sistem tanpa mendefinisikan seluruh proyek di awal. Tim mengidentifikasi fitur-fitur yang dibutuhkan, seperti sistem login, pengajuan cuti *online*, dan pelaporan. Fitur-fitur ini kemudian diprioritaskan dan dibagi ke dalam sprint (siklus pengembangan singkat). misalnya, fokus pada pembuatan modul *login* dan pendaftaran akun.

2.1.2. Implementasi (Development)

Implementasi merupakan proses pengembangan aktual fitur-fitur yang telah direncanakan. Misalnya, dalam dua minggu, tim menyelesaikan tampilan *dashboard* admin dan fungsi pengajuan cuti menggunakan *Python* dan *Flask*. Tahap ini menerapkan praktik seperti *pair programming*, *code*

review, dan integrasi berkelanjutan untuk menjaga kualitas kode.

2.1.3. Pengujian (Testing)

Setiap fitur yang selesai langsung diuji, baik secara fungsional (misalnya, apakah pengajuan cuti bisa diproses?) maupun non-fungsional (seperti kecepatan akses dan keamanan data). Jika ditemukan bug, tim segera memperbaikinya sebelum melanjutkan ke fitur berikutnya.

2.1.4. Dokumentasi (Documentation)

Dokumentasi dalam Agile dikembangkan secara iteratif, bukan didefinisikan seluruhnya di awal proyek. Tim membuat dokumentasi yang cukup (*just enough*) dan sesuai kebutuhan, mempertahankannya tetap *up-to-date* seiring perubahan sistem. Meski Agile lebih fokus pada produk daripada dokumen, tim tetap mencatat hal penting seperti pemodelan sistem, dan catatan perubahan fitur sehingga memudahkan pemeliharaan dan pengembangan selanjutnya.

2.1.5. Peluncuran (Deployment)

Deployment dalam Agile adalah proses merilis fitur-fitur yang telah selesai dan teruji kepada pengguna. Pendekatan *continuous deployment* diterapkan untuk merilis fitur secepat mungkin dengan pendekatan *incremental* untuk meminimalkan risiko. Tim menerapkan *deployment* otomatis jika memungkinkan, memiliki strategi *rollback* untuk mengatasi masalah, dan mengkomunikasikan perubahan kepada pengguna.

2.1.6. Pemeliharaan (Maintenance)

Pemeliharaan dalam Agile merupakan proses berkelanjutan untuk menjaga dan mengembangkan sistem berdasarkan *feedback* pengguna. Aktivitas meliputi *refactoring* kode secara berkala, perbaikan *bug*, pemantauan kinerja sistem, dan perencanaan upgrade teknologi jika diperlukan. Hasil evaluasi dari tahap ini menjadi input untuk perencanaan iterasi berikutnya, membentuk siklus pengembangan berkelanjutan.

2.2. Spesifikasi Kebutuhan Perangkat Lunak

2.2.1. Perangkat yang Digunakan

Perangkat lunak dan perangkat keras yang digunakan dalam pengembangan sistem ditampilkan pada tabel berikut:

Tabel 1. Perangkat yang digunakan

Software	Hardware
Visual Studio Code	Laptop Asus Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 4.00 GB RAM

Google Chrome / Edge

Smartphone Android/iOS
(untuk pengujian tampilan
responsif)

SQLite3 / PostgreSQL
Microsoft Office (Word,
Project)
Draw.io

2.2.2 Kebutuhan Fungsional

Kebutuhan fungsional adalah layanan utama yang harus disediakan oleh sistem agar dapat digunakan sesuai dengan fungsinya oleh masing-masing pengguna.

1. Sistem menyediakan *login*, *register*, *logout*, dan verifikasi *email*.
2. Superadmin dapat mengelola akun admin.
3. Admin dapat mengelola data pegawai, atasan, dan laporan cuti.
4. Atasan dapat memverifikasi atau menolak pengajuan cuti dari pegawai.
5. Pegawai dapat mengajukan cuti, melihat status pengajuan, dan mencetak surat cuti.
6. Sistem mendukung notifikasi, ekspor laporan cuti (PDF dan Excel), serta integrasi kalender melalui *Google Calendar API*.

2.2.3. Kebutuhan Non Fungsional

Kebutuhan non-fungsional berkaitan dengan kualitas sistem secara keseluruhan, seperti performa, keamanan, dan kemudahan penggunaan.

1. Sistem harus aman dengan otorisasi berdasarkan peran.
2. Sistem dapat diakses melalui perangkat *mobile* dan *browser*.
3. Antarmuka sistem harus mudah digunakan, responsif, dan memiliki tampilan yang konsisten.
4. Sistem harus fleksibel dan mudah dikembangkan untuk kebutuhan ke depan.

2.2.4. Spesifikasi Platform Pemrograman

Python dengan *framework* Flask dipilih sebagai dasar pengembangan sistem digitalisasi pengelolaan cuti pegawai berbasis *web* karena kombinasi sederhana, fleksibel, dan skalabel. Flask memungkinkan pembangunan aplikasi modular dengan *blueprint* untuk memisahkan fitur seperti pengajuan, persetujuan, dan pelaporan cuti, didukung ekstensi esensial seperti SQLAlchemy (ORM) dan Flask-Login (otentikasi *multi-level*). Sebagai *microframework*, Flask memberi kebebasan memilih komponen tanpa *overhead* fitur bawaan berlebihan, cocok untuk metode Agile yang iteratif. Kemampuan integrasinya dengan *database* (PostgreSQL/MySQL) dan layanan eksternal menjamin skalabilitas untuk pengembangan ke sistem kehadiran atau *payroll* di masa depan.

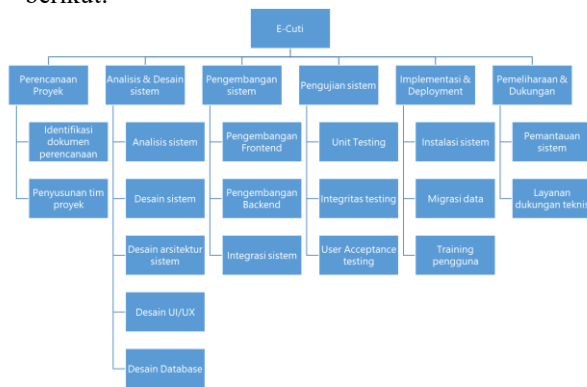
2.3. Perancangan Sistem

Perancangan merupakan fase krusial dalam pengembangan sistem atau perangkat lunak yang

bertujuan menciptakan blueprint produk. Pada tahap ini, tim merumuskan model, arsitektur, dan spesifikasi teknis sebagai panduan implementasi. Beberapa komponen rancangan yang biasanya disiapkan meliputi:

2.3.1. Work Breakdown Structure (WBS)

Penelitian pengembangan sistem digitalisasi pengelolaan cuti pegawai berbasis web dapat diorganisasikan melalui *Work Breakdown Structure* (WBS) dengan mengadopsi fase-fase dalam *Unified Software Development Process* (USDP). Berdasarkan metodologi USDP, WBS dirumuskan pada gambar berikut:



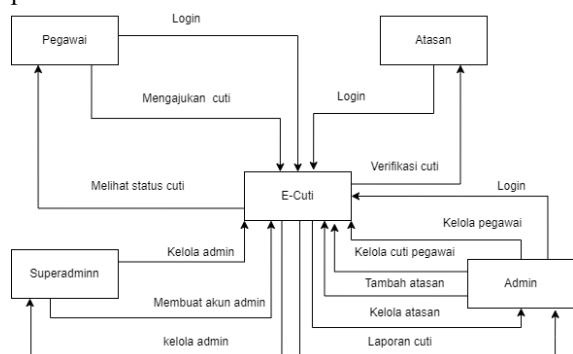
Gambar 2. *Work Breakdown Structure*

2.3.2. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) merupakan alat pemodelan sistem yang digunakan untuk memvisualisasikan alur informasi dalam pengelolaan cuti pegawai berbasis web. DFD ini disusun secara bertahap meliputi tiga level abstraksi, yaitu:

a. Diagram Konteks (*Context Diagram*)

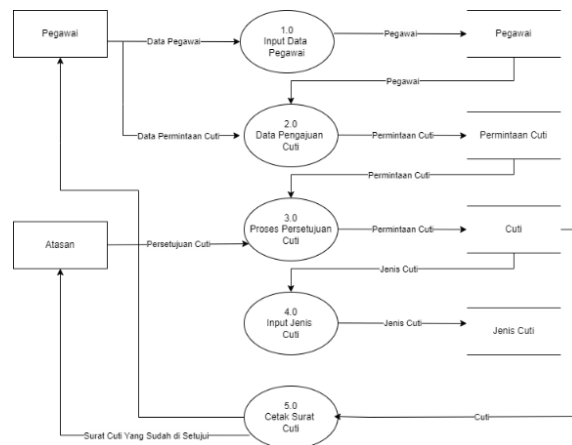
Merupakan level tertinggi dalam DFD yang menggambarkan interaksi antara sistem dengan entitas eksternal secara global. Pada perancangan sistem digitalisasi cuti, diagram konteks berfokus pada hubungan input-output antara sistem dengan aktor-aktor terkait seperti admin, pegawai, dan manajemen. Diagram ini bersifat sederhana karena hanya menampilkan gambaran makro tanpa detail proses internal.



Gambar 3. *Diagram Konteks (Context Diagram)*

b. DFD Level 0

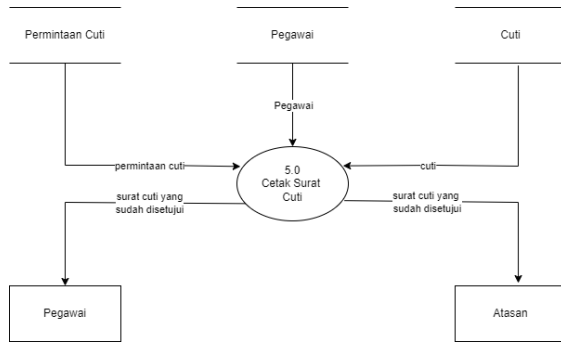
DFD Level 0 (Diagram Alir Data Level 0) adalah penguraian lebih detail dari Diagram Konteks, dimana proses dimulai ketika entitas pegawai menginput data pegawai yang kemudian disimpan ke dalam *database* dengan nama entitas pegawai. Setelah data pegawai tersimpan, pegawai tersebut dapat menginput data permintaan cuti yang juga disimpan dalam *database* dengan nama tabel permintaan cuti. Selanjutnya, entitas bagian Atasan memproses data persetujuan cuti yang menghasilkan permintaan cuti dan menyimpannya ke dalam database dengan nama tabel cuti. Selain itu, jenis cuti juga diinput dan diproses, menghasilkan data jenis cuti yang disimpan dalam database dengan nama tabel jenis cuti. Data cuti kemudian diproses dan dikirim ke atasan untuk diverifikasi. Jika permintaan cuti memenuhi persyaratan, maka akan disetujui; namun jika tidak memenuhi, permintaan tersebut akan ditolak. Seluruh proses ini tetap berjalan dalam satu alur yang terintegrasi dari awal hingga akhir.



Gambar 4. *DFD Level 0*

c. DFD Level 1 Proses 5

Proses ini menggabungkan data dari tabel pegawai, tabel cuti, dan tabel permintaan cuti untuk menghasilkan surat cuti resmi. Setelah data diverifikasi, sistem menyimpan permintaan cuti ke data store cuti, lalu mengolahnya menjadi surat persetujuan. Surat yang sudah dicetak diberikan kepada pegawai sebagai bukti sah cuti, memastikan hanya permintaan yang memenuhi syarat yang diproses. Dengan demikian, pegawai mendapatkan dokumen valid untuk keperluan administrasi.



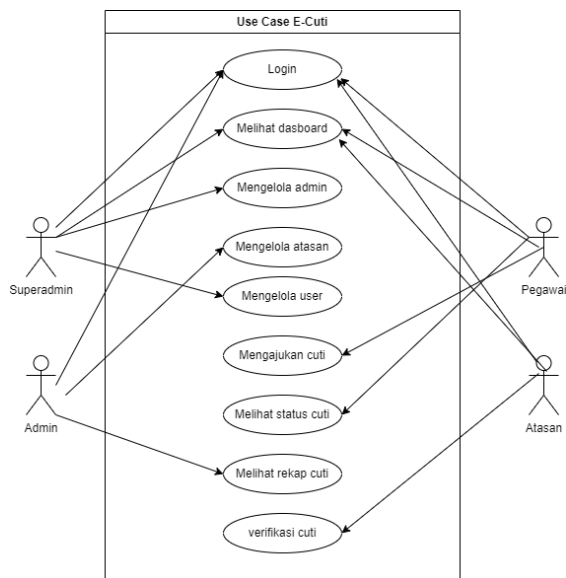
Gambar 5. DFD Level 1 Proses 5

2.3.3. Arsitektur Sistem

Metode yang digunakan dalam perancangan sistem perangkat lunak adalah *Unified Modeling Language* (UML). UML merupakan bahasa berbasis grafik/gambar yang digunakan untuk spesifikasi, visualisasi, pembangunan, dan dokumentasi sistem perangkat lunak berbasis *object oriented programming* (OOP). Terdapat berbagai jenis diagram di dalam *Unified Modeling Language* (UML) antara lain sebagai berikut:

a. Use Case Diagram

Use Case Diagram menggambarkan interaksi antara aktor (pengguna/sistem) dengan fungsionalitas sistem. Dalam pengelolaan cuti pegawai berbasis web, aktor utama meliputi *Superadmin*, *Admin*, *Atasan* dan *Pegawai*.

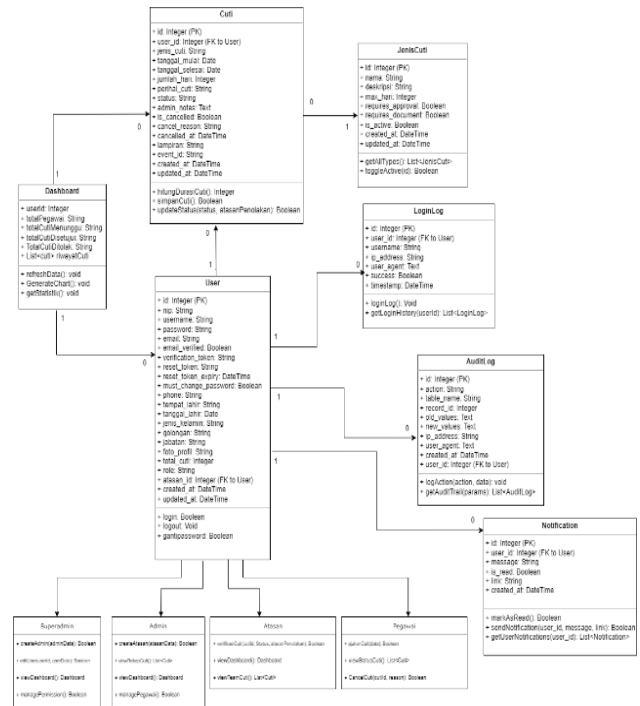


Gambar 6. Use Case Diagram E-Cuti

b. Class Diagram

Class Diagram merupakan salah satu diagram utama dalam pemodelan berorientasi objek (UML) yang menggambarkan arsitektur statis dari suatu sistem. Diagram ini menampilkan kelas-kelas beserta atribut, metode, serta relasi antar kelas seperti asosiasi,

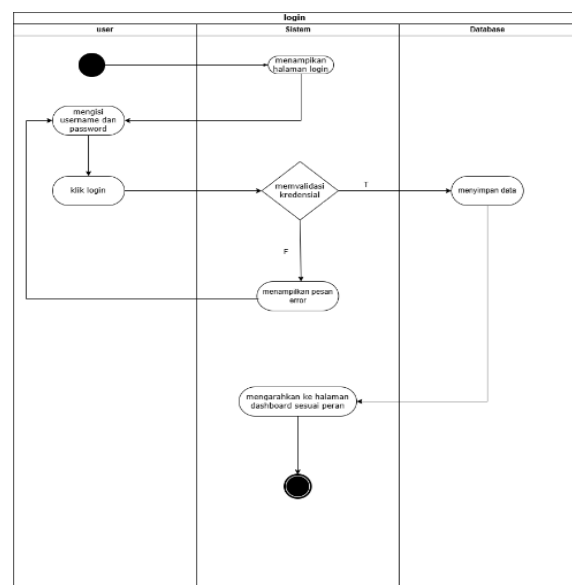
pewarisan (*inheritance*), dan ketergantungan (*dependency*).



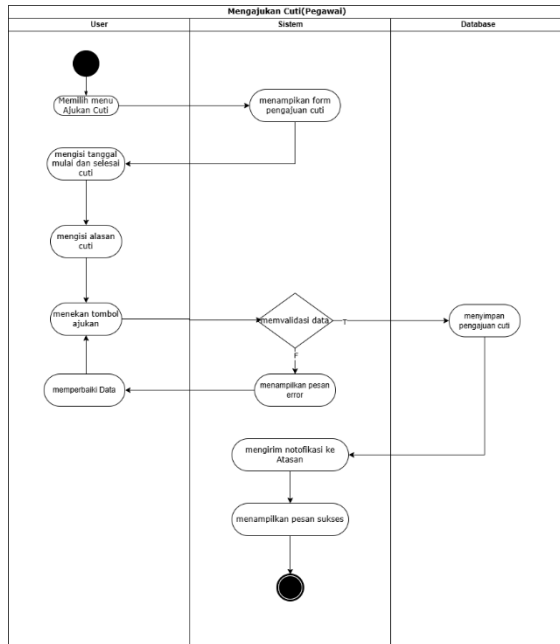
Gambar 7. Class Diagram E-Cuti

c. Activity Diagram

Activity Diagram termasuk dalam jenis diagram UML yang memodelkan alur kerja (*workflow*) atau proses bisnis secara dinamis. Berbeda dengan *flowchart* tradisional, diagram ini mendukung konsep konkurensi (*parallel processing*), *decision node*, serta *swimlane* untuk membagi tanggung jawab antar aktor.



Gambar 8. Activity Diagram Login Semua Role



Gambar 9. Activity Diagram Pegawai Mengajukan Cuti

3. Hasil dan Pembahasan

3.1. Implementasi Sistem

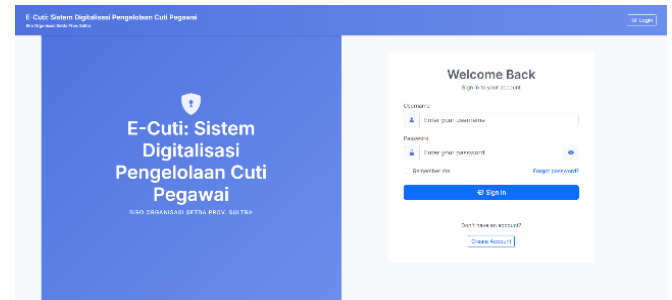
Sistem digitalisasi pengelolaan cuti pegawai dikembangkan sebagai solusi berbasis *web* untuk meningkatkan efisiensi proses administrasi cuti karyawan. Arsitektur sistem ini dibangun menggunakan bahasa pemrograman Python karena kemampuannya dalam pengembangan aplikasi *web* yang cepat dan skalabel, dengan SQLite3 sebagai basis data utama yang dipilih karena sifatnya yang ringan dan mudah diintegrasikan. Adapun hasilnya adalah sebagai berikut:

3.1.1. Tampilan Login

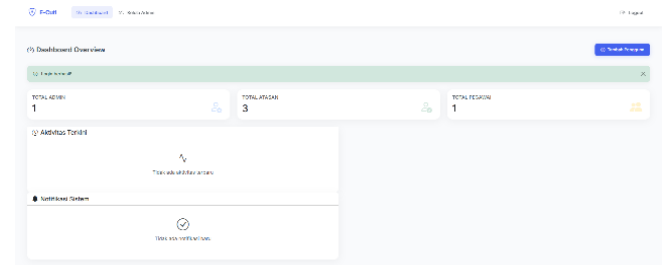
Halaman *login* berfungsi sebagai lapisan keamanan pertama sekaligus pintu masuk bagi pengguna sebelum mengakses sistem. Pada bagian ini, setiap pengguna (baik pegawai maupun admin) harus melakukan autentikasi dengan memasukkan kombinasi *username* dan *password* yang valid. Halaman login dapat dilihat pada gambar 10.

Desain antarmuka dibuat sederhana namun fungsional, memastikan kemudahan penggunaan tanpa mengorbankan aspek keamanan.

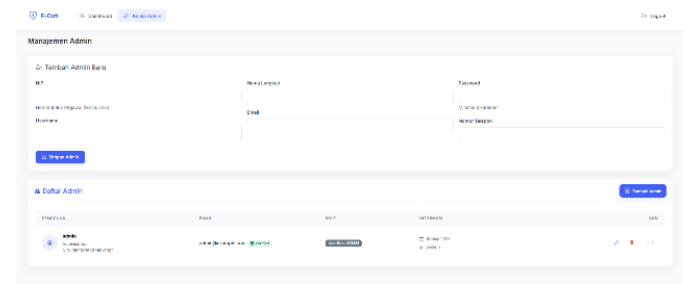
Beberapa User Interface dapat ditinjau seperti: Tampilan Dashboard Superadmin, sampai Tampilan Pegawai Kelola Profil (gambar 11 sampai dengan gambar 24)



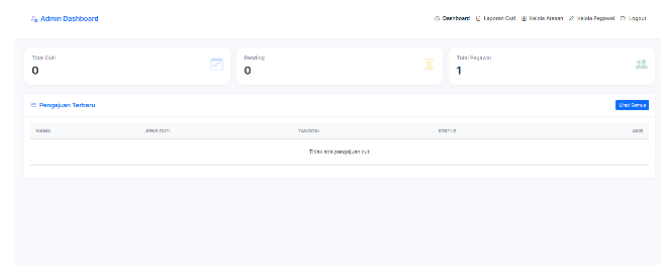
Gambar 10. Tampilan Login



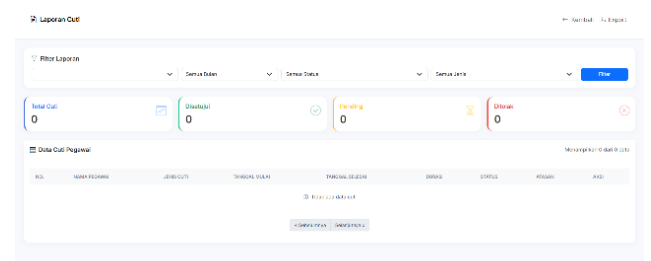
Gambar 11. Tampilan Dashboard Superadmin



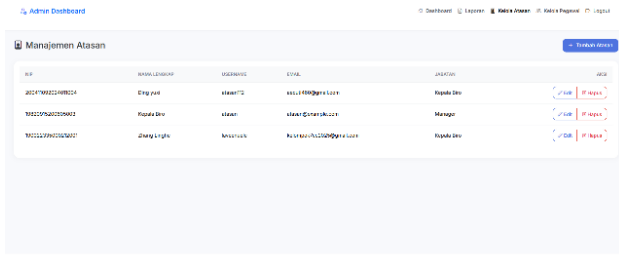
Gambar 12. Tampilan Kelola Admin



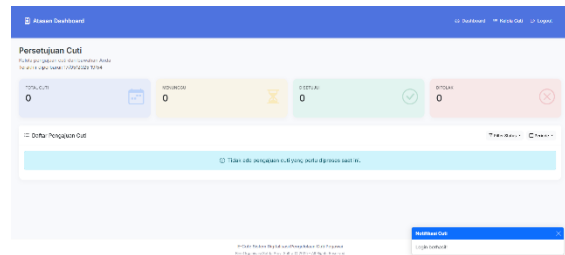
Gambar 13. Tampilan Dashboard Admin



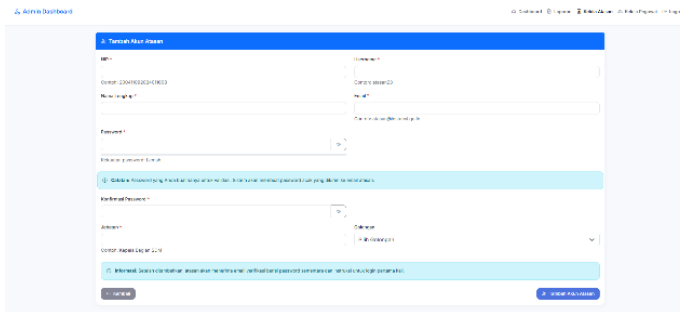
Gambar 14. Tampilan Admin pada Laporan Cuti



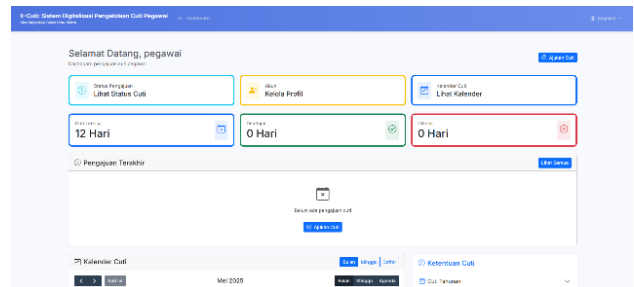
Gambar 15. Tampilan *Admin* Kelola Atasan



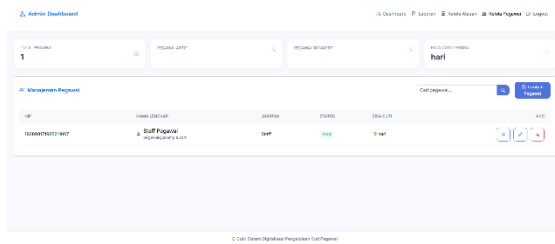
Gambar 20. Tampilan Atasan Kelola Cuti



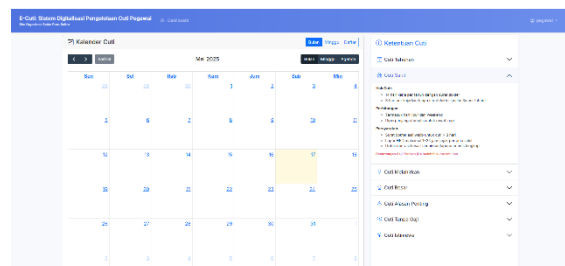
Gambar 16. Tampilan *Admin* Tambah Atasan



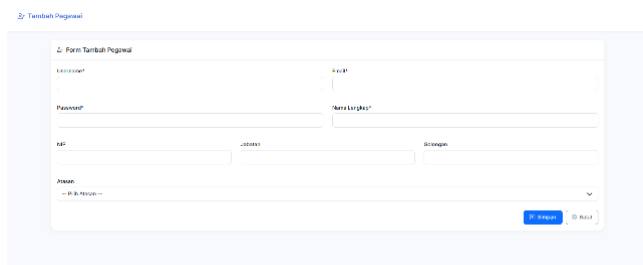
Gambar 21. Tampilan *Dashboard* Pegawai



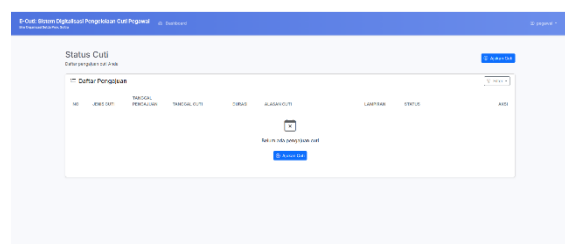
Gambar 17. Tampilan *Admin* Kelola Pegawai



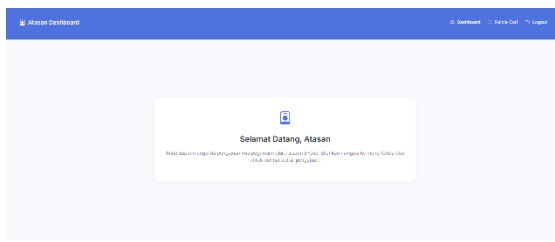
Gambar 22. Tampilan Ketentuan Cuti



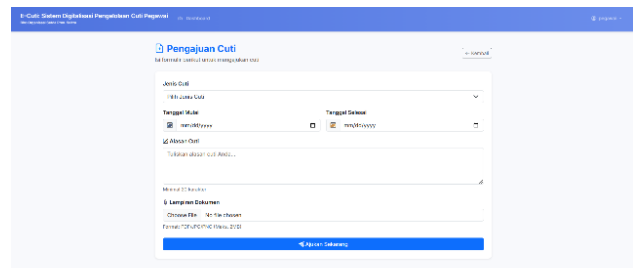
Gambar 18. Tampilan *Admin* Tambah Pegawai



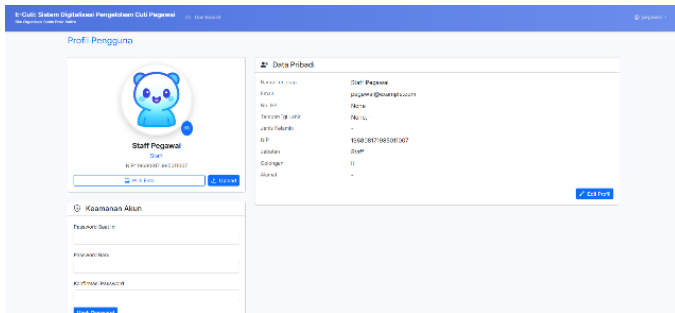
Gambar 23. Tampilan Status Cuti Pegawai



Gambar 19. Tampilan *Dashboard* Atasan



Gambar 24. Tampilan Pegawai mengajukan Cuti



Gambar 24. Tampilan Pegawai Kelola Profil

3.2. Pengujian Sistem

Lalu setelah selesai maka dilakukan Pengujian fungsional dilakukan untuk memastikan setiap fitur berjalan sesuai kebutuhan Hasil pengujian dengan *black-box testing* dapat dilihat pada tabel 2 sampai tabel 5 berikut:

Tabel 2. Tabel Pengujian *Role Superadmin*

Test Case	Prosedur Uji	Hasil yang diharapkan	Hasil Pengujian
Login Superadmin	Masukkan <i>username</i> , <i>password</i> . Klik "Login".	Redirect ke <i>dashboard Superadmin</i> .	Berhasil
Akses Kelola Admin	Klik menu "Kelola Admin"	Halaman <i>kelola admin</i> ditampilkan	Berhasil
Tambah Admin	Isi <i>form</i> tambah <i>admin</i> dan <i>submit</i>	Data <i>admin</i> tersimpan, email verifikasi terkirim	Berhasil
Edit Admin	Edit salah satu data <i>admin</i>	Data <i>admin</i> ter-update di <i>database</i>	Berhasil
Hapus Admin	Klik tombol hapus pada salah satu data <i>admin</i>	Data <i>admin</i> terhapus dari <i>database</i>	Berhasil

Tabel 3. Tabel Pengujian *Role Admin*

Test Case	Prosedur Uji	Hasil yang diharapkan	Hasil Pengujian
Login Admin	Masukkan <i>username</i> , <i>password</i> . Klik "Login".	Redirect ke <i>dashboard Admin</i> .	Berhasil
Akses Kelola Pegawai	Tambah/edit/hapus data pegawai	Data pegawai sesuai dengan aksi	Berhasil
Kelola Cuti Pegawai	Lihat cuti pegawai	Data cuti muncul	Berhasil
Cetak Surat Pegawai	Klik cetak surat pada data cuti	File surat cuti dalam bentuk dokumen tampil/tercetak	Berhasil
Tambah atasan	Isi form tambah atasan dan <i>submit</i>	Data atasan, email verifikasi terkirim	Berhasil
Edit Atasan	Edit salah satu data atasan	Data atasan ter-update di <i>database</i>	Berhasil

Hapus Atasan	Klik tombol hapus pada salah satu data atasan	Data atasan terhapus dari <i>database</i>	Berhasil
Akses menu laporan cuti	Muncul tampilan laporan cuti	Halaman laporan cuti tampil menampilkan daftar cuti pegawai	Berhasil
Export laporan ke PDF	Klik tombol <i>Export PDF</i>	File PDF terunduh	Berhasil
Export Laporan Excel	Klik tombol <i>export Excel</i>	File Excel terunduh, data sesuai tabel di sistem	Berhasil

Tabel 4. Tabel Pengujian *Role Atasan*

Test Case	Prosedur Uji	Hasil yang diharapkan	Hasil Pengujian
Login Atasan	Masukkan <i>username</i> , <i>password</i> . Klik "Login".	Redirect ke <i>dashboard Atasan</i> .	Berhasil
Persetujuan Cuti	Klik pada permohonan cuti → klik <i>acc/tolak</i>	Status cuti berubah menjadi Disetujui/Ditolak	Berhasil
Lampiran Cuti	Buka detail pengajuan cuti yang ada lampiran	Lampiran dapat dilihat atau diunduh	Berhasil

Tabel 5. Tabel Pengujian *Role Pegawai*

Test Case	Prosedur Uji	Hasil yang diharapkan	Hasil Pengujian
Login Pegawai	Isi form cuti dan klik "Ajukan"	Permohonan cuti tercatat & masuk ke antrian.	Berhasil
Lihat Status Cuti	Akses halaman status cuti	Status pengajuan tampil	Berhasil
Cetak Surat	Klik cetak pada status cuti yang disetujui	File surat cuti dalam bentuk dokumen tampil/tercetak	Berhasil
Edit Profil	Edit nama/email/telepon dan simpan	Data admin ter-update di <i>database</i>	Berhasil
Hapus Admin	Klik tombol hapus pada salah satu data admin	Data ter-update di sistem	Berhasil

4. Kesimpulan

Pengembangan sistem digitalisasi pengelolaan cuti pegawai berbasis web dengan pendekatan Agile telah menunjukkan hasil yang positif dalam meningkatkan efisiensi operasional dan transparansi administrasi. Sistem ini tidak hanya menyederhanakan proses pengajuan dan persetujuan cuti, tetapi juga menciptakan alur kerja yang terdokumentasi dengan baik, mengurangi kesalahan manual, serta

memudahkan akses informasi bagi seluruh pegawai. Implementasi fitur-fitur seperti pengajuan online, pelacakan status cuti, dan pelaporan otomatis telah memberikan dampak signifikan terhadap produktivitas organisasi sekaligus meningkatkan kepuasan pegawai dalam mengelola hak cuti mereka secara mandiri. Dari perspektif manajemen sumber daya manusia, sistem ini menjadi contoh nyata bagaimana teknologi dapat mentransformasi proses administratif tradisional menjadi lebih terstruktur dan akuntabel.

Keberhasilan proyek ini sekaligus membuktikan bahwa pendekatan Agile sangat sesuai untuk pengembangan sistem yang membutuhkan fleksibilitas tinggi, terutama dalam menanggapi perubahan kebutuhan pengguna selama proses pengembangan. Namun, untuk penyempurnaan lebih lanjut, beberapa rekomendasi dapat dipertimbangkan. Pertama, penambahan fitur notifikasi *real-time* akan memastikan seluruh pihak terkait selalu mendapatkan pembaruan informasi secara instan. Kedua, peningkatan aspek keamanan melalui verifikasi dua langkah atau enkripsi data perlu diimplementasikan untuk melindungi informasi sensitif pegawai. Terakhir, uji coba sistem dalam skala yang lebih besar dengan melibatkan berbagai skenario penggunaan akan membantu mengidentifikasi potensi masalah sebelum diterapkan secara luas. Dengan penyempurnaan tersebut, sistem ini memiliki potensi untuk menjadi solusi komprehensif dalam digitalisasi manajemen cuti di berbagai jenis organisasi.

Daftar Rujukan

- [1] R. Wagola, A. Nurmandi, Misran, and D. Subekti, "Government digital transformation in Indonesia," in *HCI International 2023 Posters*, 2023, pp. 286–296. doi: 10.1007/978-3-031-36001-5_37.
- [2] A. Latifa and A. Frinaldi, "Transformasi digital dalam manajemen kepegawaian: Studi kasus implementasi aplikasi e-Cuti untuk mewujudkan good governance," *Future Academia: The Journal of Multidisciplinary Research on Scientific and Advanced*, vol. 2, no. 4, 2024. doi: 10.61579/future.v2i4.272.
- [3] Peraturan Menteri Pendayagunaan Aparatur Negara dan Reformasi Birokrasi (Permen PAN-RB) Nomor 7 Tahun 2022 tentang Standar Kompetensi Jabatan Aparatur Sipil Negara, Kementerian PAN-RB, 2022. [Online]. Available: <https://peraturan.bpk.go.id/Details/202233/permen-pan-rb-no-7-tahun-2022>. [Accessed: May 12, 2025].
- [4] Kebijakan Transformasi Digital BKPSDM Kota Yogyakarta, Badan Kepegawaian dan Pengembangan Sumber Daya Manusia (BKPSDM) Kota Yogyakarta. [Online]. Available: <https://bkpsdm.jogjakota.go.id/detail/index/33932>. [Accessed: May 12, 2025].
- [5] R. I. W. Roilan, P. A. Yulianto, and Y. Astuti, "Metode Agile Scrum dalam pembuatan aplikasi permohonan informasi e-PPID Bawaslu," *JOISM: Journal of Information System and Management*, vol. 5, no. 1, Jun. 2023. doi: 10.24076/joism.2023v5i1.1170.
- [6] K. Rahma, B. Kurniawan, and Suryanto, "Aplikasi pemesanan makanan di Bebek dan Ayam Tekaeng menggunakan PHP dan MySQL," *JIK: Jurnal Informatika dan Komputer*, vol. 13, no. 1, pp. 15–26, 2022. [Online]. Available: <https://journal.unmaha.ac.id/index.php/jik/article/view/130>.
- [7] A. M. Ardi, A. Fenty, and L. Lathifah, "Sistem informasi pengajuan cuti pegawai menggunakan metode pengujian ISO 25010 (studi kasus: PT Mutiara Ferindo Internusa)," *J. Inform. Rekayasa Perangkat Lunak*, vol. 4, no. 3, pp. 326–334, 2023. doi: 10.33365/jatika.v4i3.3721.
- [8] G. Y. Pambudi and F. U. Najicha, "Tinjauan yuridis hak cuti bagi pekerja pasca berlakunya Undang-Undang Nomor 11 Tahun 2020 tentang Cipta Kerja," *Gema Keadilan*, vol. 9, no. 2, 2022. doi: 10.14710/gk.2022.16204.
- [9] M. H. Izzuddin and M. D. K. H. Putra, "Implementasi metodologi Agile dalam pengembangan platform kursus online pada platform Eduskill," *Journal on Education*, vol. 6, no. 4, pp. 20158–20166, Mei–Agustus 2024. doi: 10.2654/5497.
- [10] S. Al-Saqqa, S. Sawalha, and H. Abdel-Nabi, "Agile software development: Methodologies and trends," *Int. J. Interact. Mob. Technol.*, vol. 14, no. 11, pp. 1–15, 2020.
- [11] M. Kuhrmann et al., "What makes agile software development agile?," *IEEE Trans. Softw. Eng.*, vol. 48, no. 9, pp. 3523–3539, 2022, doi: 10.1109/TSE.2021.3099532.
- [12] T. A. Pertiwi et al., "Perancangan dan implementasi sistem informasi absensi berbasis web menggunakan metode agile software development," *J. Test. Dan Implementasi Sist. Inf.*, vol. 1, no. 1, pp. 53–66, 2023.
- [13] M. Romzi and B. Kurniawan, "Implementasi pemrograman Python menggunakan Visual Studio Code," *JIK: J. Inform. Komput.*, vol. 11, no. 2, pp. 1–9, 2020. [Online]. Available: www.python.org/DIS/ECS/report2/debate.htm tanggal 4 Juli.
- [14] S. Rahman et al., *Python: Dasar dan pemrograman berorientasi objek*. Tahta Media, 2023. [Online]. Available: <https://tahtamedia.co.id/index.php/issj/article/view/344>